The vehicle routing problem with transhipment facilities

Published Version:
The vehicle routing problem with transhipment facilities / Baldacci, Roberto; Ngueveu, Sandra Ulrich; Calvo, Roberto Wolfler. - In: TRANSPORTATION SCIENCE. - ISSN 0041-1655. - ELETTRONICO. - 51:2(2017), pp. 592-606. [10.1287/trsc.2016.0711]

This version is available at: https://hdl.handle.net/11585/597694 since: 2020-07-03

Published:

DOI: http://doi.org/10.1287/trsc.2016.0711

The Vehicle Routing Problem with Transhipment Facilities

Roberto Baldacci, Sandra Ulrich Ngueveu, and Roberto Wolfler Calvo

Transportation Science 2017 51:2, 592-606

The final published version is available online at: https://doi.org/10.1287/trsc.2016.0711

Submitted to *Transportation Science*
manuscript (Please, provide the mansucript number!)

The Vehicle Routing Problem with Transhipment Facilities

Roberto Baldacci
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", Via Venezia 52, 47521 Cesena, Italy, r.baldacci@unibo.it

Sandra Ulrich Ngueveu
CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France, sandra.ulrich.ngueveu@laas.fr
Univ de Toulouse, INP, LAAS, F-31400 Toulouse, France

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

This paper proposes an exact method for solving an optimization problem arising in several distribution networks, where customers can be served either directly, using vehicle routes from a central depot, or through transhipment facilities. The problem consists of optimizing the following inter-dependent decisions: selecting transhipment facilities, allocating customers to these facilities and designing vehicle routes emanating from a central depot to minimize the total distribution cost. This problem is called the Vehicle Routing Problem with Transhipment Facilities (VRPTF). The paper describes two integer programming formulations for the VRPTF, an edge-flow based formulation and a Set Partitioning (SP) based formulation. The LP-relaxation of the two formulations are further strengthened with the addition of different valid inequalities. Moreover, two new route relaxations that are used by dual ascent heuristics to find near-optimal dual solutions of the LP-relaxation of the SP model are described. The valid inequalities and the route relaxations are used in a branch-and-cut-and-price approach to solve the problem to optimality. The proposed method is tested on a large family of instances, including real-world instances, and the computational results obtained indicate the effectiveness of the proposed method.

Key words: transhipment facilities, dual ascent heuristic, column-and-cut generation
History: June 14, 2016

1. Introduction

In several distribution networks the shipment to a customer is performed either directly, using vehicle routes emanating from a central depot, or through intermediate depots or *transhipment facilities*. In the latter case, the shipment is first delivered to a transhipment facility by a vehicle route, and then it is successively delivered to the final customer. Transhipment facilities provide a way to consolidate shipments into large vehicle loads, thereby allowing for a reduction of total distribution cost, and provide the capability to transfer shipments between different vehicles or modes of transportation (e.g., railroads, aircraft). In some cases, the transhipment facilities can be part of the same company which owns the central depot, and which makes the final delivery to the customers with its fleet of vehicles. In other cases, transhipment facilities are owned by a third-party subcontractor, who is also in charge of performing the final shipment to the customers.

The problem addressed in this paper is motivated by a real application of interest to an Italian company operating in the production and distribution of non-perishable products. More specifically, the problem consists of selecting transhipment facilities, allocating customers to these facilities and designing vehicle routes to minimize the total distribution cost. We call this problem the Vehicle Routing Problem with Transhipment Facilities (VRPTF). In the VRPTF, each customer can be served either directly by a vehicle route or through a facility selected from a set of potential facilities to which the customer can be assigned. The total load of a vehicle route, computed as the sum of the customer demands and of the quantities delivered to the facilities, must be less than or equal to the vehicle capacity. The problem objective is to minimize the total sum of routing and assignment costs.

1.1. Literature review

The VRPTF generalizes the well-known Capacitated Vehicle Routing Problem (CVRP). In the CVRP, a fleet of identical vehicles located at a central depot has to be optimally routed to supply a set of customers with known demands. Each vehicle performs at most one route, each customer must be visited exactly once, and the total demand of the customers visited by a route cannot exceed the vehicle capacity. The book edited by Toth and Vigo (2014) provides a comprehensive overview of exact methods for the CVRP and other variants.

As far as the authors know, the VRPTF has never been addressed in the literature. Closely related problems to the VRPTF are the Capacitated m-Ring-Star Problem (CmRSP), the Multiple Vehicle Traveling Purchaser Problem (MVTPP), the Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP), and the Location Routing Problem (LRP). The CmRSP, introduced in Baldacci et al. (2007), arises in the design of urban optical telecommunication networks and it consists of designing a set of rings that pass through a telephone exchange and through some transition points (also called *steiner nodes*) and/or users. Each nonvisited user must be assigned to a visited point or to a user. The number of users visited and assigned to a ring is bounded by the capacity of the ring. The objective is to minimize the total routing cost plus the assignment costs. The special case of the CmRSP arising when the users can be assigned only to steiner nodes, can be solved as a VRPTF with unit demands. The MVTPP described by Riera-Ledesma and Salazar-González (2012) models a family of routing problems combining stop selection and bus route generation. The problem consists of choosing a set of bus stops to which users are assigned, and simultaneously designing bus routes visiting such stops. The total number of users assigned to the stops of a route cannot exceed the seat capacity of a bus. The objective is to minimize the total length of all routes plus the total assignment cost. The undirected version of the MVTPP is equivalent to the VRPTF with the additional constraint imposing that the customers can only be assigned to facilities (or bus stops) and cannot be visited by a route. Both Baldacci et al. (2007) and Riera-Ledesma and Salazar-González (2012) proposed branch-and-cut approaches for the solution of the CmRSP and MVTPP, respectively. Recently, Riera-Ledesma and González (2013) also proposed a branch-and-cut-and-price algorithm for the MVTPP. The 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 visiting satellites only whereas the second level routes supply all customers. The main difference between the VRPTF and the 2E-CVRP is that in the VRPTF a customer can be either visited on a route or assigned to a facility, whereas in the 2E-CVRP each customer is visited once by exactly a second level route. The 2E-CVRP model is particularly useful when the facilities are part of the same company owing the main depot whereas in the VRPTF model the facilities are generally owned by third-party contractors, which are in charge of delivering to the final customers the quantity consolidated at the facilities. Exact methods for the 2E-CVRP have been proposed by Jepsen et al. (2013) and Baldacci et al. (2013). The LRP is a special case of the 2E-CVRP and consists of opening a set of depots and designing a set of routes for each opened depot so that the total load of the routes operated from a depot does not exceed its capacity and each customer is visited by exactly one route. The objective is to minimize the sum of the fixed costs of the opened depots and the costs of the routes operated from the depots. A recent review of location routing problem variants and heuristic and exact algorithms can be found in Prodhon and Prins (2014).

Another related problem to the VRPTF is the Multi-Vehicle Covering Tour Problem (m-CTP) introduced by Hachicha et al. (2000). In the m-CTP two sets of locations are given. The first set, consists of potential locations at which some vehicles may stop, and the second set are locations not actually on vehicle routes, but within an acceptable distance from a vehicle route. The m-CTP consists of determining a set of total minimum length vehicle routes on a subset of the first set of locations, subject to side constraints, such that every location of the second set is within a prespecified distance from a route. Há et al. (2013) proposed a branch-and-cut for the variant named the m-CTP-p where an upper bound on the number of vertices per route is given with a parameter p and the m number of vehicles used is a decision variable.

The VRPTF does not require any specific synchronization of incoming and outgoing vehicles at the facilities. In some practical applications, a correct synchronization can be required and in this case the facilities are generally referred as cross-docking facilities. For an overview of the cross-docking concept and extensive review of the existing literature the reader is referred to Belle et al. (2012). In this context, a generic class of VRPs that has recently received attention in the literature is the class of VRPs with Multiple Synchronization Constraints (VRPMSs). VRPMSs exhibit synchronization requirements between the vehicles, concerning spatial, temporal, and load aspects. A review of VRPMS presenting a classification of different types of synchronizations and a discussion about heuristic and exact algorithms can be found in Drexl (2012).

1.2. Contributions of this paper

This paper addresses a new problem of practical relevance and proposes both heuristic and exact methods for its solution. More specifically, we introduce a *two-index* formulation (TI) and we describe different valid inequalities for it, both by adapting those already proposed for the CmRSP, and by introducing new ones specific for the VRPTF. We also describe lower bounds derived from a set-partitioning based formulation (SP) of the problem, and computed using two efficient dual ascent heuristics that use two new route relaxations, called q-$*$route and ng-$*$route, respectively. The proposed methods have been tested on a large family of instances, including both instances derived from the literature and real-world instances. The computational results show that real-world instances with up to 142 customers and 18 facilities were solved to optimality and that high quality solutions were computed for instances with up to 164 customers. In addition, tight lower bounds were computed, with average percentage deviations equal to 98.7% and 97.4% for real-word and literature-based instances, respectively.

This paper is organized as follows. The next section formally introduces the VRPTF and presents formulation TI for which different valid inequalities are described in Section 3. Section 4 presents formulation SP and lower bounds based on its LP-relaxation; some properties of the LP-relaxation of SP are also investigated in the section. A bounding method used to compute a lower bound on the VRPTF is described in Section 5. Section 6 describes the exact method used to solve the VRPTF to optimality together with two heuristic algorithms. Section 7 reports computational results, and concluding remarks are given in Section 8.

2. Problem description and Two-Index (TI) formulation

This section describes the VRPTF and presents a *edge-flow* based formulation to model it.

The VRPTF is defined on a mixed graph $G = (V, E \cup A)$, where $V = \{0\} \cup V'$ is the node set, $E = \{\{i, j\} : i, j \in V, i \neq j\}$ is the edge set, and A is the arc set. Node set V' is partitioned into two subsets: $V_C = \{1, \dots, n_C\}$ containing a node for each customer and $V_F = \{n_C + 1, \dots, n_C + n_F\}$

containing a node for each transhipment facility. Node 0 represents a central depot. Each customer $i \in V_C$ requires a supply of q_i units from the depot (we assume $q_i = 0$, $\forall i \in \{0\} \cup V_F$) that can be delivered either directly from a vehicle route emanating from the depot or through a facility selected from a set $F_i \subseteq V_F$ of facilities to which customer i can be assigned. Set A represents the possible assignments between customers and facilities, i.e., $A = \{(i,j) : i \in V_C, j \in F_i\}$. Set E is the set of possible route edges, each edge $e = \{i,j\} \in E$ is associated with a non-negative *routing* cost $r_e = r_{\{i,j\}}$, while each arc $(i,j) \in A$ is associated with a non-negative *assignment* cost d_{ij}. Henceforth, if e connects the two nodes i and j then $\{i,j\}$ and e will be used interchangeably to denote the same edge.

A *route* is defined by a pair (R, A') where $R = (0, i_1, \ldots, i_r, 0)$, $r \geq 1$, is a simple cycle in G passing through the depot, visiting nodes $V(R) = \{i_1, \ldots, i_r\} \subseteq V'$, and $A' \subseteq A$ are assignments between customers of $V_C \setminus V(R)$ and nodes of $V(R) \cap V_F$. Notice that if $r = 1$ then route R represents the single-node route $R = (0, i_1, 0)$. We say that a customer i is *assigned* to a route R if it is either visited by the simple cycle (i.e., $i \in V(R)$) or it is connected to a node of the route representing a facility (i.e., a node $j \in V(R) \cap V_F$ exists such that $(i,j) \in A'$). The total load of a route is computed as the sum of the demands of the customers assigned to the route. The route is *feasible* if its total load does not exceed the vehicle capacity Q. The *cost* of a route is equal to the sum of the routing costs of the edges forming the route plus the sum of the assignment costs of the arcs in A'.

The aim of VRPTF is to design a set of routes so that each customer is assigned to exactly one route, each intermediate facility is visited at most once and the sum of the route costs is minimized.

We will use the following notation throughout. For any $S \subseteq V'$, let $V_C(S) = S \cap V_C$ and $V_F(S) = S \cap V_F$ denote the set of customers and of facilities in S, respectively. Let $F_i(S) = V_F(S) \cap F_i$ denote the set of facilities in S associated with customer $i \in V_C$. Also for any set $S \subseteq V$, define $\delta(S) = \{\{i,j\} \in E : i \in S, j \notin S\}$ (if $S = \{i\}$, we simply write $\delta(i)$ instead of $\delta(\{i\})$).

Let x_e be an integer variable which takes value in $\{0,1\}$, $\forall e \in E \setminus \{\{0,j\} : j \in V'\}$ and value in $\{0,1,2\}$, $\forall e \in \{\{0,j\} : j \in V'\}$. Notice that $x_{\{0,j\}} = 2$ when the single-node cycle $R = (0,j,0)$ is selected in the solution. For each arc $(i,j) \in A$, let z_{ij} be a binary variable which is equal to 1 if and only if customer i is assigned to node j. Moreover, for each $i \in V'$, let y_i be a binary variable which is equal to 1 if and only if node i is on a route. Formulation TI is as follows:

$$(TI) \quad \min \sum_{e \in E} r_e x_e + \sum_{(i,j) \in A} d_{ij} z_{ij} \tag{1}$$

$$s.t. \sum_{e \in \delta(i)} x_e = 2y_i, \qquad \forall i \in V' \tag{2}$$

$$y_i + \sum_{j \in F_i} z_{ij} = 1, \qquad \forall i \in V_C \tag{3}$$

$$\sum_{e\in\delta(S)} x_e \geq \frac{2}{Q}\left(\sum_{i\in V_C(S)} q_i y_i + \sum_{(i,j)\in A: j\in V_F(S)} q_i z_{ij}\right), \qquad \forall S\subseteq V' : S\neq\emptyset \qquad (4)$$

$$x_e \in \{0,1\}, \qquad \forall e\in E\setminus\{\{0,j\}: j\in V'\} \qquad (5)$$

$$x_e \in \{0,1,2\}, \qquad \forall e\in \{\{0,j\}: j\in V'\} \qquad (6)$$

$$z_{ij} \in \{0,1\}, \qquad \forall (i,j)\in A \qquad (7)$$

$$y_i \in \{0,1\}, \qquad \forall i\in V'. \qquad (8)$$

Constraints (2) impose that the degree of each node $i\in V'$ is 2 if the node is on a route. Constraints (3) state that a customer $i\in V_C$ is either on a route or is assigned to one of its facilities. Inequalities (4) are the *fractional route capacity inequalities* (FrCC). These constraints, within the integrality of x, z and w variables, impose that for a given subset S of nodes, at least $\left\lceil(\sum_{i\in S} q_i y_i + \sum_{(i,j)\in A: j\in S} q_i z_{ij})/Q\right\rceil$ routes are needed to visit the customers assigned to nodes in S.

3. Strengthening the LP-relaxation of formulation TI

A number of valid inequalities can be used to improve the quality of the lower bound obtained from the LP-relaxation of formulation TI. In this section, we first derive valid inequalities by extending the results proposed for the CmRSP by Baldacci et al. (2007) to the VRPTF. Then, a new class of valid inequalities specifically devised for the VRPTF is introduced. The separation procedures for different valid inequalities are then described in Section 5.2.

Simple valid inequalities are the following: (i) $x_{\{i,j\}} \leq y_j, i\in V_C, j\in V_C, i\neq j$; (ii) $x_{\{i,j\}} \leq y_j, i\in V_F, j\in V', i\neq j$; (iii) $x_{\{i,j\}} + z_{ij} \leq y_j, i\in V_C, j\in F_i$, (iv) $y_j \leq \sum_{i\in V_C: j\in F_i} z_{ij}, \forall j\in V_F$. Further, the following inequalities are also valid.

a) *Connectivity inequalities* (CI):

$$\sum_{e\in\delta(S)} x_e \geq 2\left(y_i + \sum_{j\in V_F(S)\cap F_i} z_{ij}\right), \quad \forall S\subseteq V', \forall i\in V_C(S), S\neq\emptyset. \qquad (9)$$

b) *Multistar inequalities* (MI):

$$\sum_{e\in\delta(S)} x_e \geq \frac{2}{Q}\left(\sum_{i\in V_C(S)} q_i y_i + \sum_{(i,j)\in A: j\in V_F(S)} q_i z_{ij} + \sum_{i\in V_C(\overline{S})}\sum_{j\in S} q_i x_{\{i,j\}}\right), \forall S\subseteq V', S\neq\emptyset. \qquad (10)$$

where $\overline{S} = V'\setminus S$.

c) *Rounded capacity constraints I* (RCI):

$$\sum_{e\in\delta(S)} x_e \geq 2\left\lceil \sum_{i\in S: F_i\subseteq S} q_i/Q \right\rceil, \quad \forall S\subseteq V', V_C(S)\neq\emptyset. \qquad (11)$$

d) *Rounded capacity constraints II* (RCII):

$$\sum_{e\in\delta(S)} x_e \geq 2\left\lceil \left(\sum_{i\in V_C(S)} q_i y_i + \sum_{\substack{(i,j)\in A:\\ j\in V_F(S)}} q_i z_{ij} \right) /Q \right\rceil, \quad \forall S\subseteq V', S\neq\emptyset. \tag{12}$$

Notice that CI inequalities are not dominated by MI inequalities whereas MI inequalities dominate FrCC inequalities. RCII inequalities (12) are clearly nonlinear. In the next section, we describe two ways of linearizing inequalities (12). The first linearization extends to the VRPTF a similar linearization proposed for the CmRSP, whereas the second one is new and it is based on mixed integer optimization.

3.1. Linearized versions of inequalities RCII

A first family of valid inequalities can be obtained using the following lemma, proposed by Baldacci et al. (2007).

LEMMA 1. Let m,n and o be three non-negative integer values with $m>o$ and $mod(m,o)\neq 0$:

$$\left\lceil \frac{m-n}{o} \right\rceil \geq \left\lceil \frac{m}{o} \right\rceil - \frac{n}{mod(m,o)}. \tag{13}$$

□

The term $\sum_{i\in V_C(S)} q_i y_i + \sum_{(i,j)\in A: j\in V_F(S)} q_i z_{ij}$ of RCII inequalities (12) can be rewritten as:

$$q(V_C) - \left(\sum_{i\in V_C(\overline{S})} q_i y_i + \sum_{(i,j)\in A: j\in V_F(\overline{S})} q_i z_{ij} \right) \tag{14}$$

and by using Lemma 1, from expression (14) we obtain the following inequality valid for any $S\subseteq V'$, $S\neq\emptyset$:

$$\sum_{e\in\delta(S)} \frac{1}{2} x_e \geq \left\lceil \frac{q(V_C)}{Q} \right\rceil - \frac{1}{mod(q(V_C),Q)} \left(\sum_{i\in V_C(\overline{S})} q_i y_i + \sum_{(i,j)\in A: j\in V_F(\overline{S})} q_i z_{ij} \right), \tag{15}$$

hereafter called RCII-a inequalities.

The term $\sum_{i\in V_C(S)} q_i y_i + \sum_{(i,j)\in A: j\in V_F(S)} q_i z_{ij}$ of RCII inequalities (12) can also be rewritten as:

$$q(V_C(S)) - \left(\sum_{\substack{(i,j)\in A:\\ i\in V_C(S), j\in V_F(\overline{S})}} q_i z_{ij} - \sum_{\substack{(i,j)\in A:\\ i\in V_C(\overline{S}), j\in V_F(S)}} q_i z_{ij} \right), \tag{16}$$

and by using Lemma 1 and by disregarding the term $\sum_{\substack{(i,j)\in A:\\ i\in V_C(\overline{S}), j\in V_F(S)}} q_i z_{ij}$ from (16) we get:

$$\sum_{e\in\delta(S)} \frac{1}{2} x_e \geq \left\lceil \frac{q(V_C(S))}{Q} \right\rceil - \frac{1}{mod(q(V_C(S)),Q)} \sum_{\substack{(i,j)\in A:\\ i\in V_C(S), j\in V_F(\overline{S})}} q_i z_{ij}, \tag{17}$$

hereafter called RCII-b inequalities.

Proposition 1 of the e-companion to this paper shows that there are no dominance relations between inequalities RCII-a and RCII-b.

The following lemma is based on mixed integer optimization. For a number $m \in \mathbb{R}$, define $\hat{m} = m - \lfloor m \rfloor$ to be its fractional part.

Lemma 2. Let $o \in \mathbb{R}$ with $\hat{o} > 0$ and $T = \{m \in \mathbb{R}, n \in \mathbb{Z} : m + n \geq o, m \geq 0\}$. The following inequality is valid for T:

$$m + \hat{o}n \geq \hat{o}\lceil o \rceil. \tag{18}$$

Proof. The proof is provided in the e-companion to this paper. □

Based on the above lemma, a second family of valid inequalities for the VRPTF can be obtained using the following theorem.

Theorem 1. Let $\alpha_e \geq 0, \forall e \in E$, $\beta_i \geq 0, \forall i \in V'$ and $\gamma_{ij} \geq 0, \forall (i,j) \in A$ and consider the following inequality valid for formulation TI:

$$\sum_{e \in E} \alpha_e x_e + \sum_{i \in V'} \beta_i y_i + \sum_{(i,j) \in A} \gamma_{ij} z_{ij} \geq o \tag{19}$$

where $o \in \mathbb{R}$ and $\hat{o} > 0$. Then the following inequality:

$$\sum_{e \in E} \varphi^o(\alpha_e) x_e + \sum_{i \in V'} \varphi^o(\beta_i) y_i + \sum_{(i,j) \in A} \varphi^o(\gamma_{ij}) z_{ij} \geq \lceil o \rceil \tag{20}$$

where $\varphi^o(m) = \lfloor m \rfloor + \min\left\{\frac{\hat{m}}{\hat{o}}, 1\right\}$, $m \in \mathbb{R}$, $n \in \mathbb{R}$, $\hat{o} > 0$, is also a valid inequality for formulation TI.

Proof. The proof is provided in the e-companion to this paper. □

Notice that, inequality (19) can be scaled by a rational number t thus obtaining the following valid inequality for formulation TI:

$$\sum_{e \in E} \varphi^{to}(t\alpha_e) x_e + \sum_{i \in V'} \varphi^{to}(t\beta_i) y_i + \sum_{(i,j) \in A} \varphi^{to}(t\gamma_{ij}) z_{ij} \geq \lceil to \rceil. \tag{21}$$

Starting from inequalities (4) and substituting the right-and side according to expressions (14) and (16) we get:

$$\sum_{e \in \delta(S)} \frac{1}{2} x_e + \sum_{i \in V_C(\overline{S})} \frac{q_i}{Q} y_i + \sum_{(i,j) \in A : j \in V_F(\overline{S})} \frac{q_i}{Q} z_{ij} \geq \frac{q(V_C)}{Q}, \tag{22}$$

and

$$\sum_{e \in \delta(S)} \frac{1}{2} x_e + \sum_{\substack{(i,j) \in A: \\ i \in V_C(S), j \in V_F(\overline{S})}} \frac{q_i}{Q} z_{ij} - \sum_{\substack{(i,j) \in A: \\ i \in V_C(\overline{S}), j \in V_F(S)}} \frac{q_i}{Q} z_{ij} \geq \frac{q(V_C(S))}{Q}. \tag{23}$$

First of all, notice that for $m,n \in \mathbb{R}$, $mod(m,n) = n((m/n) - \lfloor m/n \rfloor)$. Then, by setting $o = \frac{q(V_C)}{Q}$ and as $\varphi^o(\frac{1}{2}) = \min\left\{\frac{Q}{2mod(q(V_C),Q)}, 1\right\}$ and $\varphi^o(\frac{q_i}{Q}) = \min\left\{\frac{q_i}{mod(q(V_C),Q)}, 1\right\}$, $\forall i \in V_C$, from Theorem 1 and inequality (22) we obtain the following valid inequality:

$$\sum_{e\in\delta(S)} \min\left\{\frac{Q}{2\,mod(q(V_C),Q)}, 1\right\} x_e \geq \left\lceil \frac{q(V_C)}{Q} \right\rceil - \sum_{i\in V_C(\overline{S})} \min\left\{\frac{q_i}{mod(q(V_C),Q)}, 1\right\} y_i - \sum_{(i,j)\in A: j\in V_F(\overline{S})} \min\left\{\frac{q_i}{mod(q(V_C),Q)}, 1\right\} z_{ij}. \tag{24}$$

Also from Theorem 1, by disregarding the negative term of inequality (23) we obtain:

$$\sum_{e\in\delta(S)} \min\left\{\frac{Q}{2\,mod(q(V_C(S)),Q)}, 1\right\} x_e \geq \left\lceil \frac{q(V_C(S))}{Q} \right\rceil - \sum_{\substack{(i,j)\in A:\\ i\in V_C(S), j\in V_F(\overline{S})}} \min\left\{\frac{q_i}{mod(q(V_C(S)),Q)}, 1\right\} z_{ij}. \tag{25}$$

We call inequalities (24) and (25) RCII-c and RCII-d inequalities, respectively. Inequalities RCII-c and RCII-d are stronger than the pure integer rounding inequalities obtained from inequalities (22) and (23). In addition, notice that the coefficients of variables $\{x_e\}$ in both inequalities (24) and (25) are greater than 0.5 and less than or equal to 1. If $q_i = 1$, $\forall i \in V_C$, inequalities RCII-a and RCII-b dominate inequalities RCII-c and RCII-d. In general, no dominance relations exist among the four types of inequalities RCII-a, RCII-b, RCII-c and RCII-d.

4. Lower bounds based on a Set-Partitioning (SP) formulation

In this section, we first describe a Set-Partitioning (SP) based formulation for the VRPTF. Then, we investigate lower bounds based on the LP-relaxation of formulation SP. We introduce a theorem that is used to derive two dual ascent heuristics to find near-optimal dual solutions of the LP-relaxation of the SP model. Then, we describe how the valid inequalities described for the TI formulation in the previous sections can be used for strengthening the value of the LP-relaxation of formulation SP. Finally, we derive some properties of the LP-relaxation of formulation SP.

Let $\mathscr{R}$ be the index set of all feasible routes. Given a route $\ell \in \mathscr{R}$, we denote with R_ℓ the sequence $(i_1 = 0, i_2, \ldots, i_r = 0)$ of the nodes visited by the route and with $V_C(R_\ell)$ and $V_F(R_\ell)$ the sets $V_C \cap V(R_\ell)$ and $V_F \cap V(R_\ell)$, respectively. In addition, $V_A(R_\ell)$ denotes the customers of the route assigned to facilities in $V_F(R_\ell)$. Let $a_{i\ell}$ be a (0-1) binary coefficient equal to 1 if node $i \in V(R_\ell)$, 0 otherwise. In addition, let $b^j_{i\ell}$ be a (0-1) binary coefficient equal to 1 if customer $i \in V_A(R_\ell)$ is assigned to node $j \in V_F(R_\ell)$, 0 otherwise. Given a route ℓ, we denote with c_ℓ its routing cost computed as $\sum_{h=2}^{|R_\ell|} r_{\{i_{h-1},i_h\}}$, and with p_ℓ its assignment cost computed as $\sum_{j\in V_F(R_\ell)} \sum_{i\in V_A(R_\ell)} b^j_{i\ell} d_{ij}$. Let ξ_ℓ,

$\ell \in \mathscr{R}$, be a (0-1) binary variable equal to 1 if and only if route ℓ is in the optimal solution. Formulation SP is as follows:

$$(SP) \quad \min \sum_{\ell \in \mathscr{R}} (c_\ell + p_\ell)\xi_\ell \tag{26}$$

$$s.t. \sum_{\ell \in \mathscr{R}} \overline{a}_{i\ell}\xi_\ell = 1, \quad \forall i \in V_C \tag{27}$$

$$\sum_{\ell \in \mathscr{R}} a_{i\ell}\xi_\ell \leq 1, \quad \forall i \in V_F \tag{28}$$

$$\xi_\ell \in \{0,1\}, \qquad \forall \ell \in \mathscr{R}, \tag{29}$$

where $\overline{a}_{i\ell} = a_{i\ell} + \sum_{j \in V_F(R_\ell)} b^j_{i\ell}$, $i \in V_C$, $\ell \in \mathscr{R}$. In the formulation, constraints (27) and (28) impose that each customer is assigned exactly once and each facility is visited at most once, respectively.

We denote by LSP the LP-relaxation of formulation SP and by DSP the dual of LSP. The variables of DSP are given by the vector $\mathbf{u} = \{u_1, \ldots, u_{|V_C|}, u_{|V_C|+1}, \ldots, u_{|V'|}\}$, where $u_1, \ldots, u_{|V_C|}$ are associated with constraints (27), and $u_{|V_C|+1}, \ldots, u_{|V'|}$, with constraints (28). In the following, we denote with $q^{min} = \min_{i \in V_C}\{q_i\}$. The following theorem holds.

Theorem 2. Let us associate penalties $\lambda_i \in \mathbb{R}$, $\forall i \in V_C$, with constraints (27), and $\lambda_i \leq 0$, $\forall i \in V_F$, with constraint (28). Let $\mathscr{R}_i = \{\ell \in \mathscr{R} : \overline{a}_{i\ell} > 0\}$. For each $i \in V_C$ compute:

$$\nu_i = q_i \min_{\ell \in \mathscr{R}_i} \left\{ \frac{(c_\ell + p_\ell) - \sum_{j \in V_C} \overline{a}_{j\ell}\lambda_j - \sum_{j \in V_F} a_{j\ell}\lambda_j}{\sum_{j \in V_C} \overline{a}_{j\ell} q_j} \right\}. \tag{30}$$

A feasible DSP solution $\mathbf{u}$ of cost $z(DSP(\boldsymbol{\lambda}))$ is given by the following expressions:

$$u_0 = 0 \quad \text{ and } u_i = \nu_i + \lambda_i, \forall i \in V_C, \quad \text{ and } u_i = \lambda_i, \forall i \in V_F. \tag{31}$$

Proof. The proof is provided in the e-companion to this paper. □

The pricing problem associated with formulation SP is a strongly $\mathcal{NP}$–hard problem, since it requires finding minimum cost elementary routes over a graph with both positive and negative edge and arc costs. In the special case where $V_F = \emptyset$, the pricing problem consists of finding capacitated elementary cycles, a strongly $\mathcal{NP}$-hard problem (see Poggi and Uchoa 2014).

Therefore, in practice we enlarge the set of routes $\mathscr{R}$ to contain also *nonnecessarily elementary* routes, i.e., coefficients $\overline{a}_{i\ell}$ are general nonnegative integers, thus a node can be visited in a route more than once and/or a customer can be assigned more than once to facilities of the routes. Although non-elementary routes are infeasible, this relaxation has the advantage that the pricing subproblem becomes solvable in pseudo-polynomial time (by dynamic programming). Moreover, Theorem 2 remains valid if the set of routes $\mathscr{R}$ is enlarged to contain also nonnecessarily elementary routes.

In Section 5, we introduce two route relaxations called q-$*$route and ng-$*$route, used by two dual ascent heuristics based on Theorem 2 to find near-optimal solutions of problem DSP. q-$*$route and ng-$*$route relaxations are based on route relaxations already proposed for the CVRP and on the observation that given a route $R_\ell = (i_1 = 0, i_2, \dots, i_r = 0)$, a lower bound on its cost $c_\ell + p_\ell$ can be computed as $\sum_{h=2}^{|R_\ell|} r_{\{i_{h-1}, i_h\}} + \sum_{j \in V_F(R_\ell)} lb_j$, where $lb_j \le \sum_{i \in V_A(R_\ell)} b^j_{i\ell} d_{ij}$. Each value lb_j, $j \in V_F(R_\ell)$, can be computed as the minimum of the costs of all possible assignments of facility j involving customers in $\{i :\in V_C : j \in F_i\}$ with a total load $q = \sum_{i \in V_A(R_\ell)} b^j_{i\ell} q_i$.

Formulation LSP can be strengthened by adding valid inequalities derived for the TI formulation as follows. For each $\ell \in \mathscr{R}$, let coefficients η^ℓ_e be defined as follows: if ℓ is a route covering node h only, then $\eta^\ell_{\{0,h\}} = 2$ and $\eta^\ell_{\{i,j\}} = 0$, $\forall \{i,j\} \in E \setminus \{0,h\}$; if ℓ is not a single-node route, then $\eta^\ell_{\{i,j\}} = 1$ for each edge $\{i,j\}$ traversed by route R_ℓ, and $\eta^\ell_{\{i,j\}} = 0$ otherwise.

Any feasible solution ξ of SP can be transformed into a feasible TI solution (x, z, w) by setting:

$$x_e = \sum_{\ell \in \mathscr{R}} \eta^\ell_e \xi_\ell, \quad \forall e \in E, \tag{32}$$

$$z_{ij} = \sum_{\ell \in \mathscr{R}} b^j_{i\ell} \xi_\ell, \quad \forall (i,j) \in A, \tag{33}$$

$$y_i = \sum_{\ell \in \mathscr{R}} a_{i\ell} \xi_\ell = 1 - \sum_{j \in F_i} \sum_{\ell \in \mathscr{R}} b^j_{i\ell} \xi_\ell, \quad \forall i \in V_C, \text{ and} \tag{34}$$

$$y_i = \sum_{\ell \in \mathscr{R}} a_{i\ell} \xi_\ell, \quad \forall i \in V_F. \tag{35}$$

The following theorem shows that any feasible solution of formulation LSP already satisfies some valid inequalities derived from formulation TI.

THEOREM 3. The LP-relaxation of the SP formulation satisfies both CI and FrCI inequalities, and a weak form of MI inequalities.

Proof. The proof is provided in the e-companion to this paper. □

5. Bounding procedure

This section presents a method for computing a lower bound on the VRPTF which combines in sequence two dual ascent heuristics (see Section 5.1), and a column-and-cut generation method (see Section 5.2), all based on formulation LSP.

5.1. Dual ascent heuristics

The dual ascent heuristics are based on Theorem 2 where the set of routes $\mathscr{R}$ is enlarged with set $\mathscr{R}^>$ containing also nonnecessarily elementary routes (i.e., $\mathscr{R}^> \supseteq \mathscr{R}$). In particular, two different route relaxations are used, called q-$*$route and ng-$*$route, to compute lower bounds LB_1 and LB_2

on the VRPTF, respectively. The two dual ascent heuristics are based on a column generation-like method, called CG for solving the following problem:

$$LCG = \max_{\boldsymbol{\lambda}} \{z(DSP(\boldsymbol{\lambda}))\}. \tag{36}$$

CG executes a number of macro-iterations to compute a dual solution $\mathbf{u}$ of the master problem DSP, defined by the route subset $\overline{\mathscr{R}} \subseteq \mathscr{R}^{>}$, and then CG solves problem (36) with a predefined number $Maxit2$ of subgradient iterations to modify the penalty vector $\boldsymbol{\lambda}$.

5.1.1. Route relaxation *q*-∗route q-∗routes are based on the q-path relaxation proposed by Christofides et al. (1981). We define a q-∗path as a nonnecessarily elementary partial route in G from depot 0 to node $i \in V'$ with a load equal to q. In a q-∗path a node $i \in V'$ can be visited more than once and a customer $i \in V_C$ can be assigned more than once. In the following, we describe a dynamic programming algorithm for computing q-∗paths, with the restriction that a q-∗path can not contain loops formed by three consecutive nodes. Let $f(q,i)$ be the cost of the least cost q-∗path from node 0 to node i and let $\pi(q,i)$ be the node immediately before i in the least cost path of value $f(q,i)$. Let $g(q,i)$ be the cost of the least cost q-∗path from node 0 to node i, such that $\gamma(q,i) \neq \pi(q,i)$, where $\gamma(q,i)$ is the node immediately before i in the least cost path corresponding to $g(q,i)$. For a given value of q, let $h(i,j)$ be the cost of the least cost q-∗path from 0 to j, with $i \in V'$ just before j and without loops. In addition, for each facility $k \in V_F$, let $lb_k(q)$ be a lower bound on the assignment cost of any assignment of load q of customers to the facility k. $lb_k(q)$, for each $k \in V_F$ and $q^{min} \leq q \leq Q$, can be computed as the optimal solution cost of the following knapsack problem $KP(q,k)$:

$$(KP(q,k)) \quad lb_k(q) = \min \sum_{i \in V_C : k \in F_i} d_{ik} \chi_i \tag{37}$$

$$s.t. \sum_{i \in V_C : k \in F_i} q_i \chi_i = q \tag{38}$$

$$\chi_i \in \{0,1\}, \qquad \forall i \in V_C : k \in F_i. \tag{39}$$

We assume that $lb_k(q) = \infty$ if problem $KP(q,k)$ does not admit a feasible solution for the given pair q and k. For each $q = q^{min}, \ldots, Q$ and $i,j \in V'$, $i \neq j$, compute:

$$h(i,j) = \begin{cases} \begin{cases} f(q-q_j,i) + r_{\{i,j\}}, & \text{if } \pi(q-q_j,i) \neq j \\ g(q-q_j,i) + r_{\{i,j\}}, & \text{otherwise.} \end{cases} & , j \in V_C \\ \min_{q^{min} \leq w \leq Q} \begin{cases} f(q-w,i) + r_{\{i,j\}} + lb_j(w), & \text{if } \pi(q-w,i) \neq j \\ g(q-w,i) + r_{\{i,j\}} + lb_j(w), & \text{otherwise.} \end{cases} & , j \in V_F \end{cases} \tag{40}$$

Then, compute:

$$\begin{cases} f(q,j) = \min_{i \in V' \setminus \{j\}} \{h(i,j)\} \\ \pi(q,j) = i' \end{cases} \tag{41}$$

where i' is the node producing the above minimum,

$$\begin{cases} g(q,j) = \min_{i\in V'\setminus\{j,i'\}}\{h(i,j)\} \\ \gamma(q,j) = i'' \end{cases} \tag{42}$$

where i'' is the node producing the above minimum. The functions are initialized as follows:

- $f(q_j, j) = r_{0j}$, $\pi(q_j, j) = 0$, $j \in V_C$;
- $f(q,j) = \infty$, $\pi(q,j) = 0$, $q = 0,\dots,Q$,$q \neq q_j$, $j \in V_C$;
- $f(q,j) = r_{0j} + lb_j(q)$, $\pi(q,j) = 0$, $q = 0,\dots,Q$, $j \in V_F$;
- $g(q,j) = \infty$, $\gamma(q,j) = 0$, $q = 0,\dots,Q$, $j \in V'$.

A q-$*$route is obtained from a q-$*$path ending in i by adding arc $(i,0)$.

5.1.2. Route relaxation ng-$*$route ng-$*$routes are based on the route relaxations proposed by Baldacci et al. (2011) for the CVRP. Let $N_i \subseteq V'$ be a set of selected nodes for node $i \in V'$ (according to some criterion) such that $N_i \ni i$ and $|N_i| \leq \Gamma$, where Γ is a parameter (e.g., $\Gamma = 5$, $\forall i \in V'$, and N_i contains i and the four nearest nodes to i).

With a forward path $P = (0, i_1, \dots, i_k)$, we associate a set $\Pi(P) \subseteq V'$ defined as:

$$\Pi(P) = \{i_r : i_r \in \bigcap_{s=r+1}^{k} N_{i_s}, r = 1,\dots,k-1\} \cup \{i_k\}. \tag{43}$$

A *forward* ng-$*$path (NG,q,i) is a non-necessarily elementary partial route $P = (0, i_1, \dots, i_{k-1}, i_k = i)$ starting from the depot with a load equal to q, ending at customer i, and such that $NG = \Pi(P)$, and $i \notin \Pi(P')$, where $P' = (0, i_1, \dots, i_{k-1})$. Let $f(NG,q,i)$ be the cost of a least-cost forward ng-$*$path (NG,q,i). The dynamic programming (DP) recursion for computing functions $f(NG,q,i)$ is defined on a state-space graph $\mathscr{H} = (\mathscr{E}, \Psi)$ defined as:

$$\begin{aligned} \mathscr{E} &= \{(NG,q,i) : q_i \leq q \leq Q, \forall NG \subseteq N_i \text{ s.t. } NG \ni i, \forall i \in V\} \\ \Psi &= \{((NG',q',j),(NG,q,i)) : \forall (NG',q',j) \in \Psi^{-1}(NG,q,i),\ \forall (NG,q,i) \in \mathscr{E}\}, \end{aligned} \tag{44}$$

where $\Psi^{-1}(NG,q,i) = \{(NG', q - q_i, j) : \forall NG' \subseteq N_j \text{ s.t. } NG' \ni j \text{ and } NG' \cap N_i = NG \setminus \{i\},\ \forall j \in V \setminus \{i\}\}$, if $i \in V_C$, and $\Psi^{-1}(NG,q,i) = \{(NG',q',j) : 0 \leq q' \leq q - \min_{i\in V_C}\{q_i\}, \forall NG' \subseteq N_j \text{ s.t. } NG' \ni j \text{ and } NG' \cap N_i = NG \setminus \{i\},\ \forall j \in V \setminus \{i\}\}$, if $i \in V_F$.

The DP recursion for computing functions $f(NG,q,i)$, for each state $(NG,q,i) \in \mathscr{E}$ is as follows:

i) $i \in V_F$:$f(NG,q,i) = \min_{(NG',q',j)\in\Psi^{-1}(NG,q,i)}\{f(NG',q',j) + r_{\{j,i\}} + lb_i(q - q')\}$, $\forall (NG,q,i) \in \mathscr{E}$,

ii) $i \in V_C$: $f(NG,q,i) = \min_{(NG',q',j)\in\Psi^{-1}(NG,q,i)}\{f(NG',q',j) + r_{\{j,i\}}\}$, $\forall (NG,q,i) \in \mathscr{E}$,

where functions $lb_i(q)$ are computed as described in Section 5.1.1 and the initialization $f(\{0\},0,0) = 0$ and $f(\{0\},q,0) = \infty$, $\forall\, 0 < q \leq Q$ is required. We define a ng-$*$route as a route obtained by adding, to an ng-$*$path (NG,q,i), edge $e = \{0,i\}$; the cost of an ng-$*$route is equal to the cost of ng-$*$path (NG,q,i) plus r_e.

5.1.3. Procedure CG Let $\overline{\mathscr{R}} \subseteq \mathscr{R}^{>}$ be a subset of routes satisfying a given route relaxation. Moreover, given a route ℓ, we denote with $q(R_\ell) = \sum_{i \in V_C(R)} q_i + \sum_{i \in V_A(R_\ell)} q_i$ its load. Procedure CG works as follows.

Step 1. *Initialization.* Generate a route set $\overline{\mathscr{R}}$ to initialize the master problem which corresponds to LSP, where $\mathscr{R}$ is replaced with $\overline{\mathscr{R}}$. We assume that $\overline{\mathscr{R}}$ contains at least one route containing each customer $i \in V_C$. Set $LCG = 0$ and $iter = 1$.

Step 2. *Find a master dual solution* $\overline{\mathbf{u}}$ *of cost* $\overline{z}$. Initializes $\overline{z} = 0$ and performs $Maxit2$ iterations of the following operations.

(i) Compute a dual solution $\mathbf{u}$ of the master of cost z by means of expressions (30) and (31), where $\mathscr{R}$ is replaced with $\overline{\mathscr{R}}$ and by using the current vector $\boldsymbol{\lambda}$. Let $\tilde{\mathscr{R}}$ be the index set of routes producing ν_i, $i \in V_C$, in expressions (30), and let $\ell(i)$ be the index of the route in $\tilde{\mathscr{R}}$ associated with ν_i, $i \in V_C$. Define a non-necessarily feasible solution $\boldsymbol{\xi}$ of LSP as $\xi_\ell = \sum_{i \in V_C} \overline{a}_{i\ell} \frac{q_i}{q(R_\ell)} \zeta_\ell^i$, $\ell \in \tilde{\mathscr{R}}$, by setting $\zeta_{\ell(i)}^i = 1$ and $\zeta_\ell^i = 0$, $\forall \ell \in \tilde{\mathscr{R}} \setminus \{\ell(i)\}$, $\forall i \in V_C$. If $z > \overline{z}$, update $\overline{z} = z$, $\overline{\boldsymbol{\xi}} = \boldsymbol{\xi}$, $\overline{\mathbf{u}} = \mathbf{u}$.

(i) Update the penalty vectors $\boldsymbol{\lambda}$ as follows. Compute $\alpha_i = \sum_{\ell \in \tilde{\mathscr{R}}} \overline{a}_{i\ell} \xi_\ell$, $i \in V_C$, and $\alpha_i = \sum_{\ell \in \tilde{\mathscr{R}}} a_{i\ell} \xi_\ell$, $i \in V_F$. Then, vector $\boldsymbol{\lambda}$ is modified as follows: $\lambda_i = \lambda_i - \epsilon\gamma(\alpha_i - 1), i \in V_C$, and $\lambda_i = \min\{0, \lambda_i - \epsilon\gamma(\alpha_i - 1)\}, i \in V_F$. where ϵ is a positive constant and $\gamma = \frac{0.2\overline{z}}{\sum_{i \in V'} (\alpha_i - 1)^2}$.

Step 3. *Check if* $\overline{\mathbf{u}}$ *is a feasible DSP solution.* Generate the largest subset $\mathscr{N} \subseteq \mathscr{R}^{>}$ of routes having negative reduced cost with respect to the current dual master solution $\mathbf{u}$ and such that $|\mathscr{N}| \leq \Delta$ (Δ is an a priori defined parameter). If $\mathscr{N} = \emptyset$ and $\overline{z}$ is greater than LCG, then $LCG = \overline{z}$, $\mathbf{u}^* = \overline{\mathbf{u}}$, $\boldsymbol{\xi}^* = \overline{\boldsymbol{\xi}}$ and $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}$; otherwise, $\overline{\mathscr{R}} = \overline{\mathscr{R}} \cup \mathscr{N}$ is updated.

Step 4. *Termination criterion.* Set $iter = iter + 1$. If $iter = Maxit1$, stop.

Computing lower bound LB_1 Lower bound LB_1 corresponds to lower bound LCG computed by procedure CG using q-$*$route relaxation. The initial route set $\overline{\mathscr{R}}$ of the master problem contains a feasible solution generated with the heuristic algorithm described in 6.1. We initialize $\boldsymbol{\lambda} = \mathbf{0}$.

Define the modified routing cost $\overline{r}_{\{i,j\}} = r_{\{i,j\}} - (1/2)(\overline{u}_i + \overline{u}_j)$, $\forall \{i,j\} \in E$ (we assume $\overline{u}_0 = 0$), and the modified assignment cost $\overline{d}_{ij} = d_{ij} - \overline{u}_i$, $\forall (i,j) \in A$, with respect to the current dual solution $\overline{\mathbf{u}}$. The set $\mathscr{N}$ is computed as follows. We compute functions $lb_k(q)$, $f(q,i)$ and $g(q,i)$ using the modified routing and assignment costs $\overline{r}_{\{i,j\}}$ and $\overline{d}_{ij}$ instead $r_{\{i,j\}}$ and d_{ij}. Let $h(i) = \min_{q_i \leq q \leq Q}\{f(q,i) + \overline{r}_{\{0,i\}}\}$, if $\forall i \in V_C$, and $h(i) = \min_{q^{min} \leq q \leq Q}\{f(q,i) + \overline{r}_{\{0,i\}}\}$, $\forall i \in V_F$. The set $\mathscr{N}$ contains any q-$*$route corresponding to $h(i) < 0$, $i \in V'$. Set $\mathbf{u}^1 = \mathbf{u}^*$, $\boldsymbol{\lambda}^1 = \boldsymbol{\lambda}^*$, and $LB_1 = LCG$.

Computing lower bound LB_2 Lower bound LB_2 corresponds to lower bound LCG computed by procedure CG using ng-$*$route relaxation.

We initialize $\boldsymbol{\lambda} = \boldsymbol{\lambda}^1$, define $r^1_{\{i,j\}} = r_{\{i,j\}} - (1/2)(u^1_i + u^1_j)$, $\forall \{i,j\} \in E$ (we assume $u^1_0 = 0$), $d^1_{ij} = d_{ij} - u^1_i$, $\forall (i,j) \in A$, and compute N_i to be the Γ nearest nodes to i according to $r^1_{\{i,j\}}$. We compute functions $f(NG,q,i)$ and $lb_k(q)$ using $r^1_{\{i,j\}}$ and d^1_{ij} instead of $r_{\{i,j\}}$ and d_{ij}, respectively, and the costs $h(i) = min_{(NG,q,i)\in\mathscr{E}}\{f(NG,q,i) + r^1_{\{0,i\}}\}$, of the least cost ng-$*$route visiting i immediately before arriving at the depot. The initial route set $\overline{\mathscr{R}}$ contains the ng-$*$routes corresponding to $h(i) < 0$, $i \in V'$. At each iteration of procedure CG, to generate the set $\mathscr{N}$, we compute functions $f(NG,q,i)$ and $lb_k(q)$ with the modified routing cost $\overline{r}_{\{i,j\}} = r_{\{i,j\}} - (1/2)(\overline{u}_i + \overline{u}_j)$, $\forall \{i,j\} \in E$, and the modified assignment cost $\overline{d}_{ij} = d_{ij} - \overline{u}_i$, $\forall (i,j) \in A$, with respect to the current solution $\overline{\mathbf{u}}$. $\mathscr{N}$ contains every ng-$*$route corresponding to $h(i) = min_{(NG,q,i)\in\mathscr{E}}\{f(NG,q,i) + \overline{r}_{\{0,i\}}\} < 0$, $i \in V'$. Set $LB_2 = LCG$.

5.2. Column-and-cut generation method

In this section, we describe a bounding procedure that computes a lower bound on the VRPTF as the cost of an optimal solution of problem $\overline{LSP}$ obtained from formulation LSP by substituting the route set $\mathscr{R}$ with the set $\mathscr{R}^{>}$ of ng-$*$route and by adding valid inequalities derived from a family $\mathscr{F}$ of valid inequalities described for formulation TI.

Any valid inequality $t \in \mathscr{F}$ can be expressed in general form as

$$\sum_{e\in E} \alpha^t_e x_e + \sum_{i\in V'} \beta^t_i y_i + \sum_{(i,j)\in A} \gamma^t_{ij} z_{ij} \geq \omega^t, \tag{45}$$

and can be transformed into the following valid inequality for formulation SP using equations (32)-(35), where $\mathscr{R}$ is substituted by $\mathscr{R}^{>}$:

$$\sum_{\ell\in\mathscr{R}^{>}} (\varphi^t_\ell + \phi^t_\ell + \psi^t_\ell)\xi_\ell \geq \omega^t, \tag{46}$$

where $\varphi^t_\ell = \sum_{e\in E} \alpha^t_e \eta^\ell_e$, $\phi^t_\ell = \sum_{i\in V'} \beta^t_i a_{i\ell}$ and, $\psi^t_\ell = \sum_{(i,j)\in A} \gamma^t_{ij} b^j_{i\ell}$.

The bounding procedure solves problem $\overline{LSP}$ by using column and cut generation. The initial master problem is obtained from the computation of lower bound LB_2 by replacing the route set $\mathscr{R}^{>}$ with the route set $\overline{\mathscr{R}}$ generated by procedure CG during the computation of LB_2. The initial set of valid inequalities $\overline{\mathscr{F}}$ is set to the empty set. At each iteration (say k), the procedure performs the following steps.

1. Solve problem $\overline{LSP}$. Let $\overline{\boldsymbol{\xi}}$ and $(\overline{\mathbf{u}}, \overline{\mathbf{v}})$ be the optimal primal and dual solutions, respectively. Vector $\overline{\mathbf{u}}$ is given by $\overline{\mathbf{u}} = \{\overline{u}_1, \ldots, \overline{u}_{|V_C|}, \overline{u}_{|V_C|+1}, \ldots, \overline{u}_{|V'|}\}$, where $\overline{u}_1, \ldots, \overline{u}_{|V_C|}$ are associated with constraints (27), and $\overline{u}_{|V_C|+1}, \ldots, \overline{u}_{|V'|}$, with constraints (28). Vector $\overline{v} = \{\overline{v}_1, \ldots, \overline{v}_{|\overline{\mathscr{F}}|}\}$ is associated with the family of valid inequalities $\overline{\mathscr{F}}$.

2. Generate the largest subset $\mathscr{N} \subseteq \mathscr{R}^{>}$ of ng-$*$route having negative reduced cost with respect to the current dual master solution $(\overline{\mathbf{u}}, \overline{\mathbf{v}})$ and such that $|\mathscr{N}| \leq \Delta$ (Δ is an a priori defined parameter). If $\mathscr{N} = \emptyset$, the procedure terminates; otherwise a new iteration is made. At iteration $k+1$, the procedure solves a new master problem $\overline{LSP}$ by replacing $\overline{\mathscr{R}}$ with $\overline{\mathscr{R}} \cup \mathscr{N}$ and the valid inequalities of $\mathscr{F}$ violated by the $\overline{LSP}$ solution $\overline{\xi}$ achieved by iteration k.
3. Given the solution vector $\overline{\boldsymbol{\xi}}$, compute the corresponding solution vector $(\overline{\mathbf{x}}, \overline{\mathbf{z}}, \overline{\mathbf{w}})$ by means of equations (32)-(35) where $\mathscr{R}$ is substituted by $\overline{\mathscr{R}}$. Solve the separation problems associated with the set of valid inequalities $\mathscr{F}$ (see below) and add, if any, violated inequalities to set $\overline{\mathscr{F}}$.

It can be easily shown that the complexity of the pricing algorithm solved at Step 2 of the above procedure is not sensitive to the addition of the valid inequalities in $\overline{\mathscr{F}}$, since the values of the corresponding dual variables can be translated into subproblem costs. Indeed, at each iteration of the procedure, to generate the set $\mathscr{N}$, we compute the ng-$*$route functions $f(NG, q, i)$ and $lb_k(q)$ with the modified routing cost $\overline{r}_{\{i,j\}} = r_{\{i,j\}} - (1/2)(\overline{u}_i + \sum_{t \in \overline{\mathscr{F}}} \beta_i^t \overline{v}_t) - (1/2)(\overline{u}_j + \sum_{t \in \overline{\mathscr{F}}} \beta_j^t \overline{v}_t) - \sum_{t \in \overline{\mathscr{F}}} \alpha_{\{i,j\}}^t \overline{v}_t$, $\forall \{i,j\} \in E$, and the modified assignment cost $\overline{d}_{ij} = d_{ij} - \overline{u}_i - \sum_{t \in \overline{\mathscr{F}}} \gamma_{ij}^t \overline{v}_t$, $\forall (i,j) \in A$, with respect to the current dual solution $(\overline{\mathbf{u}}, \overline{\mathbf{v}})$ (we assume $\overline{u}_0 = 0$). $\mathscr{N}$ contains every ng-$*$route corresponding to $h(i) < 0$, $i \in V'$.

We conducted preliminary experiments to identify a good separation strategy to be used at Step 3. As a result of our experimentation, we decided to use the following inequalities to define the family set $\mathscr{F}$: CI, MI, RCI, RCII-a, RCII-b, RCII-c, and RCII-d inequalities. For a given solution $(\overline{\mathbf{x}}, \overline{\mathbf{z}}, \overline{\mathbf{w}})$, we identified (as far as possible) violated inequalities of above seven types by applying the corresponding separation procedures as described below.

5.2.1. Separation procedures The separation problems of CI, RCII-a and RCII-c inequalities can be reduced to max-flow/min-cut problems using a standard construction, and therefore solved in polynomial time; we omit the details for sake of brevity (see Baldacci et al. (2007)). Concerning MI inequalities, the following theorem holds.

Theorem 4. Let (x, z, y) be a solution of the LP-relaxation of formulation TI and assume that $q_i \leq Q$, $\forall i \in V_C$, and that $x_e = 0$, $e = \{i,j\} \in E \setminus \{\{0,h\} : h \in V'\}$, if $q_i + q_j > Q$. The separation problem for MI inequalities (10) is solvable in polynomial time.

Proof. The proof is provided in the e-companion to this paper. □

RCI, RCII-b and RCII-d inequalities are separated using a heuristic separation procedure. The procedure is a Multistart Local Search that, at each iteration, generates a starting point and evolves it through a Local Search procedure. We start by generating a set $\mathscr{S}$ of $10(n-1)$ subsets of V' as follows. For the RCI inequalities the first $|V_C|$ sets of $\mathscr{S}$ are obtained by inserting in each set, for $i = 1, \ldots, |V_C|$, the nodes in F_i. The remaining sets are generated by first computing a random number m drawn from a uniform distribution in $[1,\dots,n-1]$, and then by randomly selecting m different nodes of V', again using a uniform distribution. For the RCII-b and RCII-c inequalities all the sets are randomly generated as above. Each set $S \in \mathscr{S}$ is then iteratively expanded by adding one node at each iteration until $S = V'$. For a given set S, let $\theta(S)$ denote the difference between the left-hand side and the right-hand side value of the considered inequality (i.e., the inequality can be rewritten as $\theta(S) \geq 0$ and the separation problem corresponds to compute $\arg\min_{S\subseteq V'}\{\theta(S)\}$). Each set S is expanded by choosing the node $i \in V' \setminus S$ such that $\theta(S \cup \{i\})$ is minimized

6. Solving the VRPTF to Optimality

In this section, we describe the method implemented for solving the VRPTF to optimality. We start by describing two heuristic algorithms that compute primal bounds used to initialize the exact method. The exact method is a branch-and-cut-and-price (BCP) solution method based on the SCIP (see Achterberg 2009) BCP solution framework.

6.1. Heuristic algorithms

Primal bounds for the VRPTF are computed by means of two different types of heuristic algorithms: a *constructive heuristic* and a *Lagrangean heuristic*.

The basis of the constructive algorithm is a heuristic to solve the CVRP. Given an instance of VRPTF, we define a complete graph $\overline{G} = (\overline{V}, \overline{E})$ where the node set $\overline{V} = \{0\} \cup V_C$ contains the depot and the customer nodes. Each edge $e \in \overline{E}$ has a cost given by r_e. Each customer $i \in V_C$ has a demand equal to q_i and the capacity of the vehicles is set to Q. Roughly speaking, we solve a problem obtained from VRPTF by disregarding the facility nodes (set V_F) and the connection arcs (set A). The CVRP instance is solved through an iterative multistart procedure based on a cluster-first, route-second heuristic procedure. Each iteration consists of three phases: (i) determine a partition of the customers into a number of subsets each one satisfying the capacity constraint; (ii) for each set, find the route of a single vehicle that serves all the customers in the set (i.e. we solve an instance of a Traveling Salesman Problem (TSP)); (iii) locally optimize the solution obtained at step (ii). The CVRP solution so far obtained, is then locally optimized by iteratively applying two post-optimization procedures specifically devised for the VRPTF.

The Lagrangean heuristic is based on procedure CG described in Section 5.1.3. Procedure CG is interwoven with an algorithm that produces a feasible VRPTF solution using the route set $\tilde{\mathscr{R}}$ (see Step 2 of procedure CG). The route set $\tilde{\mathscr{R}}$ is first modified to contain only customers visited at most once. Then, unrouted customers are inserted in order to obtain a feasible solution. The solution obtained is further optimized by applying the same post-optimization procedures used by the constructive algorithm.

A step-by-step description of the heuristics are given in the e-companion to this paper.

6.2. Details of the BCP method

The lower bound at the root node of the enumeration tree is first computed by using the bounding procedure described in Section 5, then by using the column-and-cut generation method described in Section 5.2. The master problem at a generic node except the root node is initialized with the set of valid inequalities $\overline{\mathscr{F}}$ and the set of routes $\overline{\mathscr{R}}$ of the parent node, where set $\overline{\mathscr{R}}$ is further modified by extracting the largest set of routes satisfying the branching conditions.

To choose a node-selection rule, we first performed some preliminary experiments with different rules and, based on these results, we decided to adopt the *best-first strategy* for all the computations of Section 7. We did not implement primal heuristics but the algorithm was initialized with the best primal solution found by the two heuristic algorithms described in the previous section that are executed at the root node. We used the default branching scheme of the SCIP framework, namely the *hybrid branching* scheme (see Achterberg and Berthold 2009), that combines ideas from pseudocost branching (Benichou et al. 1971) and strong branching (Applegate et al. 2007).

7. Computational Results

This section reports on the computational results of the exact method described in this paper and analyses the effectiveness of the dual ascent heuristics and of the different types of inequalities on the bounding procedure procedure described in Section 5.

The algorithms were coded in C++ and linked with the SCIP 3.1.1 BCP solution framework (see Achterberg 2009) using the IBM Cplex 12.6.1 linear programming solver (see IBM CPLEX 2014). The experiments were performed on an Intel Core 2 Duo at 2.66 GHz personal computer equipped with 4 Gb of RAM.

The exact method has been tested on real-world instances and on instances derived from LRP instances already proposed in the literature, used to further evaluate the performance of our algorithms. The same instances have been also used to generate 2E-CVRP instances. The following sections 7.1 and 7.2 briefly describe the real-world and LRP based instances, respectively, and report on the results obtained by the different algorithms. The complete details of the instances are provided in the e-companion to this paper.

Based on the results of preliminary experiments to identify good parameter settings for our method, we decided to use the following settings for our bounding procedure (see Section 5):

- in computing lower bound LB_1: $Maxit1 = 50$, $Maxit2 = 50$, $\epsilon = 1.5$ and $\Delta = 50$;
- in computing lower bound LB_2: $\Gamma = 12$, $Maxit1 = 100$, $Maxit2 = 50$, $\epsilon = 2.0$ and $\Delta = 50$;
- in the column-and-cut-generation method: $\Delta = 100$ at the root node of the BCP whereas $\Delta = 50$ for the remaining nodes.

7.1. Results on real-world instances

The data of this set of instances were provided by a major Italian transportation company that distributes non-perishable products over the whole Italian peninsula. The company operates through three main distribution areas (*North*, *Centre* and *South*) using three main central depots located in the provinces of Milan, Rome and Naples.

The three distribution areas operate independently in the corresponding areas to serve customer orders using an existing set of intermediate facilities. The customer orders are placed into *Euro*-pallet and distributed either to the final customers or the intermediate facilities by means of a fleet of identical capacitated vehicles which are stationed at the different central depots and whose capacity is expressed in terms of pallets. All the facilities are owned by third-party contractors, that are in charge of delivering to the final customers the orders consolidated at the facilities.

The company was interested in analyzing different distribution scenarios associated with the three distribution areas. A total number of 18 instances were provided by the company, six instances per each area or depot. The following naming convention was adopted to identify the different instances. The instance name is a string **area_a**×**b**_Q**c**, where **area** represents the area (i.e., *North*, *Centre*, *South*), **a** represents the number of customers, **b** corresponds to the number of facilities, and **c** is the vehicle capacity.

In Table 1, we report the results obtained by the heuristic algorithms, the bounding procedure and the BCP method. The columns of the table report the instance name (*Name*), the cost of the best solution found by the heuristics and BCP algorithms (z^*), the percentage deviation of the upper bound computed by the constructive heuristic ($\%UB_1$), the percentage deviation of the upper bound computed by the lagrangean heuristic ($\%UB_2$), the percentage deviation of lower bound LB_1 ($\%LB_1$), the percentage deviation of lower bound LB_2 ($\%LB_2$), the total computing time of lower bounds LB_1 and LB_2 that also includes the time spent for computing UB_2 (t_{DA}), the percentage deviation of the lower bound LB computed at the root-node of the BCP algorithm and the corresponding computing time ($\%LB$, t_{LB}), the cardinality of the sets $\overline{\mathscr{F}}$ and $\overline{\mathscr{R}}$ associated with lower bound LB (#*cuts* and #*cols*), the total number of nodes of the exact algorithms (#N), the percentage deviation of the best lower bound achieved by the exact method (%*Opt*), and the total computing time in seconds spent by the exact method (t_{TOT}), that also include the time spent for computing upper bound UB_1. The percentage deviation of value x is computed as $100 \times x/z^*$.

Table 1 Results on real-world instances

Name	z^*	$\#r$	$\#f$	$\#c$	$\%UB_1$	$\%UB_2$	$\%LB_1$	$\%LB_2$	t_{DA}	$\%LB_C$	t_C	$\%LB$	t_{LB}	$\#cuts$	$\#cols$	#N	$\%Opt$	t_{TOT}
north-68x7-Q24	8890.3	23	6	27	106.8	102.2	97.8	97.9	51.7	95.5	38.7	99.1	2.3	18	1093	1930	100.0	565
north-68x7-Q34	9748.1	17	5	25	105.0	100.7	96.3	96.3	36.5	93.8	28.1	97.1	5.3	496	1535	4126	100.0	5426
north-103x13-Q24	14251.1	34	13	46	105.8	102.2	98.1	98.1	91.6	94.3	141.8	98.9	4.5	23	1260	9888	99.5	7466
north-103x13-Q34	15613.3	27	12	61	107.2	102.1	98.4	98.5	164.8	95.7	204.2	99.1	10.7	489	2786	6641	99.6	7497
north-142x18-Q24	17876.4	49	16	65	110.7	102.8	98.7	99.0	201.4	93.5	396.8	99.3	17.1	3	3490	3382	100.0	3172
north-142x18-Q34	19623.3	39	16	74	118.0	103.3	97.5	97.8	249.0	92.7	515.4	98.3	19.2	2	3984	1583	98.7	7765
centre-74x6-Q24	12213.8	24	6	21	106.6	100.7	99.6	99.9	47.5	95.7	53.0	99.9	2.6	494	1287	3	100.0	120
centre-74x6-Q34	12930.4	19	5	28	104.6	100.9	99.1	99.6	60.3	96.0	57.1	99.7	4.2	188	1628	92	100.0	205
centre-113x9-Q24	19612.1	38	8	41	105.4	102.0	99.2	99.3	260.9	94.8	319.5	99.5	11.3	4	2464	4954	100.0	2578
centre-113x9-Q34	21877.0	31	8	41	106.7	101.4	97.2	97.6	232.4	94.1	325.4	97.7	8.1	2	4283	1639	98.1	7595
centre-164x12-Q24	27390.2	56	12	46	108.3	102.3	99.0	99.1	656.5	93.2	915.2	99.3	41.2	11	4411	3863	99.4	8452
centre-164x12-Q34	29853.5	45	11	65	113.9	102.6	99.0	99.1	910.3	94.7	1493.3	99.4	32.9	14	6055	2087	99.5	8565
south-54x4-Q24	10987.6	19	4	16	102.1	100.1	97.7	98.2	33.2	96.4	21.5	98.8	0.8	2	678	4586	100.0	1328
south-54x4-Q34	12597.7	26	4	13	106.9	100.5	94.4	94.4	39.5	94.6	26.0	95.6	1.9	33	1200	1030	96.2	7255
south-85x7-Q24	16553.7	29	7	22	107.1	100.0	97.0	96.9	116.3	93.8	101.3	97.6	5.0	8	2220	2476	97.7	7421
south-85x7-Q34	18100.4	22	7	31	105.7	102.8	98.5	98.9	200.9	96.6	126.0	99.4	6.6	389	3013	1729	100.0	5470
south-115x9-Q24	20497.5	39	8	38	104.9	102.3	98.5	98.2	424.5	94.3	283.0	99.2	10.3	3	2726	9148	99.7	7878
south-115x9-Q34	21963.2	33	9	52	107.0	101.2	97.4	97.6	342.7	93.8	187.5	98.5	10.5	2	4685	1819	98.9	7737
					107.4	101.7	98.0	98.1	228.9	94.6	290.8	98.7	10.8					2358

In order to evaluate the quality of the different lower bounds, we also computed, for each instance, the value of the lower bound obtained by solving the LP-relaxation of formulation TI strengthened with the different valid inequalities (using the separation strategy described in Section 5.2). In the table, column $\%LB_C$ reports the percentage deviation of the final lower bound obtained whereas column t_C displays the corresponding computing time.

For each instance, Table 1 also reports the following details about the best solution found: the number of routes in the solution ($\#r$), the number of facilities visited ($\#f$) and the number of customers assigned to a facility ($\#c$).

For these set of instances, a time limit of 7,200 seconds was imposed to the SCIP framework.

The last row of the table reports averages computed over the different columns. The average reported under column t_{TOT} is computed over the instances solved to optimality within the imposed time limit. If a value of 100.0 is reported for column $\%Opt$, then the algorithm terminated with an optimal solution.

Table 1 shows that 8 out of 18 instances were solved to optimality and that the final lower bound LB is on average quite tight, being equal to 98.7%. The largest instance solved to optimality involves 142 customers and 18 facilities. On these set of instances, lower bounds LB_1 and LB_2 have the same quality and are on average superior to lower bound LB_C, thus showing the effectiveness of our q-$*$route and ng-$*$route relaxations. Moreover, the different valid inequalities can substantially increases the lower bound, as shown by the improvements on instances north-68x7-Q24 and south-54x4-Q34.

The table shows that upper bound UB_2 is always better than upper bound UB_1 and that the BCP algorithm can further improve the upper bounds in almost all instances, thus producing high quality primal solutions also whenever the algorithm terminates without proving the optimality of the solution found.

It is worth mentioning that the time spent for computing upper bound UB_1 is on average equal to 187.4 seconds and that the time spent by the procedure used to compute upper bound UB_2 (called during the computation of lower bound LB_2) is on average equal to 226.8 seconds. Therefore, both UB_1 and UB_2 can be computed efficiently in practice.

7.2. Results on LRP based instances

This set of instances was derived from 75 LRP instances used in Baldacci et al. (2011) and Contardo et al. (2013) for solving the LRP and proposed by different authors. We derived two classes of test instances (A and B) having the same topology of the underlying graph, but with different cost structures.

We generated a total number of 150 instances, 75 instances per class. The dimensions of the instances vary from very small instances with 12 customers and two facilities up to large instances

Table 2 Summary results on Class A instances

	$\%UB_1$	$\%UB_2$	$\%LB_1$	$\%LB_2$	t_{DA}	$\%LB_C$	t_C	$\%LB$	t_{LB}	$\#Opt$	t_{TOT}
Akca et al. (2009)	100.3	100.6	94.3	96.8	9.9	96.1	4.6	98.5	2.1	10/12	145.3
Prins et al. (2004)	100.2	100.4	93.7	96.0	48.6	94.0	78.6	97.8	14.7	10/24	221.9
Different authors	100.2	101.0	91.9	94.3	297.5	93.7	339.8	96.6	121.3	10/39	213.0

Table 3 Summary results on Class B instances

	$\%UB_1$	$\%UB_2$	$\%LB_1$	$\%LB_2$	t_{DA}	$\%LB_C$	t_C	$\%LB$	t_{LB}	$\#Opt$	t_{TOT}
Akca et al. (2009)	102.6	101.3	94.6	96.9	5.2	95.6	4.1	98.1	3.7	9/12	274.2
Prins et al. (2004)	101.4	101.0	94.2	95.9	52.8	93.0	76.1	97.2	14.2	7/24	184.0
Different authors	101.1	102.5	92.4	94.2	187.6	93.4	324.1	96.1	141.2	8/39	284.3

with 150 customers and 20 facilities. The instance name is a string **name<a×b>**, where **name** represents the instance name, **a** represents the number of customers and **b** corresponds to the number of facilities.

For sake of presentation, the instances were grouped into the following three groups accordingly to the original LRP source:

i) Akca et al. (2009): 12 instances involving 5 facilities, and 30 or 40 customers;

ii) Prins et al. (2004): 24 instances involving 20, 50, and 100 customers, 5 or 10 facilities;

iii) Different authors: 39 instances, involving up to 150 customers and 20 facilities.

For this set of instances, a time limit of 3,600 seconds was imposed to the SCIP framework.

Tables 2 and 3 summarize the results obtained on both classes A and B. In the tables, column $\#Opt$ reports for each group of instances the total number of instances solved to optimality within the imposed time limit.

The meaning of the remaining columns is the same described in the previous section, but in the tables their values are relative to averages computed over the instances composing the three groups. The values reported under column t_{TOT} are computed over the instances solved to optimality within the imposed time limit.

Tables 2 and 3 show that 30 and 24 out of 75 instances were solved to optimality within the imposed time limit for classes A and B, respectively.

For these instances, lower bound LB_2 is on average superior with respect lower bound LB_1. As the feasible solutions associated with these instances are characterized (on average) by a larger number of customers per route, the ng-∗route relaxation performs in practice better than q-∗route relaxation. Also for these instances, the different valid inequalities can substantially increase the final lower bound (see column $\%LB$). Instances of Class B are more difficult with respect to the corresponding instances of class A. This is due to the different cost structure of class B instances and it is testified by the worse quality of lower bounds LB_C and of the final lower bound LB. Nonetheless, lower bounds LB_1 and LB_2 show the same quality of class A instances.

Table 4 Effectiveness of the dual ascent heuristics

					$t_{LB_{SP}}$				t_{LB_1}	
		$\%LB_1$	$\%LB_2$	$\%LB_{SP}$	(a)	(b)	(c)	(d)	(e)	(f)
A	Akca et al. (2009)	94.3	96.8	96.9	11.1	6.7	4.9	2.3	0.9	0.7
	Prins et al. (2004)	93.7	96.0	96.1	26.3	15.6	12.1	5.3	0.5	0.3
	Different authors	91.9	94.3	94.7	263.0	171.2	108.0	53.5	21.4	17.6
B	Akca et al. (2009)	94.6	96.9	96.9	13.0	7.9	5.8	2.7	0.9	0.7
	Prins et al. (2004)	94.2	95.9	96.0	26.4	17.4	13.2	5.8	0.5	0.4
	Different authors	92.4	94.2	94.8	248.3	168.6	116.5	56.9	21.7	18.9
Real-word		98.0	98.1	98.6	18.9	5.0	8.3	4.2	0.3	0.2
		93.6	95.5	95.8	129.9	85.2	57.4	28.0	10.3	8.7

(a) without lower bounds LB_1 and LB_2
(b) with lower bound LB_1
(c) with lower bound LB_2
(d) with lower bounds LB_1 and LB_2
(e) route set $\overline{\mathscr{R}}$ initialized with single-customer route
(f) route set $\overline{\mathscr{R}}$ initialized with the solution provided by the constructive heuristic

Concerning the upper bounds, the tables show that both the two upper bounding procedures can compute good quality solutions. The average computing time of upper bound UB_1 (UB_2) is equal to 70.8 and 72.9 seconds (148.0 and 89.5 seconds) for classes A and B, respectively. Therefore, the computation of LB_2 requires a higher computing time with respect to the real-world instances and this is due to the larger vehicle capacity that characterizes most of the instances in classes A and B.

The detailed results reported in the e-companion to this paper show that instances with up to 100 customers and 10 facilities were solved to optimality.

7.3. Effectiveness of the dual ascent heuristics and valid inequalities

Table 4 reports an analysis of the effectiveness of the dual ascent heuristics when used to initialize the master problem of problem $\overline{LSP}$ (see Section 5.2). In order to assess the quality of lower bounds LB_1 and LB_2, we solved problem $\overline{LSP}$ without adding valid inequalities, i.e., we computed the optimal solution cost LB_{SP} of formulation LSP and the LP-relaxation of formulation SP with ng-$*$route. In addition, the Lagrangean heuristic has been disabled during the computation of LB_1 and LB_2.

The table reports the average percentage deviations of lower bounds LB_1, LB_2, and LB_{SP} under columns $\%LB_1$, $\%LB_2$ and $\%LB_{SP}$, respectively. The table then reports, under heading $t_{LB_{SP}}$, the average total computing times spent in computing lower bound LB_{SP} under the following options: (a) without computing lower bounds LB_1 and LB_2 (b) by computing lower bound LB_1 (c) by computing lower bound LB_2, and (d) by computing both lower bounds LB_1 and LB_2. In case (a), the master problem of LSP is initialized with single-customer routes whereas in case (b), the master problem is initialized using the dual solution provided by lower bound LB_1, that is used to generate an initial set of ng-$*$route. In cases (c) and (d), the master problem is initialized with the route set generated by procedure CG during the computation of LB_2 (as described in 5.2). Moreover, in case (c) the master problem associated with the computation of LB_2, is initialized as for LB_1, i.e., using the solution provided by the constructive heuristic described in Section 6.1.

All values in the table are relative to averages computed over the instances composing the three groups of classes A and B, and over the real-world instances. The last row of the table reports averages computed over all instances.

The table shows that the bounding procedure based on the use of both lower bounds LB_1 and LB_2 (case (d)) is about five times faster than the standard column generation method (case (a)). Generally speaking, standard column generation methods are time-consuming as the LP-relaxation of the master problem is usually highly degenerate and degeneracy implies alternative optimal dual solutions. Consequently, the generation of new columns and their associated variables may not change the value of the objective function of the master problem, the master problem may become large, and the overall method may become slow computationally. In case (d), the computation of lower bound LB_{SP} starts from a near-optimal dual solution of the LP-relaxation of SP with ng-$*$route provided by lower bound LB_2, as shown by the percentage deviations of lower bounds LB_2 and LB_{SP}. This allows us to generate an initial master problem containing the routes having a very small reduced cost that are likely to be in the optimal LSP solution.

The analysis of cases (b) and (c) shows that it is also computationally convenient to compute LB_1 or LB_2. In particular, computing LB_1 before the computation of LB_2 speedup the computation of LB_2 as procedure CG used to compute LB_2 takes advantage from the master initialization provided by the dual solution corresponding to LB_1.

Table 4 also reports the computational results obtained when calculating the lower bound LB_1 under the following two ways of initializing the corresponding master problem: (i) by using the heuristic solution provided by the constructive heuristic (case (e)) (ii) by using single-customer routes (case (f)). The table shows that on average, the difference is slightly marginal. Nevertheless, as in our implementation the constructive heuristic is executed before computing LB_1, it is worthwhile to initialize the master of LB_1 with the solution found by the heuristic.

Table 5 analyses the impact of the valid inequalities on the column-and-cut bounding procedure described in Section 5.2 at the root node of the BCP method.

The table reports average percentage deviations of the lower bounds obtained by the bounding procedure under the following cases: (i) without adding valid inequalities (under column heading "no cuts") (ii) by adding CI, MI and RCI inequalities ("+ CI + MI + RCI") (iii) by adding CI, MI, RCI, RCII-a, and RCII-b inequalities ("+ RCII-a + RCII-b"), and (iv) by adding CI, MI, RCI, RCII-a, RCII-b, RCII-c, RCII-d inequalities ("+ RCII-c + RCII-d"). The last case corresponds to the final procedure we adopted in our computational results and, as mentioned in Section 5.2,

Table 5 Effectiveness of the different type of inequalities on column-and-cut generation procedure

		no cuts		+ CI + MI + RCI			+ RCII-a + RCII-b			+ RCII-c + RCII-d		
		$\%LB$	t_{LB}	$\%LB$	t_{LB}	$\#cuts$	$\%LB$	t_{LB}	$\#cuts$	$\%LB$	t_{LB}	$\#cuts$
A	Akca et al. (2009)	96.9	2.3	97.9	3.1	7.3	98.5	4.0	200.6	98.5	4.0	169.9
	Prins et al. (2004)	96.1	5.3	97.1	10.0	8.3	97.8	17.2	521.5	97.8	18.2	539.8
	Different authors	94.7	53.5	95.7	92.1	23.8	96.5	139.3	1083.2	96.6	164.4	1181.3
B	Akca et al. (2009)	96.9	2.7	97.4	3.6	13.3	98.0	5.5	655.6	98.1	5.6	763.0
	Prins et al. (2004)	96.0	5.8	96.3	10.7	14.6	97.1	16.2	623.8	97.2	17.7	737.0
	Different authors	94.8	56.9	95.6	109.0	61.4	96.0	173.3	1385.3	96.1	184.3	1566.1
Real-word		98.6	4.2	98.7	6.6	4.1	98.7	6.9	106.0	98.7	12.9	121.1
		95.8	28.0	96.5	50.8	24.9	97.1	78.8	809.2	97.2	88.1	905.2

the sequence of separation procedures was defined after conducting preliminary computational experiments performed to identify a good separation strategy.

For each group of inequalities, the table reports the average percentage deviations of the lower bounds obtained and the corresponding average computing times ($\%LB$, t_{LB}), and the average cardinalities of the sets $\overline{\mathscr{F}}$ associated with the lower bound computation ($\#cuts$). As for Table 4, the Lagrangean heuristic has been disabled during the computation of LB_1 and LB_2. In addition, the time t_{LB} also includes the time spent for computing LB_1 and LB_2.

As for Table 4, all values in the table are relative to averages computed over the instances composing the three groups of classes A and B, and over the real-world instances. The last row of the table reports averages computed over all instances.

The table shows that the average percentage gaps left by considering in turn the different three groups of valid inequalities are equal to 3.5, 2.9 and 2.8, respectively. With respect to the "no cuts" case, a final gap reduction of about 1.4% has been achieved. The contribution given by inequalities RCII-c and RCII-d is on average equal to 0.1% as shown by the table. During preliminary computational experiments, we observed that their addition generally results in separating additional RCI and RCII-b inequalities, which separation procedures are heuristics.

8. Conclusions

In this paper, we considered a vehicle routing problem with transhipment facilities, called the Vehicle Routing Problem with Transhipment Facilities (VRPTF), that was motivated by a real-world application of interest to an Italian company operating in the production and distribution of non-perishable products. The VRPTF consists of selecting transhipment facilities, allocating customers to these facilities and designing vehicle routes emanating from a central depot to minimize the total distribution cost. A feature of the problem is that a customer can be either served on a vehicle route emanating from the central depot or through an intermediate facility, where the demand is first delivered by a vehicle route, and then it is successively delivered to the final customer.

We proposed two integer programming formulations for the VRPTF, a two-index formulation (TI) and a set-partitioning based formulation (SP). The formulations were used to derive a bounding method based on two dual ascent heuristics and a column-and-cut generation procedure. In particular, we proposed valid inequalities to strengthen the linear relaxations of the two formulations and two different route relaxations, called q-∗route and ng-∗path, that have the advantage that the pricing subproblem associated with the linear relaxation of formulation SP can be efficiently solved (by dynamic programming).

All our findings have been used to develop branch-and-cut-and-price algorithm that has been tested on a large family of instances, including both real-world instances and instances derived from the literature.

The implementation solved to optimality different instances from our real-world instances involving up to 142 customers and 18 facilities. The implementation was also tested on literature-based instances to better evaluate the limits of the algorithms, and the new approaches can find optimal solutions on some difficult instances with up to 100 customers and 10 facilities.

Acknowledgements
The authors thank two anonymous referees for making several suggestions that improved the presentation of the paper.

References
Achterberg, T. 2009. Scip: Solving constraint integer programs. *Mathematical Programming Computation* **1**(1) 1–41.

Achterberg, T., T. Berthold. 2009. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 6th International Conference, CPAIOR 2009 Pittsburgh, PA, USA, May 27-31, 2009 Proceedings*, chap. Hybrid Branching. Springer Berlin Heidelberg, Berlin, Heidelberg, 309–311.

Akca, Z., R. T. Berger, T. K. Ralphs. 2009. A branch-and-price algorithm for combined location and routing problems under capacity restrictions. J. W. Chinneck, B. Kristjansson, M. J. Saltzman, eds., *Operations Research and Cyber-Infrastructure*, vol. 47. Springer US, 309–330.

Applegate, D. L., R. E. Bixby, V. Chvatal, W. J. Cook. 2007. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA.

Baldacci, R., M. Dell'Amico, J. Salazar González. 2007. The Capacitated m-Ring-Star Problem. *Operations Research* **55**(6) 1147–1162.

Baldacci, R., A. Mingozzi, R. Roberti, R. Wolfler Calvo. 2013. An Exact Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem. *Operations Research* **61**(2) 298–314.

Baldacci, R., A. Mingozzi, R. Wolfler Calvo. 2011. An Exact Method for the Capacitated Location-Routing Problem. *Operations Research* **59**(5) 1284–1296.

Belle, J. V., P. Valckenaers, D. Cattrysse. 2012. Cross-docking: State of the art. *Omega* **40** 827846.

Benichou, M., J. M. Gauthier, P. Girodet, G. Hentges, G. Ribiere, O. Vincent. 1971. Experiments in mixed-integer linear programming. *Mathematical Programming* **1**(1) 76–94.

Christofides, N., A. Mingozzi, P. Toth. 1981. Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. *Mathematical Programming* **10** 255–280.

Contardo, C., J.-F. Cordeau, B. Gendron. 2013. An Exact Algorithm Based on Cut-and-Column Generation for the Capacitated Location-Routing Problem. *INFORMS Journal on Computing* **26**(1) 88–102.

Drexl, M. 2012. Synchronization in vehicle routing - a survey of vrps with multiple synchronization constraints. *Transportation Science* **46**(3) 297–316.

Há, M. H., N. Bostel, A. Langevin, L.M. Rousseau. 2013. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research* **226**(2) 211–220.

Hachicha, M., M John Hodgson, G. Laporte, F. Semet. 2000. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research* **27**(1) 29–42.

IBM CPLEX. 2014. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html.

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

Poggi, M., E. Uchoa. 2014. Chapter 3: New exact algorithms for the capacitated vehicle routing problem. *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics, 59–86.

Prins, C., C. Prodhon, R. Wolfler Calvo. 2004. Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. A. Dolgui, S. Dauzère-Pérès, eds., *MOSIM'04*, vol. 2. Lavoisier, Ecole des Mines de Nantes, 1115–1122.

Prodhon, C., C. Prins. 2014. A survey of recent research on location-routing problems. *European Journal of Operational Research* **238**(1) 1–17.

Riera-Ledesma, J., J. Salazar González. 2013. A column generation approach for a school bus routing problem with resource constraints. *Computers & Operations Research* **40**(2) 566–583.

Riera-Ledesma, J., J.-J. Salazar-González. 2012. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research* **39**(2) 391–404.

Toth, P., D. Vigo. 2014. *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization, SIAM, Philadelphia.