DTU Library

An Exact Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem

Baldacci, Roberto; Mingozzi, Aristide; Roberti, Roberto; Clavo, Roberto Wolfler

Published in:
4 O R

Link to article, DOI:
10.1287/opre.1120.1153

Publication date:
2013

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

OPERATIONS RESEARCH
Vol. 61, No. 2, March–April 2013, pp. 298–314
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.1120.1153

An Exact Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem

Roberto Baldacci
Department of Electronics, Computer Science, and Systems, University of Bologna, 40126 Bologna, Italy, r.baldacci@unibo.it

Aristide Mingozzi
Department of Mathematics, University of Bologna, 40126 Bologna, Italy, mingozzi@csr.unibo.it

Roberto Roberti
Department of Electronics, Computer Science, and Systems, University of Bologna, 40126 Bologna, Italy, roberto.roberti6@unibo.it

Roberto Wolfler Calvo
Laboratoire d'Informatique de Paris Nord, Université de Paris 13; and Sorbonne Paris Cité, CNRS (UMR 7538), 93430 Villetaneuse, France, roberto.wolfler@lipn.univ-paris13.fr

In the *two-echelon capacitated vehicle routing problem* (2E-CVRP), the delivery to customers from a *depot* uses intermediate depots, called *satellites*. The 2E-CVRP involves two levels of routing problems. The first level requires a design of the routes for a vehicle fleet located at the depot to transport the customer demands to a subset of the satellites. The second level concerns the routing of a vehicle fleet located at the satellites to serve all customers from the satellites supplied from the depot. The objective is to minimize the sum of routing and handling costs. This paper describes a new mathematical formulation of the 2E-CVRP used to derive valid lower bounds and an exact method that decomposes the 2E-CVRP into a limited set of *multidepot capacitated vehicle routing problems* with side constraints. Computational results on benchmark instances show that the new exact algorithm outperforms the state-of-the-art exact methods.

Subject classifications: two-echelon vehicle routing; dual ascent; dynamic programming.
Area of review: Transportation.
History: Received November 2011; revision received September 2012; accepted December 2012.

1. Introduction

The *two-echelon capacitated vehicle routing problem* (2E-CVRP) is a two-level distribution system where the deliveries to customers from a *depot* are managed through intermediate capacitated depots, called *satellites*. The first level consists of vehicle routes that start and end at the depot and deliver the customer demands to a subset of satellites. In the 2E-CVRP we consider, a satellite has a limited capacity and can be serviced by more than one first-level route. The second level consists of vehicle routes that start and end at the same satellite and supply all customers. A homogeneous vehicle fleet is used at each level. The first-level vehicles are located at the depot and supply the satellites only. The second-level vehicles have a capacity smaller than that of the first-level vehicles and supply the customers from the satellites. The unloading of first-level vehicles and loading of second-level vehicles at the satellites imply a *handling cost* proportional to the quantity loaded/unloaded.

The 2E-CVRP aims to find two sets of first and second-level routes such that each customer is visited exactly once by a second-level route and the total routing and handling cost is minimized.

1.1. Literature Review

The 2E-CVRP has become a relevant distribution system for supplying customers located in large cities. Because many municipalities impose legal restrictions to keep large vehicles out of city centers, distribution companies create suburban platforms (satellites) where they transport goods with large vehicles. Then, small vehicles service downtown customers from the satellites.

Nonetheless, only recently the 2E-CVRP has received some attention in the literature. Feliu et al. (2007) described a commodity flow formulation and an exact branch-and-cut algorithm that solved instances with up to 32 customers and two satellites. This algorithm was improved by Perboli et al. (2010), Perboli, Tadei, and Vigo (2011) by adding valid inequalities. Perboli, Tadei, and Vigo (2011) reported optimal solutions for instances with up to 32 customers and two satellites, but their model has been shown by Jepsen, Spoorendonk, and Ropke (2013) not to be correct on instances with three or more satellites. Jepsen, Spoorendonk, and Ropke (2013) extended the problem considered by Feliu et al. (2007) and Perboli, Tadei, and Vigo (2011) by introducing fixed costs for the routes of both levels and satellite capacities; they described an exact branch-and-cut algorithm, based on the new formulation and new valid inequalities, that outperforms the method of Perboli, Tadei, and Vigo (2011). Heuristic methods can be found in Crainic et al. (2008, 2011) and Perboli, Tadei, and Vigo (2011). Recently, an adaptive large neighborhood search heuristic has been proposed by Hemmelmayr et al. (2012). Variants of the 2E-CVRP were considered by Tan et al. (2006) and Nguyen et al. (2010).

We consider the 2E-CVRP studied by Jepsen, Spoorendonk, and Ropke (2013). This 2E-CVRP generalizes the *capacitated location routing problem* (LRP), which consists of opening one or more depots, on given locations, and designing, for each open depot, a number of routes to supply customers. A fixed cost and a capacity are associated with each depot. The objective is to minimize the sum of the fixed costs for opening the depots and the routing cost. Exact algorithms for the LRP were presented by Laporte et al. (1986), Akca et al. (2009), Belenguer et al. (2011), and Baldacci et al. (2011b).

1.2. Contributions of This Paper

We introduce a new mathematical formulation of the 2E-CVRP that is used to derive both integer and continuous relaxations. We present a new bounding procedure based on dynamic programming (DP), a dual-ascent method, and an exact algorithm that decomposes the 2E-CVRP into a limited set of *multidepot capacitated vehicle routing problems* (MDCVRP) with side constraints. Extensive computational results on instances from the literature and on new instances show that the proposed method outperforms previous exact algorithms, both for the quality of the lower bounds achieved and the number and the size of the instances solved.

This paper is organized as follows. Section 2 describes the 2E-CVRP and the new mathematical formulation. The relaxations used to derive valid lower bounds are described in §3. Section 4 describes the bounding procedure based on the relaxations derived in §3. Section 5 presents the exact algorithm. Section 6 reports the computational results. Concluding remarks are given in §7.

2. Problem Description and Mathematical Formulation

An undirected graph $G = (N, E)$ is given, where the vertex set N is partitioned as $N = \{0\} \cup N_S \cup N_C$. Vertex 0 represents the depot, $N_S = \{1, 2, \ldots, n_s\}$ represents n_s satellites, and $N_C = \{n_s + 1, \ldots, n_s + n_c\}$ represents n_c customers. The edge set E is defined as $E = \{\{0, j\}: j \in N_S\} \cup \{\{i, j\}: i, j \in N_S \cup N_C,\ i < j\}$. A travel cost d_{ij} is associated with each edge $\{i, j\} \in E$. We assume that matrix d_{ij} satisfies the triangle inequality. Each customer $i \in N_C$ requires q_i units of goods from depot 0. We denote with $q_{\text{tot}} = \sum_{i \in N_C} q_i$ the sum of the customer demands.

A fleet of m^1 identical vehicles of capacity Q_1 are located at depot 0 and are used to transport goods to satellites. If used, a first-level vehicle incurs a fixed cost U_1 and performs a route passing through the depot 0 and a subset of satellites. The cost of a first-level route is the sum of the costs of the traversed edges plus the fixed cost U_1. Each satellite $k \in N_S$ can be visited by more than one first-level route and has a capacity B_k that limits the total customer demand that can be delivered to it by the first-level routes. Moreover, a fleet of m_k identical vehicles of capacity $Q_2 < Q_1$ are available at satellite $k \in N_S$ for servicing the customers. Nevertheless, at most $m^2 \leqslant \sum_{k \in N_S} m_k$ second-level vehicles can be globally used. If used, a second-level vehicle incurs a fixed cost U_2 and performs a route, that is a simple cycle in G passing through a satellite and a subset of customers and such that the total demand of the visited customers does not exceed the vehicle capacity Q_2. The cost of a second-level route is the sum of the traversed edges plus the fixed cost U_2. The handling cost at satellite $k \in N_S$ is given by H_k times the quantity delivered to satellite k.

The problem asks to design the vehicle routes of both levels so that each customer is visited exactly once, the quantity delivered to customers from each satellite is equal to the quantity received from the depot, and the sum of the routing and handling costs is minimized.

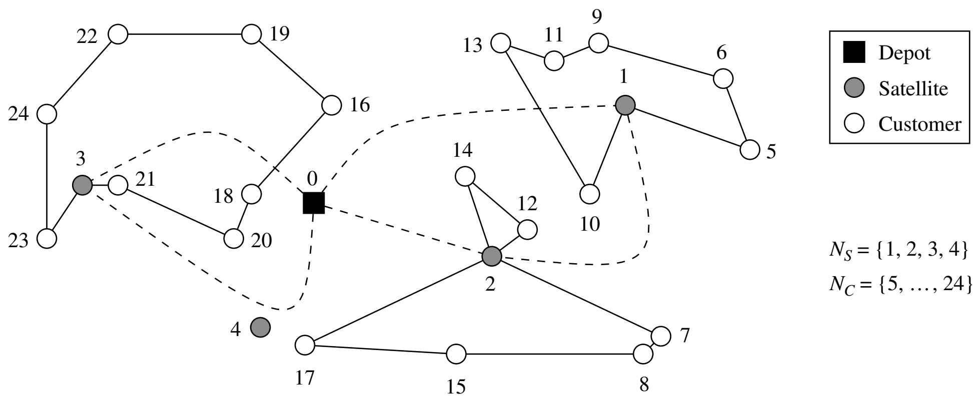
Figure 1 shows a solution to a 2E-CVRP instance with four satellites (the gray circles) and 20 customers (the white circles). Two first-level routes (the dashed lines) are routed at the depot (the black square); the two first-level routes visit one and two satellites, respectively. Satellite 4 is unused. Four second-level routes deliver goods to final customers.

To consider vehicle fixed costs U_1 and U_2, we assume that the travel cost matrix $[d_{ij}]$ is modified as follows: (i) for each satellite $k \in N_S$, cost $(1/2)U_1$ is added to d_{0k}, and (ii) for each satellite $k \in N_S$ and each customer $i \in N_C$, cost $(1/2)U_2$ is added to d_{ki}.

2.1. Formulation of the 2E-CVRP

The 2E-CVRP can be formulated as follows. Let $\mathcal{M}$ be the index set of all first-level routes, and let $\mathcal{M}_k \subseteq \mathcal{M}$ be the subset of first-level routes serving satellite $k \in N_S$. Let R_r and $E(R_r)$ be the subset of satellites visited and the subset of edges traversed by route $r \in \mathcal{M}$, respectively. The cost g_r of route $r \in \mathcal{M}$ is $g_r = \sum_{\{i, j\} \in E(R_r)} d_{ij}$. We assume that the route set $\mathcal{M}$ contains $\lceil \min\{m_k Q_2, q_{\text{tot}}\}/Q_1 \rceil$ copies of the single-satellite route $(0, k, 0)$, for each satellite $k \in N_S$. Let $w^{\min}$ and $w_r^{\max}$ be the minimum and maximum loads of first-level route $r \in \mathcal{M}$, computed as $w^{\min} = \max\{q_{\text{tot}} - (m^1 - 1)Q_1,\ 0\}$ and $w_r^{\max} = \min\{Q_1, q_{\text{tot}}, \sum_{k \in R_r} m_k Q_2\}$. Moreover, we denote by $W_r = \{w \in \mathbb{Z}_+ : w^{\min} \leqslant w \leqslant w_r^{\max}\}$ the set of possible loads of first-level route $r \in \mathcal{M}$. Because

Figure 1. A solution to the 2E-CVRP.



in real-world applications the number of satellites is small (say $n_s \leqslant 10$), in the following we assume that we can enumerate the $\mathcal{M}$.

Let $\mathcal{R}_k$ be the index set of the second-level routes passing through satellite $k \in N_S$, and let $\mathcal{R}_{ik} \subseteq \mathcal{R}_k$ be the subset of routes passing through satellite $k \in N_S$ and customer $i \in N_C$. We indicate with $\mathcal{R} = \bigcup_{k \in N_S} \mathcal{R}_k$ the set of all second-level routes and with π_l the satellite visited by route $l \in \mathcal{R}$. Moreover, we indicate with R_{kl} and $E(R_{kl})$ the subset of customers visited and the subset of edges traversed by route $l \in \mathcal{R}_k$, respectively. A load $w_{kl} = \sum_{i \in R_{kl}} q_i$ and a cost $c_{kl} = \sum_{\{i,j\} \in E(R_{kl})} d_{ij} + H_k w_{kl}$ are associated with route $l \in \mathcal{R}_k$.

Let y_r be a binary variable equal to one if and only if route $r \in \mathcal{M}$ is in solution, x_{kl} a binary variable equal to one if and only if route $l \in \mathcal{R}_k$ of satellite $k \in N_S$ is in solution, and q_{kr} a nonnegative integer variable representing the quantity delivered by first-level route $r \in \mathcal{M}$ to satellite $k \in R_r$ (we assume $q_{kr} = 0$, $k \in N_S \backslash R_r$). The 2E-CVRP can be formulated as follows:

$$(F) \qquad z(F) = \min \sum_{k \in N_S} \sum_{l \in \mathcal{R}_k} c_{kl} x_{kl} + \sum_{r \in \mathcal{M}} g_r y_r \tag{1}$$

$$\text{s.t.} \sum_{k \in N_S} \sum_{l \in \mathcal{R}_{ik}} x_{kl} = 1, \quad i \in N_C, \tag{2}$$

$$\sum_{l \in \mathcal{R}_k} x_{kl} \leqslant m_k, \quad k \in N_S, \tag{3}$$

$$\sum_{k \in N_S} \sum_{l \in \mathcal{R}_k} x_{kl} \leqslant m^2, \tag{4}$$

$$\sum_{l \in \mathcal{R}_k} w_{kl} x_{kl} \leqslant B_k, \quad k \in N_S, \tag{5}$$

$$\sum_{r \in \mathcal{M}} y_r \leqslant m^1, \tag{6}$$

$$\sum_{r \in \mathcal{M}_k} q_{kr} = \sum_{l \in \mathcal{R}_k} w_{kl} x_{kl}, \quad k \in N_S, \tag{7}$$

$$\sum_{k \in R_r} q_{kr} \leqslant Q_1 y_r, \quad r \in \mathcal{M}, \tag{8}$$

$$x_{kl} \in \{0, 1\}, \quad k \in N_S,\ l \in \mathcal{R}_k, \tag{9}$$

$$y_r \in \{0, 1\}, \quad r \in \mathcal{M}, \tag{10}$$

$$q_{kr} \in \mathbb{Z}_+, \quad k \in R_r,\ r \in \mathcal{M}. \tag{11}$$

The objective function (1) states to minimize the total cost. Constraints (2) specify that each customer $i \in N_C$ must be visited by exactly one second-level route. Constraints (3), (4), and (6) impose the upper bounds on the number of first and second-level routes in solution. Constraints (5) impose the satellite capacities. The balance between the quantity delivered by first-level routes to a satellite and the customer demands supplied from the satellite is imposed by constraints (7). Finally, constraints (8) impose that the vehicle capacity of the first-level vehicles is not exceeded.

To help the reader throughout the rest of the paper, we report a glossary of the symbols introduced so far in the e-companion to this paper (available as supplemental material at http://dx.doi.org/10.1287/opre.1120.1153; see Table EC.1).

2.2. The Special Case of the Location Routing Problem (LRP)

The 2E-CVRP contains the LRP as a special case. The LRP is defined on an undirected graph $G' = (N', E')$, where N' is partitioned as $N' = L \cup V$, where L represents possible depot locations and V a set of customers. A travel cost d_{ij} is associated with each edge $\{i, j\} \in E'$. A fixed cost C_k and a capacity B_k are associated with each depot location $k \in L$. Each customer $i \in V$ has associated a nonnegative demand q_i. An unlimited fleet of identical vehicles of capacity Q are available at the depots to supply the customers. If used, a vehicle incurs into a fixed cost U and performs a route passing through one of the depot locations and such that the total demand of the visited customers is at most Q. The cost of a route is the sum of the costs of the traversed edges plus the fixed cost U. The LRP consists of opening a set of depots and designing a set of routes for each open depot so that the total load of the routes operated from a depot $k \in L$ does not exceed its capacity B_k and each customer is visited by exactly one route. The objective is to minimize the sum of the cost of open depots and the costs of the routes.

Any LRP instance can be converted into an equivalent 2E-CVRP instance as follows:

(a) Define graph $G = (N, E)$ by setting $N_S = L$, $N_C = V$, and define the edge costs $d_{0k} = (1/2)C_k$, $k \in N_S$, and $d_{kj} = \infty$, $k, j \in N_S$, $k < j$.

(b) Define the first-level vehicle fleet by setting $m^1 = |N_S|$, $Q_1 = \infty$ and $U_1 = 0$.

(c) Define the second-level vehicle fleet by setting $m^2 = |N_C|$, $U_2 = U$, $Q_2 = Q$ and $m_k = |N_C|$, $k \in N_S$.

Any optimal solution of the resulting 2E-CVRP instance is also an optimal solution of the original LRP instance. Because $d_{kj} = \infty$, for any $k, j \in N_S$, $k < j$, each first-level vehicle route can only be a single-satellite route $(0, k, 0)$ of cost C_k, $k \in N_S$. Because $Q_1 = \infty$, any optimal solution contains at most a single-satellite route for each $k \in N_S$ representing the opening of depot k.

2.3. An Overview of the Proposed Exact Method

The exact method we propose consists of three main steps.

(1) We enumerate the set $\mathcal{M}$ of first-level routes and compute lower bound LD1 and upper bound UB1 on the 2E-CVRP (see §§3 and 4).

(2) We generate the set $\mathcal{P}$ of all possible subsets (*configurations*) of first-level routes that could be used in any optimal 2E-CVRP solution (see §5). Different bounding functions and dominance criteria are used to limit the size of the set $\mathcal{P}$.

(3) For each configuration $M \in \mathcal{P}$, we solve the corresponding problem F by fixing $y_r = 1$, $r \in M$, and $y_r = 0$, $r \in \mathcal{M}\backslash M$. This problem is a MDCVRP with side constraints (see §5), which is solved with an extension of the method of Baldacci and Mingozzi (2009). The optimal solution cost, $z(F)$, of the 2E-CVRP corresponds to the minimum solution cost of such MDCVRPs. We propose different bounding functions to solve to integrality only few MDCVRPs.

3. Relaxations of Formulation *F*

Let LF be the LP-relaxation of formulation F, and let $z(LF)$ be its optimal solution cost. Notice that, in any optimal LF solution, y_r is equal to $(\sum_{k\in R_r} q_{kr})/Q_1$, $r \in \mathcal{M}$. Thus, the higher the first-level routing cost, the worse the lower bound $z(LF)$. In the following, we describe an integer relaxation of the 2E-CVRP, called RF, that can provide a lower bound better than $z(LF)$.

3.1. Relaxation *RF*

This relaxation derives from problem F by relaxing, in a Lagrangean fashion, constraints (2)–(4) with penalties $\lambda_i \in \mathbb{R}$, $i \in N_C$, $\mu_k \in \mathbb{R}_-$, $k \in N_S$, and $\mu_0 \in \mathbb{R}_-$, respectively, and by defining the *marginal routing costs* β_{ik} for servicing customer $i \in N_C$ from satellite $k \in N_S$ as a solution of

$$\sum_{i\in N_C} a_{ikl}\beta_{ik} \leqslant c_{kl} - \sum_{i\in N_C} a_{ikl}\lambda_i - \mu_k - \mu_0, \quad l \in \mathcal{R}_k,\ k \in N_S, \tag{12}$$

where a_{ikl} is the number of times customer $i \in N_C$ is visited by route $l \in \mathcal{R}_k$ of satellite $k \in N_S$.

Problem RF involves binary variables ξ_{ik} equal to one if customer $i \in N_C$ is supplied from satellite $k \in N_S$ (0 otherwise) and variables y_r and q_{kr}, as defined for problem F. Relaxation RF is

$$(RF) \quad z(RF(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})) = \min \sum_{k\in N_S}\sum_{i\in N_C} \beta_{ik}\xi_{ik} + \sum_{r\in\mathcal{M}} g_r y_r + \sum_{i\in N_C} \lambda_i + \sum_{k\in N_S} m_k\mu_k + m^2\mu_0 \tag{13}$$

$$\text{s.t.} \quad \sum_{k\in N_S} \xi_{ik} = 1, \quad i \in N_C, \tag{14}$$

$$\sum_{r\in\mathcal{M}_k} q_{kr} = \sum_{i\in N_C} q_i\xi_{ik}, \quad k \in N_S, \tag{15}$$

$$\sum_{i\in N_C} q_i\xi_{ik} \leqslant B_k, \quad k \in N_S, \tag{16}$$

$$(6), (8), (10) \text{ and } (11), \tag{17}$$

$$\xi_{ik} \in \{0, 1\}, \quad i \in N_C,\ k \in N_S. \tag{18}$$

Theorem 1. *$z(RF(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$ is a valid lower bound on the* 2E-CVRP *for any solution* $\boldsymbol{\beta}$ *of inequalities* (12) *and any pair of penalty vectors* $\boldsymbol{\lambda} \in \mathbb{R}^{N_C}$ *and* $\boldsymbol{\mu} \in \mathbb{R}_-^{N_S+1}$.

Proof. The proof is provided in §EC.2 of the e-companion to this paper.

From Theorem 1, the following corollary follows.

Corollary 1. *Let* $z(\text{UB})$ *be a valid upper bound on the* 2E-CVRP. *For a given pair of vectors* $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ *and any solution* $\boldsymbol{\beta}$ *of inequalities* (12), *let*

$$\tilde{c}_{kl} = c_{kl} - \sum_{i\in N_C} a_{ikl}(\beta_{ik} + \lambda_i) - \mu_k - \mu_0$$

be the reduced cost of second-*level route* $l \in \mathcal{R}_k$, $k \in N_S$.

Any optimal 2E-CVRP *solution of cost smaller than* $z(\text{UB})$ *cannot contain any* second-*level route* $l \in \mathcal{R}_k$, $k \in N_S$, *of reduced cost* $\tilde{c}_{kl} \geqslant z(\text{UB}) - z(RF(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$.

Proof. The proof is provided in §EC.2 of the e-companion to this paper.

Theorem 2. *The relation* $\max_{\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}}\{z(RF(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))\} \geqslant z(LF)$ *holds, and such inequality can be strict.*

Proof. The proof is provided in §EC.2 of the e-companion to this paper.

3.2. Relaxation $\overline{RF}$

Solving RF to optimality can be time consuming, so we describe a further relaxation of RF, called $\overline{RF}$, that can be efficiently solved. Let ϕ_{rw} be a lower bound on the cost for delivering a demand $w \in W_r$ to customers from the subsets of satellites R_r visited by first-level route $r \in \mathcal{M}$. For a given vector $\boldsymbol{\beta}$ satisfying inequalities (12), the value ϕ_{rw} is defined as the optimal solution cost of the following continuous knapsack problem, called $KP(r, w)$:

$$(KP(r,w)) \qquad \phi_{rw} = \min \sum_{i \in N_C} \min_{k \in R_r} \{\beta_{ik}\} z_i$$
$$\text{s.t.} \sum_{i \in N_C} q_i z_i = w,$$
$$0 \leqslant z_i \leqslant 1, \quad i \in N_C.$$

For each pair (r, w), we denote by $z_i^*(r, w)$, $i \in N_C$, the optimal solution of problem $KP(r, w)$, and we define $V(r, w) = \{i \in N_C: z_i^*(r, w) > 0\}$. Let ζ_{rw}, $r \in \mathcal{M}$, $w \in W_r$, be a binary variable equal to one if and only if first-level route r delivering w units of goods is in solution. Problem $\overline{RF}$ is defined as

$$(\overline{RF}) \quad z\big(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})\big)$$
$$= \min \sum_{r \in \mathcal{M}} \sum_{w \in W_r} (g_r + \phi_{rw}) \zeta_{rw} + \sum_{i \in N_C} \lambda_i + \sum_{k \in N_S} m_k \mu_k + m^2 \mu_0 \qquad (19)$$
$$\text{s.t.} \sum_{r \in \mathcal{M}} \sum_{w \in W_r} w \zeta_{rw} = q_{\text{tot}}, \qquad (20)$$
$$\sum_{w \in W_r} \zeta_{rw} \leqslant 1, \quad r \in \mathcal{M}, \qquad (21)$$
$$\zeta_{rw} \in \{0, 1\}, \quad r \in \mathcal{M},\ w \in W_r. \qquad (22)$$

Problem $\overline{RF}$ is a *multiple-choice knapsack problem* and can be conveniently solved by DP. The following theorem shows that $\overline{RF}$ is a relaxation of problem RF.

THEOREM 3. $z\big(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})\big) \leqslant z\big(RF(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})\big)$ *for any solution* $\boldsymbol{\beta}$ *of inequalities* (12) *and for any penalty vectors* $\boldsymbol{\lambda}$ *and* $\boldsymbol{\mu}$.

PROOF. The proof is provided in §EC.2 of the e-companion to this paper.

Because of Theorem 3 and Corollary 1, any optimal 2E-CVRP solution of cost smaller than a known upper bound $z(\text{UB})$ cannot contain any second-level route $l \in \mathcal{R}_k$, $k \in N_S$, of reduced cost $\tilde{c}_{kl} \geqslant z(\text{UB}) - z\big(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})\big)$, where $\tilde{c}_{kl}$ is defined as above.

A valid lower bound LD1 on the 2E-CVRP can be computed as the cost of a near-optimal solution of problem

$$\text{LD1} = \max_{\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}} \{z\big(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})\big)\}. \qquad (23)$$

4. Lower Bound LD1 and Bounding Procedure DP[1]

In this section, we describe a bounding procedure, called DP[1], and a heuristic algorithm based on relaxation $\overline{RF}$ to compute lower and upper bounds LD1 and UB1, respectively. Bounding procedure DP[1] finds a near-optimal solution of problem (23) and uses a DP algorithm to solve problem $\overline{RF}$.

Bounding procedure DP[1] is based on a relaxation of the 2E-CVRP, where the second-level route sets $\mathcal{R}_k$, $k \in N_S$, are enlarged to also contain nonnecessarily elementary routes. The method used by DP[1] to find a feasible solution of inequalities (12) is based on the following theorem.

THEOREM 4. *Let us associate penalties* $\lambda_i \in \mathbb{R}$, $i \in N_C$, *with constraints* (2), $\mu_k \in \mathbb{R}_-$, $k \in N_S$, *with constraints* (3), *and* $\mu_0 \in \mathbb{R}_-$ *with constraint* (4). *Let* $\hat{\mathcal{R}}_k \supseteq \mathcal{R}_k$ *be the index set of nonnecessarily elementary routes for satellite k. A feasible solution* β_{ik} *of inequalities* (12) *is given by*

$$\beta_{ik} = q_i \min_{l \in \hat{\mathcal{R}}_{ik}} \left\{ \frac{c_{kl} - \sum_{i \in N_C} a_{ikl} \lambda_i - \mu_k - \mu_0}{\sum_{i \in N_C} a_{ikl} q_i} \right\}, \quad i \in N_C,\ k \in N_S. \qquad (24)$$

PROOF. The proof is provided in §EC.2 of the e-companion to this paper.

In procedure DP[1], the route set $\hat{\mathcal{R}}_k$ is defined as the set of ng-routes introduced by Baldacci et al. (2011a) that are shortly described below.

Let $N_i \subseteq N_C$, $i \in N_C$, be a set of selected customers for customer i (according to some criterion), such that $N_i \ni i$ and $|N_i| \leqslant \Delta(N_i)$, where $\Delta(N_i)$ is a parameter. The sets N_i allow us to associate with each path $P = (k, i_1, \ldots, i_t)$ that starts from satellite $k \in N_S$, visits vertices $i_1, \ldots, i_t \in N_C$, and ends at vertex i_t, the subset $\Pi(P)$ containing i_t and every customer i_s, $s = 1, \ldots, t-1$, of P that belongs to all sets $N_{s+1}, \ldots, N_{i_t}$ associated with the customers $i_{s+1}, \ldots, i_t$ visited after i_s. The set $\Pi(P)$ is defined as $\Pi(P) = \{i_s\colon\ i_s \in \bigcap_{j=s+1}^{t} N_{i_j},\ s = 1, \ldots, t-1\} \cup \{i_t\}$. A forward ng-path (NG, k, q, i) is a nonnecessarily elementary path $P = (k, i_1, \ldots, i_{t-1}, i_t = i)$ that starts from satellite $k \in N_S$, ends at customer i, visits a subset of customers of total demand equal to q, and such that $NG = \Pi(P)$ and $i \notin \Pi(P')$, where $P' = (k, i_1, \ldots, i_{t-1})$. An (NG, k, q, i)-route (or simply ng-route) is obtained by adding edge $\{i, k\}$ to an ng-path (NG, k, q, i).

Algorithm DP[1] uses column generation to solve Equations (24) and subgradient optimization to solve problem (23).

4.1. Description of Procedure DP[1]

To solve Equations (24), procedure DP[1] uses a limited set $\bar{\mathcal{R}}_k \subseteq \hat{\mathcal{R}}_k$, $k \in N_S$, of ng-routes. Procedure DP[1] initializes each set $\bar{\mathcal{R}}_k$ with all single-customer routes (k, i, k), $i \in N_C$, and sets $\boldsymbol{\lambda} = \mathbf{0}$, $\boldsymbol{\mu} = \mathbf{0}$, LD1 = 0 and UB1 = ∞. Bounding procedure DP[1] executes an a priori defined number ($Maxit1$) of macro iterations where, at each macro iteration, the following steps are performed.

(1) Initialize $z^* = 0$, and perform $Maxit2$ iterations of the following steps:

(i) Compute values β_{ik}, $i \in N_C$, $k \in N_S$, through expression (24), where each $\hat{\mathcal{R}}_k$ is replaced with $\bar{\mathcal{R}}_k$, $k \in N_S$.

(ii) Solve $\overline{RF}$ using values β_{ik} as described in §4.2. If $z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})) > z^*$, then update $z^* = z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$, $\boldsymbol{\beta}^* = \boldsymbol{\beta}$, $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}$ and $\boldsymbol{\mu}^* = \boldsymbol{\mu}$.

(iii) Update penalty vector $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ as described in §4.3.

(2) Generate a set of ng-routes $\mathcal{N}_k \subseteq \hat{\mathcal{R}}_k \setminus \bar{\mathcal{R}}_k$, $k \in N_S$, for which inequalities (12) are violated by $\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, and $\boldsymbol{\mu}^*$ as described in §4.4. There are two cases:

(i) $\mathcal{N}_k = \varnothing$, for each $k \in N_S$. If LD1 $< z^*$, then update LD1 $= z^*$, $\boldsymbol{\beta}^1 = \boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^1 = \boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^1 = \boldsymbol{\mu}^*$, and execute the heuristic algorithm described in §4.5 producing upper bound $z(\text{UB})$. Update UB1 $= \min\{\text{UB1}, z(\text{UB})\}$.

(ii) $\mathcal{N}_k \neq \varnothing$, for some $k \in N_S$, then update $\bar{\mathcal{R}}_k = \bar{\mathcal{R}}_k \cup \mathcal{N}_k$.

Notice that $\boldsymbol{\beta}^1$, $\boldsymbol{\lambda}^1$, and $\boldsymbol{\mu}^1$ are the vectors producing lower bound LD1 in problem (23).

4.2. Solving Problem $\overline{RF}$

Problem $\overline{RF}$ can be solved by DP as follows. Let $h(r, w)$ be the optimal solution cost of $\overline{RF}$ obtained by using the first-level routes $1, \ldots, r$, $0 \leqslant r \leqslant |\mathcal{M}|$, and replacing q_{tot} in Equation (20) with $w \in \mathbb{Z}_+$, $w^{\min} \leqslant w \leqslant q_{\text{tot}}$. The DP recursion for computing functions $h(r, w)$, $r = 1, \ldots, |\mathcal{M}|$, $w^{\min} \leqslant w \leqslant q_{\text{tot}}$, is

$$h(r, w) = \min\left\{ h(r-1, w), \min_{w^{\min} \leqslant w' \leqslant \min\{w, w_r^{\max}\}} \{h(r-1, w-w') + g_r + \phi_{rw'}\} \right\}. \quad (25)$$

The recursion is initialized by setting $h(r, 0) = 0$, $r = 0, \ldots, |\mathcal{M}|$, and $h(0, w) = \infty$, $w = 1, \ldots, q_{\text{tot}}$. The $\overline{RF}$ optimal solution cost is $z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu})) = h(|\mathcal{M}|, q_{\text{tot}})$.

Let $\bar{z}$ be an upper bound on $z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$. The number of states (r, w) to generate in order to compute $z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$ can be reduced by using bounding functions $\text{lb}(r, w)$, described below, to eliminate any state (r, w) that cannot lead to any $\overline{RF}$ solution of cost smaller than $\bar{z}$.

We denote by $\text{lb}(r, w)$ a lower bound on the optimal solution cost of problem $\overline{RF}$, where $\mathcal{M}$ is replaced with the subset $\{r, r+1, \ldots, |\mathcal{M}|\}$ and q_{tot} with w. Let $\alpha_r = \min_{w^{\min} \leqslant w \leqslant w_r^{\max}} \{(g_r + \phi_{rw})/w\}$, $r \in \mathcal{M}$. By assuming that the routes in $\mathcal{M}$ are indexed so that $\alpha_1 \leqslant \alpha_2 \leqslant \cdots \leqslant \alpha_{|\mathcal{M}|}$, functions $\text{lb}(r, w)$ can be computed using the following backward recursion.

Initialize (i) $\text{lb}(r, 0) = 0$, $r = 1, \ldots, |\mathcal{M}|$; (ii) $\text{lb}(r, w) = \infty$, $0 < w < w^{\min}$, $r = 1, \ldots, |\mathcal{M}|$; (iii) $\text{lb}(|\mathcal{M}|, w) = w\alpha_{|\mathcal{M}|}$, $w \in W_{|\mathcal{M}|}$; and (iv) $\text{lb}(|\mathcal{M}|, w) = \infty$, $w_r^{\max} < w \leqslant q_{\text{tot}}$.

For each $r = |\mathcal{M}| - 1, |\mathcal{M}| - 2, \ldots, 1$ and $w^{\min} \leqslant w \leqslant q_{\text{tot}}$ compute

$$\text{lb}(r, w) = \begin{cases} w\alpha_r & \text{if } w \leqslant w_r^{\max} \\ w_r^{\max}\alpha_r + \text{lb}(r+1, w - w_r^{\max}) & \text{if } w_r^{\max} + 1 \leqslant w \leqslant q_{\text{tot}}. \end{cases}$$

Thus, a state (r, w), $r < |\mathcal{M}|$, is fathomed if $h(r, w) + \text{lb}(r+1, q_{\text{tot}} - w) \geqslant \bar{z}$.

In performing recursion (25), the upper bound $\bar{z}$ is initialized as $\bar{z} = \text{UB1}$ and dynamically updated, at the end of stage r, as $\bar{z} = \min\{\bar{z}, h(r, q_{\text{tot}})\}$.

4.3. Computing a Subgradient

Usual backtracking can be used to derive the $\overline{RF}$ solution $\boldsymbol{\zeta}$ of cost $h(|\mathcal{M}|, q_{\text{tot}})$. Given $\boldsymbol{\zeta}$ and the sets $V(r, w)$, as defined in §3.2, associated to ϕ_{rw}, we derive the index sets $\tilde{\mathcal{R}}_k \subseteq \bar{\mathcal{R}}_k$, $k \in N_S$, of the second-level routes in solution and the index $l(i, k)$ of the route in $\tilde{\mathcal{R}}_k$ associated with β_{ik} as follows:

(i) Initialize $\tilde{\mathcal{R}}_k = \varnothing$, $k \in N_S$, and $l(i, k) = 0$, $i \in N_C$, $k \in N_S$.

(ii) Repeat the following steps for each route $r \in \mathcal{M}$ such that $\zeta_{rw} = 1$ for some load $w \in W_r$:

(a) Compute $\bar{k}(i) = \arg\min_{k \in R_r} \{\beta_{ik}\}$, $i \in V(r, w)$. Let $l(i, \bar{k}(i))$, $i \in V(r, w)$, be the index of the route in $\hat{\mathcal{R}}_k$ associated with $\beta_{i\bar{k}(i)}$ in expressions (24).

(b) For each $i \in V(r, w)$, set $\tilde{\mathcal{R}}_{\bar{k}(i)} = \tilde{\mathcal{R}}_{\bar{k}(i)} \cup \{l(i, \bar{k}(i))\}$.

A subgradient to function $z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))$, at point $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, can be computed as follows. Let $\tilde{\mathbf{x}}$ be a vector whose components are computed as

$$\tilde{x}_{kl} = \sum_{i \in N_C:\, l(i,k) = l} \frac{a_{ikl} q_i}{\sum_{i \in N_C} a_{ikl} q_i}, \qquad l \in \tilde{\mathcal{R}}_k,\ k \in N_S.$$

Let

$$\alpha_i = \sum_{k \in N_S} \sum_{l \in \tilde{\mathcal{R}}_k} a_{ikl} \tilde{x}_{kl}, \quad i \in N_C,$$

$$\delta_k = \sum_{l \in \tilde{\mathcal{R}}_k} \tilde{x}_{kl}, \quad k \in N_S, \qquad \text{and} \qquad \delta_0 = \sum_{k \in N_S} \delta_k.$$

Then, penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are modified as $\lambda_i = \lambda_i - \epsilon\gamma(\alpha_i - 1)$, $i \in N_C$, $\mu_k = \min\{0, \mu_k - \epsilon\gamma(\delta_k - m_k)\}$, $k \in N_S$, and $\mu_0 = \min\{0, \mu_0 - \epsilon\gamma(\delta_0 - m^2)\}$, where ϵ is a positive constant and

$$\gamma = \frac{0.2 z(\overline{RF}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\mu}))}{\sum_{i \in N_C} (\alpha_i - 1)^2 + \sum_{k \in N_S} (\delta_k - m_k)^2 + (\delta_0 - m^2)^2}.$$

4.4. Generating the *ng*-Route Set $\mathcal{N}_k$ for a Given Satellite k

We describe the procedure to generate, for a given satellite $k \in N_S$, the set of ng-routes $\mathcal{N}_k \subseteq \hat{\mathcal{R}}_k$ that violate inequalities (12) for given vectors $\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, and $\boldsymbol{\mu}^*$, when $\mathcal{R}_k$ is replaced by $\hat{\mathcal{R}}_k$.

Define the modified edge costs $\bar{d}_{ij} = d_{ij} - (1/2)(\beta_{ik}^* + \lambda_i^* - H_k q_i) - (1/2)(\beta_{jk}^* + \lambda_j^* - H_k q_j)$ and the sets $N_i \subseteq N_C$, $i \in N_C$, to contain the $\Delta(N_i)$ nearest customers to i according to $\bar{d}_{ij}$.

Let $f(NG, k, q, i)$ be the cost of a least-cost *ng*-path (NG, k, q, i) using the modified edge cost $\bar{d}_{ij}$. Functions $f(NG, k, q, i)$ are computed using the *DP* recursions described by Baldacci et al. (2011a) on the state-space graph $\mathcal{H} = (\mathcal{V}, \Psi)$, defined for a given satellite $k \in N_S$, as

$$\mathcal{V} = \Big\{(NG, k, q, i)\colon q_i \leqslant q \leqslant Q_2, \quad \forall\, NG \subseteq N_i \text{ s.t. } NG \ni i \text{ and } \sum_{j\in NG} q_j \leqslant q,\ \forall\, i \in N_C\Big\},$$

$$\Psi = \big\{((NG', k, q', j), (NG, k, q, i))\colon \forall\,(NG', k, q', j) \in \Psi^{-1}(NG, k, q, i),\ \forall\,(NG, k, q, i) \in \mathcal{E}\big\},$$

where $\Psi^{-1}(NG, k, q, i) = \{(NG', k, q - q_i, j)\colon \forall NG' \subseteq N_j$ s.t. $NG' \ni j$ and $NG' \cap N_i = NG\backslash\{i\}$, $j \in N_C$ s.t. $\{i, j\} \in E$ if $i < j$ or $\{j, i\} \in E$ if $j < i\}$.

Let

$$r(i, k) = \min_{(NG, k, q, i)\in\mathcal{V}} \big\{f(NG, k, q, i) - \mu_k^* - \mu_0^* + \bar{d}_{ik}\big\}$$

be the cost of a least-cost *ng*-route visiting $i \in N_C$ immediately before arriving at satellite k. The route set $\mathcal{N}_k$ contains the *ng*-routes corresponding to $r(i, k) < 0$, $i \in N_C$.

4.5. A Lagrangean Heuristic

Procedure DP[1] is interwoven with a heuristic algorithm that produces a feasible 2E-CVRP solution of cost $z(\text{UB})$ using the second-level route sets $\tilde{\mathcal{R}}_k$, $k \in N_S$, and vector $\tilde{\mathbf{x}}$ associated with an $\overline{RF}$ solution (see §4.3). First, the routes in $\tilde{\mathcal{R}}_k$, $k \in N_S$, are modified with the objective of obtaining a solution vector $\mathbf{x}$ satisfying constraints (2)–(5). Then, the solution vector $\mathbf{x}$ is used to derive solution vectors $\mathbf{y}$ and $\mathbf{q}$ such that $(\mathbf{x}, \mathbf{y}, \mathbf{q})$ represents a feasible 2E-CVRP solution.

Description of the Heuristic Algorithm

(1) [*Initialization*]. Let $\tilde{\mathcal{R}} = \bigcup_{k\in N_S} \tilde{\mathcal{R}}_k$. Initialize $SOL = \varnothing$ and $\delta(i) = 0$, $i \in N_C$.

(2) [*Extract a subset of routes* $SOL \subseteq \tilde{\mathcal{R}}$]. Let l^* be the route of $\tilde{\mathcal{R}}$, where $\tilde{x}_{\pi_{l^*} l^*} = \max\{\tilde{x}_{\pi_l l}\colon l \in \tilde{\mathcal{R}}\}$. Remove l^* from $\tilde{\mathcal{R}}$. If $\delta(i) = 0$, for some $i \in R_{\pi_{l^*} l^*}$, then update $SOL = SOL \cup \{l^*\}$ and $\delta(i) = \delta(i) + 1$, $i \in R_{\pi_{l^*} l^*}$. Repeat Step 1 until $\tilde{\mathcal{R}} = \varnothing$.

(3) [*Modify the route set SOL*]. Remove from *SOL* any route $l \in SOL$ such that $\delta(i) > 1$, $i \in R_{\pi_l l}$ and update $\delta(i) = \delta(i) - 1$. For each $l \in SOL$, compute the savings that can be achieved by removing from route l every customer $i \in R_{\pi_l l}$ having $\delta(i) > 1$. Let $l^* \in SOL$ be the route of maximum saving. Remove from route l^* every customer $i \in R_{\pi_{l^*} l^*}$ with $\delta(i) > 1$, and update $\delta(i) = \delta(i) - 1$. Repeat Step 3 until $\delta(i) \leqslant 1$, for each $i \in N_C$.

(4) [*Insert unrouted customers*]. For each unrouted customer i (i.e., $\delta(i) = 0$) perform the following operations. Compute the minimum extra-mileage $exm(i, l)$ for inserting i in route $l \in SOL$. We set $exm(i, l) = \infty$ if the total load of the resulting route l exceeds the vehicle capacity Q_2. Let l^* be such that $exm(i, l^*) = \min_{l\in SOL}[exm(i, l)]$. If $exm(i, l^*) = \infty$, then set $z(\text{UB}) = \infty$ and stop; otherwise, insert customer i in route l^* in the position of cost $exm(i, l^*)$ and set $\delta(i) = 1$.

(5) [*Define the F solution* $\mathbf{x}$]. Define $x_{\pi_l l} = 1$, for each $l \in SOL$, and $x_{\pi_l l} = 0$, for each $l \in \mathcal{R}\backslash SOL$. If $\mathbf{x}$ does not satisfy constraints (3)–(5), then set $z(\text{UB}) = \infty$ and the algorithm terminates.

(6) [*Improve the cost of the routes in SOL*]. The post-optimization procedure for improving the total cost of the routes in *SOL* applies the following procedure in the order specified below.

(a) [*Exchange of one customer between two routes of SOL*]. For each customer $i \in N_C$, compute the saving $move(i, l)$ achieved by removing i from its current route l_i and inserting i in the least-cost position of route $l \in SOL$. Set $move(i, l_i) = 0$ and $move(i, l) = -\infty$ if the load of the resulting route l violates constraint (5) for satellite π_l or if customer i cannot be inserted in route l without violating the vehicle capacity Q_2. Let i^* and l^* be determined such that $move(i^*, l^*) = \max[move(i, l)\colon i \in N_C,\ l \in SOL]$. If $move(i^*, l^*) > 0$, then remove customer i^* from its current route and insert it in the best position of route l^*. This procedure is repeated until $move(i^*, l^*) \leqslant 0$.

(b) [*Exchange of two customers between two routes of SOL*]. For all pairs of routes $l, l' \in SOL$ and for each pair of customers $i \in R_{\pi_l l}$ and $j \in R_{\pi_{l'} l'}$, compute the saving $sav(i, j)$ obtained by moving customer i from route l to route l' and customer j from route l' to route l. We set $sav(i, j) = -\infty$ if the exchange violates constraint (5) for one of the two satellites π_l, $\pi_{l'}$ or if the total load of one of the two routes exceeds the vehicle capacity Q_2. The two customers i^* and j^* producing the maximum saving are then exchanged if $sav(i^*, j^*) > 0$. This procedure is repeated until $sav(i^*, j^*) \leqslant 0$. Whenever this procedure improves the solution, then the post-optimization routing is restarted from the beginning.

(c) Optimize each route $l \in SOL$ using a three-optimal method.

(7) [*Constructing a feasible* 2E-CVRP *solution*]. Let $\omega_k = \sum_{l\in\mathcal{R}_k} w_{kl} x_{kl}$, $k \in N_S$, be the total demand associated with satellite k by the solution vector $\mathbf{x}$ defined above. We solve to optimality the following problem $F(\mathbf{x})$ with an integer programming solver:

$$(F(\mathbf{x})) \qquad z\big(F(\mathbf{x})\big) = \min \sum_{r\in\mathcal{M}} g_r y_r$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{M}} y_r \leqslant m^1,$$

$$\sum_{r\in\mathcal{M}_k} q_{kr} = \omega_k, \quad k \in N_S,$$

$$\sum_{k\in R_r} q_{kr} \leqslant Q_1 y_r, \quad r \in \mathcal{M},$$

$$y_r \in \{0, 1\}, \quad r \in \mathcal{M},$$

$$q_{kr} \geqslant 0, \quad k \in R_r,\ r \in \mathcal{M}.$$

Let $(\mathbf{y}, \mathbf{q})$ be the optimal $F(\mathbf{x})$ solution (we assume $z(F(\mathbf{x})) = \infty$ if problem $F(\mathbf{x})$ does not admit a feasible solution). If problem $F(\mathbf{x})$ admits a feasible solution, then the vectors $(\mathbf{x}, \mathbf{y}, \mathbf{q})$ represent a feasible 2E-CVRP solution of cost $z(\text{UB}) = z(F(\mathbf{x})) + \sum_{k\in N_S}\sum_{l\in\mathcal{R}_k} c_{kl}x_{kl}$.

5. An Exact Method for Solving the 2E-CVRP

The method for solving the 2E-CVRP is based on the following reformulation of problem F.

Let $\mathcal{P} = \{M \subseteq \mathcal{M}\colon |M|Q_1 \geqslant q_{\text{tot}}, |M| \leqslant m^1\}$. We call *configuration* each element M of the set $\mathcal{P}$. For each configuration $M \in \mathcal{P}$, let $N_S(M) = \bigcup_{r\in M} R_r$, $M_k = M \cap \mathcal{M}_k$, $k \in N_S$, and $U(M) = \sum_{r\in M} g_r$. An optimal 2E-CVRP solution can be computed as

$$z(F) = \min_{M\in\mathcal{P}}\{U(M) + z(F(M))\}, \tag{26}$$

where $z(F(M))$ is the optimal solution cost of the following problem $F(M)$:

$$(F(M)) \quad z(F(M)) = \min \sum_{k\in N_S(M)}\sum_{l\in\mathcal{R}_k} c_{kl}x_{kl}$$

$$\text{s.t.} \quad \sum_{k\in N_S(M)}\sum_{l\in\mathcal{R}_{ik}} x_{kl} = 1, \quad i \in N_C,$$

$$\sum_{l\in\mathcal{R}_k} x_{kl} \leqslant m_k, \quad k \in N_S(M),$$

$$\sum_{k\in N_S(M)}\sum_{l\in\mathcal{R}_k} x_{kl} \leqslant m^2,$$

$$\sum_{l\in\mathcal{R}_k} w_{kl}x_{kl} \leqslant B_k, \quad k \in N_S(M),$$

$$\sum_{r\in M_k} q_{kr} = \sum_{l\in\mathcal{R}_k} w_{kl}x_{kl}, \quad k \in N_S(M),$$

$$\sum_{k\in R_r} q_{kr} \leqslant Q_1, \quad r \in M,$$

$$x_{kl} \in \{0,1\}, \quad k \in N_S(M), l \in \mathcal{R}_k,$$

$$q_{kr} \geqslant 0, \quad k \in R_r, r \in M.$$

We assume $z(F(M)) = \infty$ if $F(M)$ has no feasible solution for configuration $M \in \mathcal{P}$.

Problem $F(M)$ is an extension of the multidepot vehicle routing problem considered by Baldacci and Mingozzi (2009). The methods for generating the set $\mathcal{P}$, solving problems $F(M)$ and (26) are described in §§5.1–5.3, respectively.

5.1. Generating the Set of Configurations $\mathcal{P}$

The generation of the set $\mathcal{P}$ of configurations is based on the following propositions.

Let LB_R be a lower bound on the second-level routing cost of any optimal solution computed as

$$\text{LB}_R = \sum_{i\in N_C}\min_{k\in N_S}\{\beta^1_{ik}\} + \sum_{i\in N_C}\lambda^1_i + \sum_{k\in N_S} m_k\mu^1_k + m^2\mu^1_0,$$

and let $\text{LBW}(M)$ be a lower bound on $z(F(M))$ computed as

$$\text{LBW}(M) = \sum_{i\in N_C}\min_{k\in N_S(M)}\{\beta^1_{ik}\} + \sum_{i\in N_C}\lambda^1_i + \sum_{k\in N_S(M)} m_k\mu^1_k + m^2\mu^1_0.$$

Proposition 1. *Let* $z(\text{UB})$ *be a valid upper bound on the* 2E-CVRP. *A configuration* $M \in \mathcal{P}$ *can belong to an optimal* 2E-CVRP *solution if and only if it satisfies the following conditions*:

$$\left.\begin{array}{ll} |R_r \cap R_{r'}| \leqslant 1, \quad r, r' \in M,\ r \neq r', & \text{(a)}\\ \sum_{r\in M}\min\Big\{Q_1, \sum_{k\in R_r} m_kQ_2\Big\} \geqslant q_{\text{tot}}, & \text{(b)}\\ \sum_{r\in M}\sum_{k\in R_r} m_k \geqslant \lceil q_{\text{tot}}/Q_2\rceil, & \text{(c)}\\ U(M) < z(\text{UB}) - \text{LB}_R, & \text{(d)}\\ U(M) < z(\text{UB}) - \text{LBW}(M). & \text{(e)} \end{array}\right\}$$

Condition (a) is a property of the feasible solutions of the *split delivery vehicle routing problem* (see Dror and Trudeau 1990). Conditions (b) and (c) are feasibility conditions. Conditions (d) and (e) follow from the properties of any optimal 2E-CVRP solution of cost less than $z(\text{UB})$.

Proposition 2. *For a given configuration* $M \in \mathcal{P}$, *let* $\theta(k)$, $k \in N_S(M)$, *be a lower bound on the quantity that must be supplied to satellite* k *in any feasible* $F(M)$ *solution by the* first-*level routes* $M_k \subseteq M$ *passing through satellite* k. *Problem* $F(M)$ *has no feasible solution if either* $\sum_{k\in N_S(M)}\lceil\theta(k)/Q_2\rceil > m^2$ *or* $\lceil\theta(k)/Q_2\rceil > m_k$, *for some* $k \in N_S$. *If so,* M *can be removed from* $\mathcal{P}$.

Lower bound $\theta(k)$ can be computed as the optimal solution cost of the following problem:

$$\theta(k) = \min \sum_{r\in M_k} q_{kr} \tag{27}$$

$$\text{s.t.} \quad \sum_{h\in R_r} q_{hr} \leqslant Q_1, \quad r \in M, \tag{28}$$

$$\sum_{r\in M}\sum_{h\in R_r} q_{hr} = q_{\text{tot}}, \tag{29}$$

$$\sum_{r\in M_h} q_{hr} \geqslant q^{\min}, \quad h \in N_S(M), \tag{30}$$

$$\sum_{r\in M_h} q_{hr} \leqslant m_kQ_2, \quad h \in N_S(M), \tag{31}$$

$$q_{hr} \geqslant 1, \quad h \in R_r,\ r \in M, \tag{32}$$

where

$$q^{\min} = \max\Big\{\min_{i\in N_C}\{q_i\}, q_{\text{tot}} - (m^2 - 1)Q_2\Big\}.$$

We assume $\theta(k) = \infty$ if problem (27)–(32) has no feasible solution. The set $\mathcal{P}$ is generated by pure enumeration by using Propositions 1 and 2 to eliminate any configuration M that cannot lead to an optimal 2E-CVRP solution.

5.2. Solving Problem $F(M)$

Problem $F(M)$ is solved with the following three-phase method. In the *first phase*, bounding procedure DP^1 is used to compute lower bound $\text{LD1}(M)$ on $z(F(M))$ by replacing $\mathcal{M}$ with M. In the *second phase*, a near-optimal dual solution of the LP-relaxation of $F(M)$ strengthened by valid inequalities, called problem $\bar{F}(M)$, is computed. In the *third phase*, the $\bar{F}(M)$ dual solution is used to generate the subsets $\mathcal{R}'_k \subseteq \mathcal{R}_k$, $k \in N_S(M)$, of all second-level routes of any $F(M)$ optimal solution. An $F(M)$ optimal solution is obtained by replacing, in $F(M)$, each set $\mathcal{R}_k$ with $\mathcal{R}'_k$, $k \in N_S(M)$, and solving the resulting problem, called $F'(M)$, with an integer programming solver.

5.2.1. Phase 1: Computing Lower Bound $\text{LD1}(M)$. We execute procedure DP^1, by replacing the set $\mathcal{M}$ with M, to compute lower bound $\text{LD1}(M)$ and upper bound $\text{UB1}(M)$ on $F(M)$. If $\text{LD1}(M)$ is greater than a known upper bound on the 2E-CVRP, Phases 2 and 3 are skipped.

5.2.2. Phase 2: Solving $\bar{F}(M)$. Problem $\bar{F}(M)$ corresponds to the LP-relaxation of problem $F(M)$ strengthened with the following valid inequalities:

(a) *Capacity constraints*. Let $\mathcal{S} = \{H\colon\ H \subseteq N_C,\ |H| \geqslant 2\}$. The capacity constraints are

$$\sum_{k\in N_S(M)} \sum_{l\in\mathcal{R}_k:\, R_{kl}\cap H\neq\varnothing} x_{kl} \geqslant \left\lceil \frac{\sum_{i\in H} q_i}{Q_2} \right\rceil, \quad H \in \mathcal{S}. \tag{33}$$

(b) *Clique inequalities*. Let $\mathcal{R}(M) = \bigcup_{k\in N_S(M)} \mathcal{R}_k$, and let $\mathcal{G} = (\mathcal{R}(M), \mathcal{E})$ be the conflict graph associated with the route set $\mathcal{R}(M)$, where the edge set $\mathcal{E}$ contains every edge $\{l, l'\}$, $l, l' \in \mathcal{R}(M)$, such that $l < l'$ and $R_{\pi_l l} \cap R_{\pi_{l'} l'} \neq \varnothing$. Let $\mathscr{C}$ be the set of all cliques of graph $\mathcal{G}$. The clique inequalities are

$$\sum_{l\in C} x_{\pi_l l} \leqslant 1, \quad C \in \mathcal{C}. \tag{34}$$

Problem $\bar{F}(M)$ is solved with a column-and-cut generation procedure that starts by setting $\mathcal{S} = \varnothing$, $\mathcal{C} = \varnothing$. The master problem is initialized with a set of elementary routes obtained from the final set of *ng*-routes generated in Phase 1 for computing lower bound $\text{LD1}(M)$ by removing, from each *ng*-route, the customers visited more than once. At each iteration, a set of negative reduced cost routes are generated and a set of violated inequalities (33) and (34) are added as described in Baldacci and Mingozzi (2009). The procedure ends when no negative reduced cost routes exist and no inequalities (33) and (34) are violated and provides an $\bar{F}(M)$ dual solution of cost $z(\bar{F}(M))$.

5.2.3. Phase 3: Solving $F(M)$ to Optimality. In Phase 3, two steps are performed.

(1) Define the reduced problem $F'(M)$ resulting from $F(M)$ by doing the following:

(i) Replace the route set $\mathcal{R}_k$, $k \in N_S(M)$, with the largest subset $\mathcal{R}'_k \subseteq \mathcal{R}_k$ of routes such that $c'_{kl} < z(\text{UB}) - (U(M) + z(\bar{F}(M)))$, $l \in \mathcal{R}'_k$, $k \in N_S(M)$, where c'_l is the reduced cost of route $l \in \mathcal{R}'_k$ with respect to the $\bar{F}(M)$ dual solution achieved at Phase 2 and $z(\text{UB})$ is the current best upper bound on the 2E-CVRP.

(ii) Add all constraints (33) and (34) saturated by the final $\bar{F}(M)$ solution.

(2) Solve problem $F'(M)$ with a general purpose integer programming solver.

5.3. Description of the Exact Method

The exact method we propose for solving the 2E-CVRP can be described as follows:

(1) *Generate the set $\mathcal{M}$ and compute a lower bound on the* 2E-CVRP.

(a) Generate the set $\mathcal{M}$ of first-level routes by pure enumeration.

(b) Execute bounding procedure DP^1 to produce lower and upper bounds LD1 and UB1.

(2) *Generate the set $\mathcal{P}$ of configurations as described in* §5.1.

(3) *Solve the* 2E-CVRP.

(a) Initialize $z(F) = \text{UB1}$, $\text{LB} = \text{UB1}$, $z(\text{UB}) = \text{UB1}$, $\bar{\mathcal{P}} = \varnothing$ and $r^{\max} = 0$.

(b) If $\mathcal{P} = \varnothing$, then stop. Let

$$M = \arg\min_{M'\in\mathcal{P}} \{\text{LBW}(M')\}.$$

Remove M from $\mathcal{P}$. If $\text{LBW}(M) \geqslant z(F)$ then stop ($z(F)$ is the optimal 2E-CVRP solution cost).

(c) *Solve problem $F(M)$*.

(i) Execute Phase 1 (see §5.2.1) to compute lower bound $\text{LD1}(M)$ and upper bound $\text{UB1}(M)$ on $F(M)$. Update $z(\text{UB}) = \min\{z(\text{UB}), \text{UB1}(M)\}$, $z(F) = \min\{z(F), \text{UB1}(M)\}$, and $\text{LB} = \min\{\text{LB}, z(F)\}$. If $\text{LD1}(M) \geqslant z(F)$, go to Step 3.b.

(ii) Execute Phase 2 (see §5.2.2) to compute lower bound $z(\bar{F}(M))$, and update

$$\text{LB} = \min\big\{\text{LB}, \max\{\text{LD1}(M), U(M) + z(\bar{F}(M))\}\big\}.$$

If $U(M) + z(\bar{F}(M)) \geqslant z(F)$, go to Step 3(b). If the $\bar{F}(M)$ solution of cost $z(\bar{F}(M))$ is integer, update $z(F) = U(M) + z(\bar{F}(M))$ and go to Step 3(b).

(iii) Execute Phase 3 (see §5.2.3) to solve problem $F(M)$. Let $z(F(M))$ be the optimal solution cost of $F'(M)$. Update $z(F) = \min\{z(F), U(M) + z(F(M))\}$, $\bar{\mathcal{P}} = \bar{\mathcal{P}} \cup \{M\}$ and $r^{\max} = \max\{r^{\max}, \sum_{k\in N_S(M)} |\mathcal{R}'_k|\}$. Go to Step 3(b).

Notice that Step 3(c(iii)) is executed for any configuration $M \in \mathcal{P}$ such that $U(M) + z(\bar{F}(M)) < z(F)$ and the $\bar{F}(M)$ solution is not integer. Thus, if Step 3(c(iii)) is never executed, the algorithm terminates with $\bar{\mathcal{P}} = \varnothing$, implying that $\text{LB} = z(F)$ and the optimal 2E-CVRP solution corresponds to either the initial upper bound $\text{UB1}(M)$ computed at Step 3(c(i)) or to the integer $\bar{F}(M)$ solution achieved at Step 3(c(ii)) for some configuration $M \in \mathcal{P}$.

At the end of the exact method, LB represents a valid lower bound on the 2E-CVRP because it corresponds to $\text{LB} = \min_{M\in\mathcal{P}}\{\max\{\text{LD1}(M),\ U(M) + z(\bar{F}(M))\}\}$. Value $r^{\max}$ is the maximum number of second-level routes generated, and set $\bar{\mathcal{P}}$ contains the configurations for which the corresponding problem $F(M)$ was solved to optimality at Step 3(c(iii)). Value $z(\text{UB})$ is the cost of the best upper bound computed at Step 1 or at Step 3(c(i)). Finally, because we impose a limit $\Delta^{\max}$ on the maximum number of second-level routes, $\bigcup_{k\in N_S(M)}\mathcal{R}'_k$, to generate at Step 3(c(iii)), whenever such limit is reached for some configuration $M\in\bar{\mathcal{P}}$, at the end of the algorithm, the value $z(F)$ is an upper bound on the 2E-CVRP but is not necessarily the optimal solution cost.

6. Computational Results

We report on the computational results of the exact method (hereafter BMRW) described in §5.3 and its comparison with the methods of Perboli, Tadei, and Vigo (2011) (PTV) and Jepsen, Spoorendonk, and Ropke (2013) (JSR). BMRW was coded in Fortran 77. CPLEX 12.1 (see CPLEX 2009) was used as the linear programming and integer programming solver. All tests were run on an IBM Intel Xeon X7350 Server (2.93 GHz—16 GB of RAM).

We considered four sets of instances from the literature: set 2 and 3 (introduced by Feliu et al. 2007), set 4 (Crainic et al. 2010), and set 5 (Hemmelmayr et al. 2012). In all these instances, the handling costs are zero. We generated another set of 54 instances introduced also to evaluate the effectiveness of BMRW on instances with nonzero handling costs. The instances are partitioned into two sets, called 6A and 6B, corresponding to instances with *zero* and *nonzero* handling costs, respectively. The details on the instances can be found in §EC.3 of the online supplement.

In all instances the satellite capacities are unlimited (i.e., $W_k=\infty$, $k\in N_S$). Furthermore, in sets 2, 3, and 5 the maximum number of vehicles per satellite is unlimited (i.e., $m_k=\infty$, $k\in N_S$). Set 4 was treated differently by Jepsen, Spoorendonk, and Ropke (2013) who considered the given upper bounds on the maximum number of vehicles per satellite, m_k, and Perboli, Tadei, and Vigo (2011) who ignored such values. To compare BMRW with both JSR and PTV, we considered two versions of set 4, namely, set 4A and set 4B, where set 4A corresponds to the original set 4 whereas, in set 4B, m_k is unbounded (i.e., $m_k=\infty$, $k\in N_S$). Perboli, Tadei, and Vigo (2011) solved to optimality 66 instances with 12 customers and two satellites (therein set 1). BMRW solved all these instances in a few seconds, so corresponding results are not reported.

According to Standard Performance Evaluation Corporation (SPEC) (http://www.spec.org/benchmarks.html), our machine is 10% faster than the Intel(R) Xeon X5550 2.67 GHz with 24 GB of memory and eight cores of JSR and twice as fast as the 3 GHz Pentium PC with 1 Gb of RAM of PTV. A time limit of 10,000 seconds was imposed on PTV and JSR.

BMRW used the following parameter settings. In DP^1, we set $\Delta(N_i)=12$, and we set $Maxit1=25$, $\epsilon=1.0$, $Maxit2=200$, at Step 1, and $Maxit1=10$, $\epsilon=0.5$, $Maxit2=100$ at Step 3(c(i)). Moreover, we set $\Delta^{\max}=10^6$ and imposed a time limit of 5,000 seconds to solve problem $F'(M)$.

Tables 1–7 report the results obtained by BMRW on the seven sets of instances. The tables report the instance name, the number n_s of satellites, the cost $z(F)$ of the best

Table 1. Computational results on set 2 instances.

Name	n_s	$z(F)$	%LD1	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	$r^{\max}$	t_{tot}
E-n22-k4-s6-17	2	417.07	99.9	0.4	1	100.0	100.0	0.5	0	0	0.5
E-n22-k4-s8-14	2	384.96	99.5	0.4	1	100.0	100.0	0.7	0	0	0.7
E-n22-k4-s9-19	2	470.60	95.4	0.5	1	100.0	100.0	1.2	0	0	1.2
E-n22-k4-s10-14	2	371.50	99.6	0.5	1	100.0	100.0	0.5	0	0	0.5
E-n22-k4-s11-12	2	427.22	96.5	0.4	2	100.5	100.0	1.3	0	0	1.3
E-n22-k4-s12-16	2	392.78	96.7	0.5	2	100.0	100.0	1.1	0	0	1.1
E-n33-k4-s1-9	2	730.16	97.9	25.1	1	100.0	100.0	37.6	0	0	37.6
E-n33-k4-s2-13	2	714.63	97.8	27.8	2	100.0	100.0	34.9	0	0	34.9
E-n33-k4-s3-17	2	707.48	95.0	28.9	3	105.8	100.0	48.1	0	0	48.1
E-n33-k4-s4-5	2	778.74	94.1	23.1	4	100.9	100.0	72.5	0	0	72.5
E-n33-k4-s7-25	2	756.85	96.8	27.4	3	101.0	100.0	47.1	0	0	47.1
E-n33-k4-s14-22	2	779.05	98.7	26.0	3	100.0	100.0	31.7	0	0	31.7
E-n51-k5-s3-18	2	597.49	93.5	3.0	5	100.0	99.8	23.7	1	33,547	25.8
E-n51-k5-s5-47	2	530.76	98.1	3.1	4	101.6	99.8	25.9	1	34,110	27.5
E-n51-k5-s7-13	2	554.81	94.6	3.3	6	100.2	98.9	37.3	2	42,075	55.1
E-n51-k5-s12-20	2	581.64	95.5	3.1	3	100.5	99.3	27.1	1	39,033	44.3
E-n51-k5-s28-48	2	538.22	95.8	3.2	6	100.0	99.7	40.1	2	34,797	44.0
E-n51-k5-s33-38	2	552.28	95.4	3.8	3	100.0	100.0	13.6	0	0	13.6
E-n51-k5-s3-5-18-47	4	530.76	96.6	6.6	55	100.0	99.9	259.2	1	62,913	260.8
E-n51-k5-s7-13-33-38	4	531.92	94.7	7.6	68	100.0	99.4	263.6	1	68,796	266.6
E-n51-k5-s12-20-28-48	4	527.63	95.6	9.0	24	100.0	99.6	71.8	1	67,620	74.2

Table 2. Computational results on set 3 instances.

Name	n_s	$z(F)$	%LD1	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	r^{max}	t_{tot}
E-n22-k4-s13-14	2	526.15	96.4	0.4	4	100.0	100.0	2.1	0	0	2.1
E-n22-k4-s14-19	2	498.80	93.2	0.5	6	100.0	100.0	2.4	0	0	2.4
E-n22-k4-s13-16	2	521.09	94.9	0.4	4	100.0	100.0	2.8	0	0	2.8
E-n22-k4-s17-19	2	512.80	95.5	0.5	4	100.0	100.0	2.6	0	0	2.6
E-n22-k4-s13-17	2	496.38	96.8	0.5	1	100.0	100.0	1.2	0	0	1.2
E-n22-k4-s19-21	2	520.42	94.9	0.5	5	100.0	100.0	3.8	0	0	3.8
E-n33-k4-s22-26	2	680.37	94.7	28.2	3	100.1	99.8	71.8	1	23,690	73.3
E-n33-k4-s16-22	2	672.17	92.0	32.6	5	102.0	99.4	115.4	1	26,474	127.5
E-n33-k4-s16-24	2	666.02	94.6	38.0	5	100.1	99.9	125.2	1	23,040	128.4
E-n33-k4-s24-28	2	670.43	95.6	31.0	3	100.0	100.0	73.1	1	24,266	78.8
E-n33-k4-s19-26	2	680.37	94.0	27.3	3	100.1	99.6	70.8	2	22,549	72.8
E-n33-k4-s25-28	2	650.58	95.7	30.7	3	100.3	100.0	56.0	0	0	56.0
E-n51-k5-s13-19	2	560.73	95.6	3.3	5	100.0	99.6	43.8	2	34,800	48.0
E-n51-k5-s13-42	2	564.45	97.8	3.6	1	100.3	99.1	18.3	1	44,083	50.1
E-n51-k5-s13-44	2	564.45	96.8	3.2	3	101.4	99.0	29.7	1	50,444	73.0
E-n51-k5-s40-42	2	746.31	91.2	3.6	5	102.6	99.0	34.8	1	53,016	107.2
E-n51-k5-s41-42	2	771.56	97.7	5.7	2	100.1	98.9	36.9	1	315,861	2,078.6
E-n51-k5-s41-44	2	802.91	91.8	4.1	4	101.2	99.6	39.8	1	39,979	59.4

Table 3. Computational results on set 4A instances.

Name	n_S	$z(F)$	%	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	r^{max}	t_{tot}
Instance50-1	2	1,569.42	97.1	23.3	2	100.1	99.9	72.5	2	30,903	75.5
Instance50-2	2	1,438.33	95.8	14.6	3	100.9	99.6	110.7	2	44,411	161.9
Instance50-3	2	1,570.43	97.1	23.3	2	102.6	99.9	67.8	1	31,377	70.6
Instance50-4	2	1,424.04	96.4	20.2	2	101.7	99.4	59.6	1	44,211	101.8
Instance50-5	2	2,193.52	98.3	25.1	5	100.2	99.6	286.2	5	67,491	663.7
Instance50-6	2	1,279.87	95.2	18.5	2	100.0	100.0	42.7	0	0	42.7
Instance50-7	2	1,458.63	98.0	29.7	2	104.7	99.8	92.7	2	30,995	100.4
Instance50-8	2	1,363.74	95.5	20.8	3	100.1	99.5	199.2	2	404,659	2,261.9
Instance50-9	2	1,450.27	98.0	28.5	2	104.5	99.9	82.6	1	29,683	84.6
Instance50-10	2	1,407.65	92.9	22.9	2	100.3	99.6	71.9	1	52,513	112.9
Instance50-11	2	2,047.46	99.0	35.7	5	100.8	99.5	225.5	5	88,929	339.1
Instance50-12	2	1,209.42	93.1	25.0	2	100.1	100.0	69.4	0	0	69.4
Instance50-13	2	1,481.83	95.5	24.5	2	102.4	99.9	86.1	2	30,277	92.1
Instance50-14	2	1,393.61	93.6	21.8	3	100.9	99.4	126.2	2	181,889	1,188.3
Instance50-15	2	1,489.94	95.5	25.1	2	102.4	99.8	66.6	1	30,130	71.5
Instance50-16	2	1,389.17	95.0	16.1	2	101.1	99.8	56.0	1	36,097	62.6
Instance50-17	2	2,088.49	97.3	28.8	5	100.7	99.8	253.0	2	40,721	305.3
Instance50-18	2	1,227.61	93.1	16.9	2	100.0	99.3	60.5	1	49,390	117.2
Instance50-19	3	1,564.66	92.5	72.7	8	100.0	99.3	179.3	2	53,950	234.3
Instance50-20	3	1,272.97	93.7	24.5	8	101.3	99.1	59.4	1	84,784	140.1
Instance50-21	3	1,577.82	96.0	62.1	4	100.1	99.2	139.8	2	58,059	218.9
Instance50-22	3	1,281.83	95.1	33.9	8	101.4	100.0	76.2	1	50,177	79.2
Instance50-23	3	1,807.35	89.3	52.1	11	100.0	98.7	310.9	3	190,099	1,510.9
Instance50-24	3	1,282.68	95.1	28.6	14	100.0	100.0	80.0	0	0	80.0
Instance50-25	3	1,522.42	91.3	63.0	8	102.0	99.2	221.7	2	67,095	335.9
Instance50-26	3	1,167.46	97.2	27.1	1	100.2	99.9	51.9	1	49,094	54.0
Instance50-27	3	1,481.57	93.9	72.2	4	102.0	99.3	196.0	2	67,175	355.9
Instance50-28	3	1,210.44	93.2	38.3	10	100.0	100.0	279.5	1	55,285	295.6
Instance50-29	3	1,722.04	89.9	67.7	12	102.5	98.8	461.4	3	Δ^{max}	9,092.9
Instance50-30	3	1,211.59	93.5	32.8	13	100.5	100.0	243.1	0	0	243.1
Instance50-31	3	1,490.34	91.8	65.1	8	102.2	98.1	325.9	4	Δ^{max}	11,561.3
Instance50-32	3	1,199.00	94.1	25.9	7	100.1	98.7	262.7	1	619,322	4,009.4
Instance50-33	3	1,508.30	93.4	64.5	6	101.2	98.0	234.6	2	Δ^{max}	12,922.3
Instance50-34	3	1,233.92	93.2	30.7	10	100.0	99.0	130.8	1	85,850	207.0
Instance50-35	3	1,718.41	87.6	63.9	12	100.1	98.3	619.9	5	Δ^{max}	20,377.6
Instance50-36	3	1,228.89	93.3	28.6	14	100.0	99.3	121.6	1	59,745	154.1
Instance50-37	5	1,528.73	94.5	162.7	116	100.7	99.5	778.3	3	82,225	807.8
Instance50-38	5	1,169.20	93.9	52.5	134	100.9	99.0	429.3	1	253,662	1,648.2

Table 3. (Cont'd)

Name	n_S	$z(F)$	%LD1	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	r^{max}	t_{tot}
Instance50-39	5	1,520.92	94.6	168.0	63	100.6	99.8	688.0	2	78,320	695.0
Instance50-40	5	1,199.42	90.3	55.4	66	101.6	99.6	986.8	2	99,764	996.4
Instance50-41	5	1,667.96	95.3	195.4	64	100.3	99.6	1,302.9	4	79,889	1,344.7
Instance50-42	5	1,194.54	95.2	50.5	61	101.6	99.4	177.3	1	94,931	223.2
Instance50-43	5	1,439.67	95.4	175.5	56	101.3	99.5	1,032.2	3	87,237	1,095.7
Instance50-44	5	1,045.13	95.6	84.4	100	100.2	99.8	424.1	1	90,132	435.8
Instance50-45	5	1,450.96	94.9	160.9	34	101.7	99.2	577.7	2	135,411	774.0
Instance50-46	5	1,088.77	91.8	68.0	62	100.2	99.3	1,150.5	5	131,810	1,345.4
Instance50-47	5	1,587.29	96.2	205.5	62	102.1	99.4	1,470.7	2	98,905	1,566.3
Instance50-48	5	1,082.20	96.7	56.7	6	101.1	100.0	91.0	0	0	91.0
Instance50-49	5	1,434.88	95.0	164.3	74	102.3	100.0	714.8	0	0	714.8
Instance50-50	5	1,083.12	93.2	55.1	134	100.5	99.1	869.1	1	239,534	1,337.0
Instance50-51	5	1,398.05	94.6	179.6	64	101.0	100.0	744.0	1	73,279	748.4
Instance50-52	5	1,125.67	90.3	52.0	65	101.5	99.0	1,231.9	7	167,299	1,533.7
Instance50-53	5	1,567.77	95.0	211.4	63	100.1	98.7	1,712.0	2	268,139	4,223.3
Instance50-54	5	1,127.61	94.1	48.3	58	100.2	98.9	343.4	1	414,080	1,041.6

Table 4. Computational results on set 4B instances.

Name	n_S	$z(F)$	%LD1	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	r^{max}	t_{tot}
Instance50-1	2	1,569.42	95.0	39.2	3	101.5	100.0	113.3	2	31,196	117.0
Instance50-2	2	1,438.33	95.8	14.7	5	101.0	99.7	129.8	2	42,949	188.5
Instance50-3	2	1,570.43	95.1	38.9	3	101.5	100.0	95.1	1	31,017	97.8
Instance50-4	2	1,424.04	96.3	21.9	3	101.2	99.3	69.3	1	48,595	115.6
Instance50-5	2	2,193.52	98.3	40.8	7	100.2	99.6	326.6	5	62,190	631.5
Instance50-6	2	1,279.87	95.1	17.4	3	102.9	99.9	49.1	1	34,244	52.6
Instance50-7	2	1,408.57	98.5	41.5	3	101.6	99.9	73.0	1	31,607	76.5
Instance50-8	2	1,360.32	95.7	17.3	5	100.2	99.6	222.8	3	604,910	3,293.6
Instance50-9	2	1,403.53	98.7	40.0	3	103.3	99.9	80.3	1	35,515	81.7
Instance50-10	2	1,360.56	96.1	20.8	3	100.0	100.0	46.3	0	0	46.3
Instance50-11	2	2,047.46	99.0	50.6	7	100.8	99.5	296.8	6	88,960	450.8
Instance50-12	2	1,209.42	93.0	22.1	3	100.0	99.9	106.9	1	48,537	111.9
Instance50-13	2	1,450.93	96.1	44.1	3	102.3	100.0	94.1	0	0	94.1
Instance50-14	2	1,393.61	93.6	20.6	5	101.1	99.4	147.4	2	175,756	1,069.7
Instance50-15	2	1,466.83	95.5	39.7	3	101.1	99.9	103.4	1	29,552	106.0
Instance50-16	2	1,387.83	95.1	16.4	3	100.1	99.8	76.7	2	43,428	99.0
Instance50-17	2	2,088.49	97.3	44.3	7	100.6	99.8	306.0	2	38,365	358.3
Instance50-18	2	1,227.61	93.1	17.0	3	100.0	99.2	69.7	1	51,842	127.8
Instance50-19	3	1,546.28	93.7	86.3	16	101.2	99.2	262.5	1	60,815	293.4
Instance50-20	3	1,272.97	93.8	24.3	9	101.3	99.0	59.7	1	107,430	170.9
Instance50-21	3	1,577.82	96.0	85.6	12	100.8	99.2	199.8	2	58,718	250.9
Instance50-22	3	1,281.83	95.2	31.2	9	103.0	100.0	68.9	0	0	68.9
Instance50-23	3	1,652.98	96.6	83.6	7	102.0	100.0	193.7	0	0	193.7
Instance50-24	3	1,282.68	95.2	29.1	16	100.0	100.0	79.7	0	0	79.7
Instance50-25	3	1,408.57	98.2	87.2	7	101.9	99.9	150.9	1	45,580	155.0
Instance50-26	3	1,167.46	97.2	27.1	2	100.2	99.9	52.6	1	48,772	55.9
Instance50-27	3	1,444.51	96.5	93.1	8	102.3	99.9	194.2	1	44,411	198.1
Instance50-28	3	1,210.44	92.9	34.9	11	100.9	100.0	249.6	0	0	249.6
Instance50-29	3	1,552.66	96.7	104.4	7	103.0	100.0	257.3	1	48,508	258.4
Instance50-30	3	1,211.59	93.2	38.6	16	100.5	99.9	239.9	1	61,768	245.9
Instance50-31	3	1,440.86	94.8	87.5	13	101.2	100.0	262.2	0	0	262.2
Instance50-32	3	1,199.00	94.1	31.1	8	100.1	98.7	116.9	1	579,861	3,812.3
Instance50-33	3	1,478.86	95.4	86.0	11	101.5	99.0	239.6	1	121,197	467.9
Instance50-34	3	1,233.92	93.1	27.6	11	101.1	99.1	166.6	1	131,443	287.7
Instance50-35	3	1,570.72	94.7	95.5	7	102.4	98.9	332.0	3	221,318	1,299.7
Instance50-36	3	1,228.89	93.1	26.8	16	100.0	99.2	126.2	1	72,167	180.1
Instance50-37	5	1,528.73	93.7	206.7	223	102.0	97.9	1,520.3	10	Δ^{max}	14,522.3
nstance50-38	5	1,163.07	94.6	78.8	153	100.0	99.1	485.7	3	229,650	1,163.0
Instance50-39	5	1,520.92	93.1	212.4	112	100.3	98.8	1,097.6	7	126,528	1,791.0
Instance50-40	5	1,163.04	93.0	59.2	82	101.3	99.6	312.8	1	96,251	348.0

Table 4. (Cont'd.)

Name	n_S	$z(F)$	%LD1	t_{LD1}	$\|\mathcal{P}\|$	%UB	%LB	t_{LB}	$\|\bar{\mathcal{P}}\|$	r^{max}	t_{tot}
Instance50-41	5	1,652.98	95.3	266.4	117	101.0	99.6	2,229.6	14	153,550	2,370.6
Instance50-42	5	1,190.17	95.5	55.5	75	101.9	99.2	238.4	2	132,823	432.4
Instance50-43	5	1,406.11	95.2	210.3	79	101.3	99.6	1,065.4	1	94,023	1,098.4
Instance50-44	5	1,035.03	96.4	67.0	93	100.9	100.0	382.9	1	84,919	387.2
Instance50-45	5	1,401.87	95.3	187.3	55	102.7	99.6	464.7	1	82,077	484.7
Instance50-46	5	1,058.11	94.7	83.5	65	100.4	100.0	426.9	0	0	426.9
Instance50-47	5	1,552.66	95.8	260.8	103	103.0	100.0	1,220.1	2	146,044	1,227.0
Instance50-48	5	1,074.50	97.3	60.0	6	100.5	99.9	121.4	1	74,056	125.5
Instance50-49	5	1,434.88	94.4	217.4	142	101.1	98.1	1,498.7	8	Δ^{max}	13,940.3
Instance50-50	5	1,065.25	94.8	89.2	126	100.0	99.9	500.7	1	83,400	508.0
Instance50-51	5	1,387.51	93.9	225.0	92	102.2	98.9	845.9	2	113,910	1,299.1
Instance50-52	5	1,103.42	92.0	56.0	81	100.1	99.4	788.8	1	109,254	846.0
Instance50-53	5	1,545.73	95.2	283.1	97	101.6	99.0	2,113.7	3	232,398	2,395.8
Instance50-54	5	1,113.62	95.2	51.4	36	100.8	99.0	232.0	2	222,523	1,027.9

Table 5. Computational results on set 5 instances.

Name	n_S	$z(F)$	%LD1	t_{LD1}	$\|\mathcal{P}\|$	%UB	%LB	t_{LB}	$\|\bar{\mathcal{P}}\|$	r^{max}	t_{tot}
2eVRP_100-5-1	5	1,564.46	97.9	16.5	44	102.1	99.3	381.8	5	147,996	9,359.6
2eVRP_100-5-1b	5	1,142.53[a]	93.8	36.5	60	100.0	94.6	1,339.2	10	Δ^{max}	24,028.9
2eVRP_100-5-2	5	1,016.32	95.6	15.2	197	100.6	99.1	928.8	19	156,424	10,517.6
2eVRP_100-5-2b	5	796.53[a]	95.2	29.9	100	100.0	96.6	1,750.8	10	Δ^{max}	26,099.7
2eVRP_100-5-3	5	1,045.29	97.5	16.3	50	100.3	99.2	262.5	13	137,379	2,930.2
2eVRP_100-5-3b	5	833.94[a]	95.7	39.7	95	100.0	97.6	1,475.1	8	Δ^{max}	32,693.8

[a]Hemmelmayr et al. (2012) computed improved upper bounds for instances 2eVRP_100-5-1b, 2eVRP_100-5-2b and 2eVRP_100-5-3b of values 1,111.34, 782.25 and 828.99, respectively.

Table 6. Computational results on set 6A (instances with zero handling costs).

Name	n_S	$z(F)$	%LD1	t_{LD1}	$\|\mathcal{P}\|$	%UB	%LB	t_{LB}	$\|\bar{\mathcal{P}}\|$	r^{max}	t_{tot}
A-n51-4	4	652.00	94.6	6.7	16	100.3	99.7	108.0	4	55,040	119.1
A-n51-5	5	663.41	95.7	10.4	81	100.3	99.7	149.8	2	64,146	154.8
A-n51-6	6	662.51	94.9	15.9	246	100.6	100.0	262.8	0	0	263.1
A-n76-4	4	985.95	95.9	16.1	71	101.9	99.4	267.6	2	78,465	343.7
A-n76-5	5	979.15	95.8	27.6	281	101.8	99.6	818.9	4	91,512	857.2
A-n76-6	6	970.20	95.8	46.0	1,391	101.8	99.5	3,215.2	8	108,558	3,327.4
A-n101-4	4	1,194.17	95.9	52.7	120	101.0	99.2	1,709.6	11	461,313	5,971.3
A-n101-5	5	1,211.38	96.6	43.7	647	102.5	99.4	3,540.3	4	234,285	4,823.3
A-n101-6	6	1,158.98	95.8	91.9	4,814	101.7	98.7	24,851.2	51	Δ^{max}	118,077.4
B-n51-4	4	563.98	96.0	6.6	10	101.1	98.8	30.3	1	87,968	56.8
B-n51-5	5	549.23	94.8	9.7	64	101.0	99.0	105.8	1	100,364	130.9
B-n51-6	6	556.32	94.3	15.5	106	100.0	100.0	125.5	0	0	125.6
B-n76-4	4	792.73	94.5	12.4	23	102.1	99.3	257.9	2	119,142	333.7
B-n76-5	5	783.93	94.0	21.3	167	101.5	99.3	572.2	1	144,513	610.8
B-n76-6	6	774.17	94.4	32.6	877	102.2	99.6	2,023.2	2	164,489	2,075.0
B-n101-4	4	939.21	97.3	42.1	10	102.0	98.9	195.7	1	485,177	2,512.5
B-n101-5	5	967.82	94.8	37.3	587	102.1	99.1	4,772.3	7	331,423	7,058.8
B-n101-6	6	960.29	96.2	76.9	456	103.0	99.1	1,970.4	4	273,755	3,772.4
C-n51-4	4	689.18	95.5	6.7	26	100.3	99.4	54.2	1	51,909	78.3
C-n51-5	5	723.12	93.6	10.3	64	100.3	98.9	81.4	1	125,120	431.6
C-n51-6	6	697.00	94.4	16.3	126	101.1	99.4	145.8	1	76,426	166.0
C-n76-4	4	1,054.89	94.9	15.5	66	102.9	99.1	290.7	4	108,471	454.3
C-n76-5	5	1,115.32	92.5	21.6	303	103.0	99.2	1,089.2	6	136,054	1,817.0
C-n76-6	6	1,064.72	92.2	31.7	1,449	101.7	97.8	4,786.7	20	Δ^{max}	47,840.9
C-n101-4	4	1,305.68	95.0	49.3	116	101.7	98.4	1,392.8	10	Δ^{max}	29,626.4
C-n101-5	5	1,309.42	96.4	84.4	206	101.6	98.6	1,203.2	2	Δ^{max}	10,865.1
C-n101-6	6	1,284.48	96.2	94.6	1,373	103.3	98.7	6,804.4	8	Δ^{max}	27,969.4

Table 7. Computational results on set 6B (instances with nonzero handling costs).

Name	n_S	$z(F)$	$z(HC)$	%LD1	t_{LD1}	$\lvert\mathcal{P}\rvert$	%UB	%LB	t_{LB}	$\lvert\bar{\mathcal{P}}\rvert$	r^{max}	t_{tot}
A-n51-4	4	744.24	81.24	98.2	6.8	2	100.0	99.4	44.4	2	54,769	88.7
A-n51-5	5	811.52	104.91	97.2	9.4	45	100.0	99.9	83.3	1	60,674	86.8
A-n51-6	6	930.11	235.76	96.7	15.8	174	100.0	99.5	345.2	6	79,716	383.5
A-n76-4	4	1,385.51	351.52	97.6	13.6	69	100.4	99.4	407.8	6	100,182	666.2
A-n76-5	5	1,519.86	485.90	98.8	21.5	162	101.8	99.8	469.5	2	89,793	497.0
A-n76-6	6	1,666.06	622.02	98.1	27.4	395	100.8	99.7	634.4	4	110,043	687.7
A-n101-4	4	1,881.44	597.69	97.9	48.9	117	101.3	99.5	1,147.5	8	318,548	3,359.3
A-n101-5	5	1,709.06	364.08	98.7	60.5	304	102.3	99.4	2,012.1	8	Δ^{max}	11,084.6
A-n101-6	6	1,777.69	491.05	97.3	79.8	5,767	102.9	99.6	32,975.7	66	Δ^{max}	38,316.8
B-n51-4	4	653.09	54.39	98.3	7.9	3	100.7	99.5	24.2	1	51,627	31.6
B-n51-5	5	672.10	89.37	97.1	18.7	11	100.7	99.9	41.2	1	61,785	43.3
B-n51-6	6	767.13	165.15	97.0	17.2	23	100.2	99.0	57.6	1	96,049	123.5
B-n76-4	4	1,094.52	275.22	97.9	17.3	9	101.0	99.3	60.0	1	92,311	127.0
B-n76-5	5	1,218.13	396.14	98.6	19.8	16	101.0	99.7	110.9	1	92,539	131.7
B-n76-6	6	1,326.76	489.99	98.0	30.7	92	101.3	99.7	212.8	1	107,631	228.3
B-n101-4	4	1,505.68	552.06	97.7	38.4	38	100.8	98.9	412.5	1	Δ^{max}	5,876.8
B-n101-5	5	1,400.62	385.16	97.1	52.3	797	101.1	98.7	6,473.5	19	Δ^{max}	32,302.1
B-n101-6	6	1,450.39	394.75	97.8	54.4	87	101.3	99.1	802.7	3	Δ^{max}	6,352.4
C-n51-4	4	866.58	84.00	96.4	6.7	23	100.3	99.3	70.8	3	69,690	120.7
C-n51-5	5	943.12	168.59	95.4	10.3	62	101.0	99.2	132.5	3	86,225	238.8
C-n51-6	6	1,050.42	246.72	97.2	16.5	105	100.0	99.7	232.2	4	76,281	291.2
C-n76-4	4	1,438.96	369.94	98.2	17.1	22	102.0	99.5	124.7	1	89,304	182.0
C-n76-5	5	1,745.39	581.31	96.4	19.3	108	100.9	99.5	354.4	2	173,901	840.6
C-n76-6	6	1,756.54	671.27	98.4	29.7	256	101.2	99.5	698.2	2	158,988	1,442.6
C-n101-4	4	2,070.27	735.09	97.5	30.0	131	101.1	99.1	766.3	2	Δ^{max}	12,080.4
C-n101-5	5	1,967.80	521.89	97.0	60.2	240	101.6	99.1	1,689.9	4	Δ^{max}	12,419.1
C-n101-6	6	1,869.29	546.57	97.6	65.7	352	101.6	98.8	1,933.8	5	Δ^{max}	23,285.2

solution found, the percentage ratio %LD1 of lower bound LD1 over $z(F)$ (i.e., $\%\text{LD1} = 100\ \text{LD1}/z(F)$), the time t_{LD1} in seconds for computing LD1, the cardinality $|\mathcal{P}|$ of the set $\mathcal{P}$ at the beginning of Step 3, the percentage ratio %UB of upper bound $z(\text{UB})$ over $z(F)$, the percentage ratio %LB of lower bound LB over $z(F)$, the total time t_{LB} in seconds for computing LD1 and LB, the cardinality $|\bar{\mathcal{P}}|$ of the set $\bar{\mathcal{P}}$, the value of r^{max}, and finally, the total computing time t_{tot} in seconds. Table 7 also shows column $z(HC)$ reporting the total handling cost. Whenever $|\bar{\mathcal{P}}| > 0$, the difference $t_{tot} - t_{LB}$ is the time spent by CPLEX for solving the problems $F'(M)$, for all configurations $M \in \bar{\mathcal{P}}$.

Tables 8–10 compare BMRW with PTV and JSR. Under the headings "PTV" and "JSR," we report the percentage ratio, over $z(F)$, of the upper bound (%UB) and of the lower bound (%LB) achieved at the root node, the percentage gap (*%gap*) between the best lower and upper bound computed, and the total computing time (t_{tot}) in seconds. The values in columns $z(F)$ are in bold whenever the instances were open before BMRW. The last lines of the tables report, for each method, the number of instances solved to optimality (in columns *%gap*), and, for JSR and BMRW, the average percentage deviation of the upper and lower bounds (in columns %UB and %LB) and the average computing

Table 8. Comparison with the exact methods PTV and JSR on set 2 instances.

		PTV		JSR				BMRW				
Name	$z(F)$	%UB	*%gap*	%UB	%LB	*%gap*	t_{tot}	%UB	%LB	t_{LB}	*%gap*	t_{tot}
E-n22-k4-s6-17	417.07	100.0	0.0	100.0	96.7	0.0	0.2	100.0	100.0	0.5	0.0	0.5
E-n22-k4-s8-14	384.96	106.0	0.0	100.0	98.0	0.0	1.0	100.0	100.0	0.7	0.0	0.7
E-n22-k4-s9-19	470.60	100.0	0.0	113.1	90.4	0.0	12.4	100.0	100.0	1.2	0.0	1.2
E-n22-k4-s10-14	371.50	117.3	0.0	100.0	96.8	0.0	1.2	100.0	100.0	0.5	0.0	0.5
E-n22-k4-s11-12	427.22	100.0	0.0	104.1	94.7	0.0	3.2	100.5	100.0	1.3	0.0	1.3
E-n22-k4-s12-16	392.78	108.4	0.0	100.0	95.9	0.0	2.0	100.0	100.0	1.0	0.0	1.1
E-n33-k4-s1-9	730.16	100.9	0.0	100.0	87.3	0.0	49.4	100.0	100.0	37.6	0.0	37.6
E-n33-k4-s2-13	714.63	103.0	1.5	100.0	89.4	0.0	34.2	100.0	100.0	34.9	0.0	34.9
E-n33-k4-s3-17	707.48	104.5	1.7	113.2	91.0	0.0	1,126.8	105.8	100.0	48.1	0.0	48.1
E-n33-k4-s4-5	778.74	104.9	1.5	100.0	87.6	0.0	54.9	100.9	100.0	72.5	0.0	72.5
E-n33-k4-s7-25	756.85	100.0	1.6	100.0	86.0	0.0	87.5	101.0	100.0	47.1	0.0	47.1
E-n33-k4-s14-22	779.05	100.0	1.6	105.9	88.0	0.0	2.4	100.0	100.0	31.7	0.0	31.7

Table 8. (Cont'd)

		PTV		JSR				BMRW				
Name	$z(F)$	%UB	%*gap*	%UB	%LB	%*gap*	t_{tot}	%UB	%LB	t_{LB}	%*gap*	t_{tot}
E-n51-k5-s3-18	**597.49**	100.0	2.6	100.0	92.6	4.5	—	100.0	99.8	23.7	0.0	25.8
E-n51-k5-s5-47	530.76	102.3	1.8	102.4	97.0	0.0	13.3	101.6	99.8	25.9	0.0	27.5
E-n51-k5-s7-13	**554.81**	100.0	4.1	100.0	94.4	1.6	—	100.2	98.9	37.3	0.0	55.1
E-n51-k5-s12-20	581.64	100.4	3.7	104.2	94.2	0.0	213.6	100.5	99.3	27.1	0.0	44.3
E-n51-k5-s28-48	**538.22**	100.0	2.0	100.0	95.5	0.8	—	100.0	99.7	40.1	0.0	44.0
E-n51-k5-s33-38	552.28	104.7	0.7	100.0	95.8	0.0	2,114.0	100.0	100.0	13.6	0.0	13.6
E-n51-k5-s3-5-18-47	530.76	102.2	2.8	103.3	94.4	0.0	84.0	100.0	99.9	259.2	0.0	260.8
E-n51-k5-s7-13-33-38	531.92	107.5	3.6	102.7	94.6	0.0	3,642.8	100.0	99.4	263.6	0.0	266.6
E-n51-k5-s12-20-28-48	527.63	113.8	1.5	109.4	95.5	0.0	798.7	100.0	99.6	71.8	0.0	74.2
Avg./solved		103.6	7	102.8	93.1	18	457.9	100.5	99.8		21	53.6

Table 9. Comparison with the exact methods PTV and JSR on set 3 instances.

		PTV		JSR				BMRW				
Name	$z(F)$	%UB	%*gap*	%UB	%LB	%*gap*	t_{tot}	%UB	%LB	t_{LB}	%*gap*	t_{tot}
E-n22-k4-s13-14	526.15	100.1	0.0	102.2	98.2	0.0	3.2	100.0	100.0	2.1	0.0	2.1
E-n22-k4-s14-19	498.80	105.0	0.0	105.0	91.1	0.0	61.2	100.0	100.0	2.4	0.0	2.4
E-n22-k4-s13-16	521.09	100.0	0.0	101.0	98.2	0.0	2.3	100.0	100.0	2.8	0.0	2.8
E-n22-k4-s17-19	512.80	100.0	0.0	104.8	93.8	0.0	8.0	100.0	100.0	2.6	0.0	2.6
E-n22-k4-s13-17	496.38	100.0	0.0	100.0	93.4	0.0	1.1	100.0	100.0	1.2	0.0	1.2
E-n22-k4-s19-21	520.42	101.4	0.0	101.4	95.4	0.0	5.5	100.0	100.0	3.8	0.0	3.8
E-n33-k4-s22-26	680.37	100.0	4.2	101.5	91.0	0.0	6.3	100.1	99.8	71.8	0.0	73.3
E-n33-k4-s16-22	**672.17**	100.0	5.7	113.2	93.0	2.1	—	102.0	99.4	115.4	0.0	127.5
E-n33-k4-s16-24	666.02	100.4	6.0	100.0	96.4	0.0	747.4	100.1	99.9	125.2	0.0	128.4
E-n33-k4-s24-28	670.43	103.3	5.6	100.0	94.8	0.0	17.6	100.0	100.0	73.1	0.0	78.8
E-n33-k4-s19-26	680.37	100.0	4.7	109.2	89.4	0.0	26.4	100.1	99.6	70.8	0.0	72.8
E-n33-k4-s25-28	650.58	100.0	5.3	100.0	92.6	0.0	158.2	100.3	100.0	56.0	0.0	56.0
E-n51-k5-s13-19	560.73			100.0	95.5	0.0	1,007.9	100.0	99.6	43.8	0.0	48.0
E-n51-k5-s13-42	564.45			106.1	96.6	0.0	208.3	100.3	99.1	18.3	0.0	50.1
E-n51-k5-s13-44	564.45			107.6	96.8	0.0	288.5	101.4	99.0	29.7	0.0	73.0
E-n51-k5-s40-42	**746.31**			100.9	88.9	7.5	—	102.6	99.0	34.8	0.0	107.2
E-n51-k5-s41-42	**771.56**			100.5	95.1	0.6	—	100.1	98.9	36.9	0.0	2,078.6
E-n51-k5-s41-44	**802.91**			100.0	89.7	7.0	—	101.2	99.6	39.8	0.0	59.4
Avg./solved		100.8	6	103.0	93.9	14	181.6	100.5	99.7		18	42.5

time (in columns t_{tot}), computed over all instances solved by JSR, that are a subset of the instances solved by BMRW. The results on sets 6A and 6B show that BMRW behave similarly with zero and nonzero handling costs.

On the sets 2 and 3 (see Tables 8 and 9), BMRWsolved to optimality all 39 instances, whereas PTV and JSR solved to optimality 13 and 32 instances, respectively. Of the 54 instances of the set 4A, BMRWand JSR solved 50 and 15 of them, respectively. None of the 18 instances considered by PTV of set 4B were solved to optimality, whereas 52 instances out of 54 were solved to optimality by BMRW. Tables 8–10 show that BMRW outperforms both PTV and JSR.

Tables 1–7 show that BMRW solved 185 out of 207 instances to optimality. Columns $|\mathcal{P}|$ and $|\bar{\mathcal{P}}|$ show the effectiveness of both the procedure applied at Step 2

Table 10. Comparison with the exact method JSR on set 4A instances.

			JSR				BMRW				
Name	n_S	$z(F)$	%UB	%LB	%*gap*	t_{tot}	%UB	%LB	t_{LB}	%*gap*	t_{tot}
Instance50-1	2	**1,569.42**	112.9	91.2	1.7	—	100.1	99.9	72.5	0.0	75.5
Instance50-2	2	1,438.33	100.0	91.5	0.0	1,146.7	100.9	99.6	110.7	0.0	161.9
Instance50-3	2	**1,570.43**	112.7	88.7	1.6	—	102.6	99.9	67.8	0.0	70.6
Instance50-4	2	**1,424.04**	100.0	90.1	0.9	—	101.7	99.4	59.6	0.0	101.8
Instance50-5	2	**2,193.52**	100.3	85.6	0.4	—	100.2	99.6	286.2	0.0	663.7
Instance50-6	2	1,279.87	100.0	97.2	0.0	4,463.4	100.0	100.0	42.6	0.0	42.7
Instance50-7	2	**1,458.63**	114.2	89.9	1.5	—	104.7	99.8	92.7	0.0	100.4

Table 10. (Cont'd)

Name	n_S	$z(F)$	JSR				BMRW				
			%UB	%LB	%*gap*	t_{tot}	%UB	%LB	t_{LB}	%*gap*	t_{tot}
Instance50-8	2	1,363.74	100.0	91.8	0.0	1,164.5	100.1	99.5	199.2	0.0	2,261.9
Instance50-9	2	**1,450.27**	113.9	89.8	1.3	—	104.5	99.9	82.6	0.0	84.6
Instance50-10	2	1,407.65	101.0	98.4	0.0	3,933.1	100.3	99.6	71.9	0.0	112.9
Instance50-11	2	**2,047.46**	100.9	89.7	0.6	—	100.8	99.5	225.5	0.0	339.1
Instance50-12	2	1,209.42	100.0	95.7	0.0	22.3	100.1	100.0	69.4	0.0	69.4
Instance50-13	2	**1,481.83**	111.9	91.7	1.2	—	102.4	99.9	86.1	0.0	92.1
Instance50-14	2	**1,393.61**	101.6	91.2	0.1	—	100.9	99.4	126.2	0.0	1,188.3
Instance50-15	2	**1,489.94**	111.8	91.7	1.1	—	102.4	99.8	66.6	0.0	71.5
Instance50-16	2	1,389.17	100.0	91.0	0.0	1,045.1	101.1	99.8	56.0	0.0	62.6
Instance50-17	2	**2,088.49**	100.4	85.7	0.1	—	100.7	99.8	253.0	0.0	305.3
Instance50-18	2	1,227.61	100.0	96.5	0.0	8,130.1	100.0	99.3	60.5	0.0	117.2
Instance50-19	3	**1,564.66**	109.8	82.2	0.5	—	100.0	99.3	179.3	0.0	234.3
Instance50-20	3	**1,272.97**	116.6	93.2	0.6	—	101.3	99.1	59.4	0.0	140.1
Instance50-21	3	**1,577.82**	106.8	88.5	0.4	—	100.1	99.2	139.8	0.0	218.9
Instance50-22	3	1,281.83	105.1	86.2	0.0	8,636.7	101.4	100.0	76.2	0.0	79.2
Instance50-23	3	**1,807.35**	100.0	83.2	10.0	—	100.0	98.7	310.9	0.0	1,510.9
Instance50-24	3	1,282.68	101.8	93.9	0.0	6,559.9	100.0	100.0	79.9	0.0	80.0
Instance50-25	3	**1,522.42**	100.8	83.7	1.2	—	102.0	99.2	221.7	0.0	335.9
Instance50-26	3	1,167.46	113.4	95.1	0.0	66.4	100.2	99.9	51.9	0.0	54.0
Instance50-27	3	**1,481.57**	107.0	86.4	1.0	—	102.0	99.3	196.0	0.0	355.9
Instance50-28	3	1,210.44	104.1	86.7	0.0	2,046.0	100.0	100.0	279.5	0.0	295.6
Instance50-29	3	1,722.04	100.9	80.5	0.8	—	102.5	98.8	461.4	1.2	9,092.9
Instance50-30	3	1,211.59	101.8	92.1	0.0	17.4	100.5	100.0	243.1	0.0	243.1
Instance50-31	3	1,490.34	109.7	90.3	1.5	—	102.2	98.1	325.9	1.9	11,561.3
Instance50-32	3	**1,199.00**	104.8	86.9	0.5	—	100.1	98.7	262.7	0.0	4,009.4
Instance50-33	3	1,508.30	105.3	82.7	1.3	—	101.2	98.0	234.6	2.0	12,922.3
Instance50-34	3	**1,233.92**	102.5	85.6	0.1	—	100.0	99.0	130.8	0.0	207.0
Instance50-35	3	1,718.41	100.3	79.3	1.2	—	100.1	98.3	619.9	1.7	20,377.6
Instance50-36	3	1,228.89	100.1	85.8	0.0	2,038.2	100.0	99.3	121.6	0.0	154.1
Instance50-37	5	**1,528.73**	108.7	82.6	2.9	—	100.7	99.5	778.3	0.0	807.8
Instance50-38	5	**1,169.20**	108.2	82.6	0.2	—	100.9	99.0	429.3	0.0	1,648.2
Instance50-39	5	**1,520.92**	106.4	84.6	0.4	—	100.6	99.8	688.0	0.0	695.0
Instance50-40	5	**1,199.42**	101.0	81.5	2.6	—	101.6	99.6	986.8	0.0	996.4
Instance50-41	5	**1,667.96**	108.1	86.6	1.4	—	100.3	99.6	1,302.9	0.0	1,344.7
Instance50-42	5	**1,194.54**	112.8	83.3	1.2	—	101.6	99.4	177.3	0.0	223.2
Instance50-43	5	**1,439.67**	113.3	87.4	1.7	—	101.3	99.5	1,032.2	0.0	1,095.7
Instance50-44	5	1,045.13	109.5	80.4	0.0	144.0	100.2	99.8	424.1	0.0	435.8
Instance50-45	5	**1,450.96**	108.5	82.1	1.0	—	101.7	99.2	577.7	0.0	774.0
Instance50-46	5	**1,088.77**	101.7	77.4	1.0	—	100.2	99.3	1,150.5	0.0	1,345.4
Instance50-47	5	**1,587.29**	109.7	83.5	1.0	—	102.1	99.4	1,470.7	0.0	1,566.3
Instance50-48	5	1,082.20	115.8	86.1	0.0	133.4	101.1	100.0	91.0	0.0	91.0
Instance50-49	5	**1,434.88**	108.4	84.3	2.1	—	102.3	100.0	714.7	0.0	714.8
Instance50-50	5	**1,083.12**	105.4	77.9	1.7	—	100.5	99.1	869.1	0.0	1,337.0
Instance50-51	5	**1,398.05**	106.6	82.3	4.6	—	101.0	100.0	744.0	0.0	748.4
Instance50-52	5	**1,125.67**	100.2	81.0	1.1	—	101.5	99.0	1,231.9	0.0	1,533.7
Instance50-53	5	**1,567.77**	109.4	83.8	1.3	—	100.1	98.7	1,712.0	0.0	4,223.3
Instance50-54	5	**1,127.61**	110.6	88.2	0.9	—	100.2	98.9	343.4	0.0	1,041.6
Avg./solved			105.9	87.1	15	2,599.1	101.0	99.4		50	276.0

for generating the set $\mathcal{P}$ and the procedures applied at Steps 3(c(i)) and 3(c(ii)) for computing valid lower bounds on $z(F(M))$. Notice that few problems $F(M)$ required to be solved by CPLEX (see columns $|\bar{\mathcal{P}}|$). BMRW was able to solve 44 out of 60 instances of the sets 5, 6A, and 6B (see Tables 5–7). On the other 16 instances, BMRW could not generate all second-level routes required to solve some problems $F(M)$ to optimality because of the gap between the computed lower and upper bounds.

Tables 8 and 9 indicate that the heuristic algorithm described in §4.5 provided better solutions, on average, than the heuristic algorithms of PTV and JSR. Finally, the results obtained by Hemmelmayr et al. (2012) are better than the results obtained by our heuristic, being their average percentage ratios equal to 100.0, 100.0, 100.3, and 100.0 on instance sets 2, 3, 4B, and 5, respectively.

7. Conclusions

In this paper, we have proposed a new exact method for solving the two-echelon capacitated vehicle routing problem. We have described a bounding procedure that is used by the exact algorithm to decompose the 2E-CVRP into a limited set of multidepot capacitated vehicle routing problems with side constraints. The optimal 2E-CVRP solution is obtained by solving the set of MDCVRPs generated. The proposed method was tested on 207 instances, taken both from the literature and newly generated, with up to 100 customers and six satellites. The new exact algorithm solved to optimality 144 out of the 153 instances from literature and closed 97 of them for the first time. The comparison with the state-of-the-art exact methods shows that the new exact method outperforms the other exact methods in terms of size, number of problems solved to optimality, and computing time.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.1120.1153.

Acknowledgments

The authors thank two anonymous referees for helpful comments.

References

Akca Z, Berger RT, Ralphs TK (2009) A branch-and-price algorithm for combined location and routing problems under capacity restrictions. Chinneck JW, Kristjansson B, Saltzman MJ, eds. *Operations Research and Cyber-Infrastructure*, Vol. 47 (Springer, New York), 309–330.

Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. *Math. Programming Ser. A* 120(2):347–380.

Baldacci R, Mingozzi A, Roberti R (2011a) New route relaxation and pricing strategies for the vehicle routing problem. *Oper. Res.* 59(5): 1269–1283.

Baldacci R, Mingozzi A, Wolfler Calvo R (2011b) An exact method for the capacitated location-routing problem. *Oper. Res.* 59(5):1284–1296.

Belenguer J-M, Benavent E, Prins C, Prodhon C, Wolfler Calvo R (2011) A branch-and-cut method for the capacitated location-routing problem. *Comput. Oper. Res.* 38(6):931–941.

CPLEX (2009) IBM ILOG CPLEX 12.1 callable library. ILOG.

Crainic TG, Mancini S, Perboli G, Tadei R (2008) Clustering-based heuristics for the two-echelon vehicle routing problem. Technical Report CIRRELT-2008-46, CIRRELT, Montreal, Quebec, Canada.

Crainic TG, Mancini S, Perboli G, Tadei R (2010) Two-echelon vehicle routing problem: A satellite location analysis. *Procedia Soc. Behav. Sci.* 2(3):5944–5955.

Crainic TG, Mancini S, Perboli G, Tadei R (2011) Multi-start heuristics for the two-echelon vehicle routing problem. Merz P, Hao JK, eds. *Proc. 11th Eur. Conf. Evolutionary Comput. Combin. Optim. EvoCOP'11* (Springer-Verlag, Berlin), 179–190.

Dror M, Trudeau P (1990) Split delivery routing. *Naval Res. Logist.* 37:383–402.

Feliu JG, Perboli G, Tadei R, Vigo D (2007) The two-echelon capacitated vehicle routing problem. Technical report DEIS OR.INGCE 2007/2(R), Department of Electronics, Computer Science, and Systems, University of Bologna, Bologna, Italy.

Hemmelmayr VC, Cordeau J-F, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Comput. Oper. Res.* 39(12):3215–3228.

Jepsen M, Spoorendonk S, Ropke S (2013) A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Sci.* 47(1):23–37.

Laporte G, Norbert Y, Arpin D (1986) An exact algorithm for solving a capacitated location-routing problem. *Ann. Oper. Res.* 6(9):291–310.

Nguyen V, Prins C, Prodhon C (2010) A multi-start evolutionary local search for the two-echelon location routing problem. *Hybrid Metaheuristics.* Lecture Notes in Computer Science, Vol. 6373 (Springer, Berlin), 88–102.

Perboli G, Tadei R, Masoero F (2010) New families of valid inequalities for the two-echelon vehicle routing problem. *Electronic Notes Discrete Math.* 36:639–646.

Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Sci.* 45(3):364–380.

Tan KC, Chew YH, Lee LH (2006) A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *Eur. J. Oper. Res.* 172(3):855–885.

Roberto Baldacci is an associate professor of operations research in the Department of Electronics, Computer Science, and Systems (DEIS) at the University of Bologna, Italy. His major research interests are in the area of transportation planning, logistics and distribution, and the solution of vehicle routing and scheduling problems over street networks. His research activities are in the theory and applications of mathematical programming. He has worked in the design of new heuristic and exact methods for solving combinatorial problems, such as routing and location problems.

Aristide Mingozzi is a professor of operations research in the Department of Mathematics at the University of Bologna, Italy. His main interests include mathematical programming, combinatorial optimization, graph theory, dynamic programming, and the development of exact and heuristic algorithms for the solution of real-life problems in distribution and scheduling.

Roberto Roberti is a postdoc at the University of Bologna. His research activity concerns the study and development of heuristic and exact algorithms for solving combinatorial optimization problems, such as traveling salesman problems, routing problems, and timetabling problems.

Roberto Wolfler Calvo is a professor of operations research in the Laboratoire d'Informatique de Paris Nord at the Université de Paris 13. His main research interests include combinatorial optimization, mathematical programming, reformulation and decomposition methods, reoptimization, and environmental decision support systems. He is interested in academic research as well as industrial applications.