Transportation Science

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

A Branch-and-Cut Algorithm for the Single Truck and Trailer Routing Problem with Satellite Depots

José Manuel Belenguer, Enrique Benavent, Antonio Martínez, Christian Prins, Caroline Prodhon, Juan G. Villegas

TRANSPORTATION SCIENCE
Vol. 50, No. 2, May 2016, pp. 735–749
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2014.0571

A Branch-and-Cut Algorithm for the Single Truck and Trailer Routing Problem with Satellite Depots

José Manuel Belenguer, Enrique Benavent
Departament d'Estadística i Investigació Operativa, Universitat de València, 46100 Burjassot (València), Spain
{jose.belenguer@uv.es, enrique.benavent@uv.es}

Antonio Martínez
Southampton Management School, Faculty of Business and Law, University of Southampton, Highfield,
Southampton SO17 1BJ, United Kingdom, a.martinez-sykora@soton.ac.uk

Christian Prins, Caroline Prodhon
Institut Charles Delaunay, Laboratoire d'Optimisation des Systèmes Industriels (LOSI),
Université de Technologie de Troyes, 10004 Troyes Cedex, France
{christian.prins@utt.fr, caroline.prodhon@utt.fr}

Juan G. Villegas
Departamento de Ingeniería Industrial, Facultdad de Ingeniería, Universidad de Antioquia, 050010 Medellín, Colombia,
juan.villegas@udea.edu.co

In the single truck and trailer routing problem with satellite depots (STTRPSD), a truck with a detachable trailer based at a main depot must serve the demand of a set of customers accessible only by truck. Therefore, before serving the customers, it is necessary to detach the trailer in an appropriate parking place (called either a satellite depot or a trailer point) and transfer goods between the truck and the trailer. This problem has applications in milk collection for farms that cannot be reached using large vehicles. In this work we present an integer programming formulation of the STTRPSD. This formulation is tightened with several families of valid inequalities for which we have developed different (exact and heuristic) separation procedures. Using these elements, we have implemented a branch-and-cut algorithm for the solution of the STTRPSD. A computational experiment with published instances shows that the proposed branch-and-cut algorithm consistently solves problems with up to 50 customers and 10 satellite depots, and it has also been able to solve instances with up to 20 satellite depots and 100 clustered customers.

Keywords: branch-and-cut; cutting planes; truck and trailer routing problem; vehicle routing problem
History: Received: January 2014; revision received: July 2014; accepted: August 2014. Published online in *Articles in Advance* March 31, 2015.

1. Introduction

The single truck and trailer routing problem with satellite depots (STTRPSD) is an extension of the well-known capacitated vehicle routing problem (CVRP) (Laporte 2007, 2009), in which a single vehicle composed of a truck with a detachable trailer serves the demand of a set of customers reachable only by the truck without the trailer. This accessibility constraint implies that the trailer must be parked before the customers are visited. Therefore, there is a set of parking locations (called either trailer points or satellite depots) in which it is possible to detach the trailer and to transfer products between the truck and the trailer.

A feasible solution of the STTRPSD (depicted in Figure 1) is composed of (i) a first-level trip departing from the main depot (performed by the truck with the trailer) visiting a subset of trailer points and (ii) several second-level trips performed by the truck from any of the trailer points visited in the first-level trip. The goal of the STTRPSD is to minimize the total length of the trips (at both levels).

There are only two publications tackling exactly the same problem. Villegas et al. (2009) introduce the problem and develop a hybrid of a greedy randomized adaptive search procedure (GRASP) and evolutionary local search. Villegas et al. (2010) propose a multistart evolutionary local search and a hybrid metaheuristic based on GRASP and variable neighborhood descent (VND). In their computational experiments on a set of randomly generated instances, multistart evolutionary local search outperformed GRASP/VND in terms of solution quality and running time.

The STTRPSD is a particular case of the truck and trailer routing problem (TTRP), introduced first by Chao (2002). In the TTRP there are two types of customers: *truck customers* that must be served by a

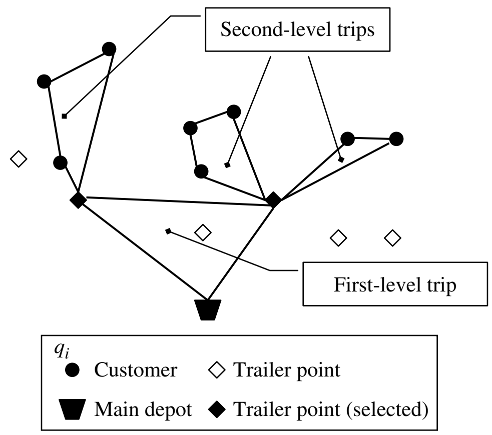

Figure 1 **Example of an STTRPSD Solution**

truck and *vehicle customers* that can be served by either a truck or a truck pulling a trailer. The truck may start and end its route at a central depot, with or without the trailer coupled to the truck. In this problem, trailers can be parked at vehicle-customer locations, and there the goods collected can be transferred from the truck to the trailer (or vice versa, in the case of delivery operations). Note that in the STTRPSD, the vehicle-customer set is empty, and the parking places for the trailers are not associated with customer locations.

Practical applications of the TTRP appear mainly in collection and delivery operations in rural areas or crowded cities with accessibility constraints. Milk collection is maybe the best known practical application of the TTRP. Hoff and Løkketangen (2007) and Hoff (2012) develop a tabu search algorithm for the solution of a routing problem for milk collection in Norway. Caramia and Guerriero (2010b) study the milk collection of an Italian dairy company by a TTRP variant with multicompartments, route-length constraints, and heterogeneous trucks and trailers. Pureza, Morabito, and Reimann (2012) introduce a variant of the vehicle routing problem with time windows arising in the delivery of goods in highly populated Brazilian cities. Because of the heavy traffic and scarcity of parking spaces, after parking the vehicle, deliverymen assigned to that vehicle serve customers on foot with the possibility of making multiple tours at each parking site.

Several metaheuristics for the TTRP are based on the cluster-first route-second principle. Chao (2002) and Scheuerer (2006) describe tabu search methods that follow this approach. Similarly, Caramia and Guerriero (2010a) propose a mathematical-programming-based heuristic that solves two subproblems sequentially: a customer-route assignment problem that assigns customers to valid clusters and a route-definition problem for each cluster. Lin, Yu, and Chou (2009) develop a very effective simulated annealing (SA) algorithm for the TTRP that uses an indirect representation of the solutions. Again using their method, they solve the variants without truck-and-trailer availability constraints (Lin, Yu, and Chou 2010) and with time windows (TTRPTW; Lin, Yu, and Lu 2011). In contrast, Villegas et al. (2011) embed a route-first cluster-second construction heuristic in a GRASP with evolutionary path relinking that outperforms most previously published methods based on the cluster-first route-second principle.

More recently, Derigs, Pullmann, and Vogel (2013) have proposed a hybrid metaheuristic for the TTRP, the TTRPTW, and a variant without the option of load transfer between trucks and trailers. Their method combines large neighborhood search with standard metaheuristic control strategies based on record-to-record travel and the attribute-based hill climber. Finally, the best results to date for the TTRP were obtained by Villegas et al. (2013) by using a two-phase matheuristic that uses the routes of the local optima of a hybrid GRASP introduced by Villegas et al. (2011) as columns in a set partitioning formulation of the TTRP.

Drexl (2007, 2011) presents the generalized truck and trailer routing problem (GTTRP); it is a generalization of the TTRP that includes, in addition to the vehicle-customers' locations, other transshipment locations. Furthermore, he considers variable costs for trailers, fixed costs for trucks, and time windows for customers. He describes a branch-and-price algorithm and a heuristic variant of this algorithm. Drexl (2014) introduces the vehicle routing problem with trailers and transshipment (VRPTT) where the fixed assignment of trailers to trucks is dropped. In the VRPTT goods can be transferred from any truck to any trailer at the transshipment locations. Two formulations for the VRPTT based on the one-commodity flow model are described, and several branch-and-cut algorithms are developed based on these formulations. Computational experiments on the set of test instances introduced by Drexl (2007) show that some VRPTTs with up to eight customers, eight transshipment locations, and eight vehicles can be solved to optimality within a time limit of three hours.

The STTRPSD is also a particular case of the two-echelon capacitated vehicle routing problem (2E-CVRP) where there are several first-level trips, and the satellites have a limited capacity and can be serviced by more than one first-level trip. Some articles on the 2E-CVRP also consider the handling costs of loading/unloading items at the satellites and the fixed costs of using the vehicles and/or the satellites, and a given fleet of vehicles for the first- and/or second-level trips. Gonzalez-Feliu et al. (2007) describe a commodity flow formulation and an exact branch-and-cut algorithm. It is improved by Perboli, Tadei, and Vigo (2011), who reported optimal solutions for instances with up to 32 customers and two satellites. Jepsen, Spoorendonk, and Røpke (2013) describe an exact branch-and-cut algorithm based on a new formulation and valid inequalities; it outperforms the method of Perboli, Tadei, and Vigo (2011). Contardo, Hemmelmayr, and Crainic (2012) consider a 2E-capacitated location routing problem (2E-CLRP) with several main depots. They propose a two-index flow formulation and a branch-and-cut algorithm that is able to solve instances with up to 50 customers and 10 satellites. Santos, da Cunha, and Mateus (2013) formulate the 2E-CVRP as a set covering problem and propose a branch-and-price algorithm that can solve instances with up to 50 customers and four satellites. Finally, Baldacci et al. (2013) introduce a set partitioning formulation and an exact algorithm that decomposes the 2E-CVRP into a set of multidepot CVRPs with side constraints. Their method outperforms all previous exact algorithms and is able to solve to optimality 144 of 153 instances tested, with up to 100 customers and six satellites.

Perboli, Tadei, and Vigo (2011) introduce two matheuristics. Their methods use information from the linear programming (LP) relaxation of the 2E-CVRP to fix some variables. After fixing the variables, both methods solve one VRP for each satellite depot and one for the first level connecting the satellites with the main depot. The methods differ in the rules used to fix the variables. Crainic et al. (2011) use a similar approach. They assign customers to satellites by means of a clustering heuristic and then solve the corresponding VRPs for the satellites and the main depot using a branch-and-cut method (Ralphs 2003). Additionally, a clustering improvement mechanism perturbs and improves the customer-to-satellite assignment to find multiple solutions in a multistart strategy. Crainic et al. (2013) enhance this approach by embedding the assignment into a GRASP where the resulting VRPs are solved heuristically and a path-relinking operator is used for post-optimization. Hemmelmayr, Cordeau, and Crainic (2012) propose an adaptive large neighborhood search (ALNS) for the 2E-CVRP. In their method, several destroy and repair operators are used in a hierarchical fashion: although some operators modify the set of satellites visited in the first level, others modify a small part of the solution, changing only the structure of some routes of the second level. To date, the best metaheuristic for the 2E-CVRP is the ALNS by Hemmelmayr, Cordeau, and Crainic (2012). Finally, for a comprehensive review of two-echelon vehicle routing problems including 2E-CLRP, 2E-CVRP, TTRP, and STTRPSD, the reader is referred to Cuda, Guastaroba, and Speranza (2015).

In this paper we introduce a two-index flow formulation of the STTRPSD that is tightened with several families of valid inequalities. We have developed and implemented several heuristic and exact separation procedures for those families that include an exponential number of inequalities. These procedures have been embedded in a branch-and-cut algorithm that was tested using a set of STTRPSD instances from the literature with between 25 and 200 customers and between five and 20 satellites. The branch-and-cut uses the upper bounds obtained by Villegas et al. (2009, 2010) and is able to optimally solve several instances with up to 100 customers. As far as we know, this is the first exact procedure specifically designed for the STTRPSD.

The remainder of this paper is organized as follows. Section 2 presents an integer programming formulation of the STTRPSD. Section 3 introduces several valid inequalities that strengthen the initial formulation. Section 4 describes the elements of the proposed branch-and-cut algorithm. Section 5 presents the results of a computational evaluation of the proposed method on a set of instances from the literature. Finally, §6 presents some conclusions and outlines future work.

2. Integer Programming Model

The STTRPSD can be modeled using an undirected graph $G=(V,E)$, where $V=\{0\}\cup V_d\cup V_c$ is the node set with the main depot at node 0, $V_d=\{1,\ldots,p\}$ is the set of satellite depots (trailer points), and $V_c=\{p+1,\ldots,p+n\}$ is the set of customers with known demands $q_i>0$ ($\forall\, i\in V_c$). The edge set E is defined as $E=\{\{0,j\}\colon j\in V_d\}\cup\{\{i,j\}\colon i,j\in V_d\cup V_c,\, i<j\}$. Each edge $\{i,j\}\in E$ has a cost c_{ij} that satisfies the triangle inequality. The parameters Q_v and Q_t are the capacities of the truck and trailer, respectively. The total demand of the customers does not exceed Q_v+Q_t to ensure feasibility with one vehicle.

Let $V_1=\{0\}\cup V_d$ be the set of nodes that can be visited by the truck with the trailer in the first-level trip, and $V_2=V_d\cup V_c$ be the set of nodes that can be visited by the truck in second-level trips. As usual, ($S\subseteq V$) $\delta(S)$ denotes the subset of edges with one node in S and the other in $V\backslash S$; $\gamma(S)$ is the subset of edges with both nodes in S; and $E(S\colon S')$, $S'\subseteq V\backslash S$ denotes the subset of edges with one node in S and the other in S'. Finally, when $S\subseteq V_c$, the quantity $k(S)$ denotes a lower bound on the number of second-level trips needed to serve the customers in S, and is computed as $k(S)=\lceil\sum_{i\in S}q_i/Q_v\rceil$. A better lower bound on the number of second-level trips, denoted by $r(S)$, can be computed by solving the bin packing problem (BPP) with bin capacity Q_v and the set of item weights $\{q_i\colon i\in S\}$. Additionally, $D(S)$ denotes the total demand of the customers in S, $D(S)=\sum_{i\in S}q_i$.

In the mathematical programming formulation, the integer variables y_{ij} represent the number of times that the truck with the trailer traverses edge $\{i, j\} \in \gamma(V_1)$ in the first-level trip. For edges $\{i, j\} \in \gamma(V_d)$, y_{ij} takes binary values, and for edges $\{0, j\} \in E(\{0\}: V_d)$, the variables y_{0j} may also take the value of 2, representing a direct trip to satellite depot $j \in V_d$. For a given subset of edges $H \subseteq \gamma(V_1)$, $y(H) = \sum_{\{i,j\}\in H} y_{ij}$.

Similarly, the integer variables x_{ij} represent the number of times edge $\{i, j\} \in \gamma(V_c) \cup E(V_d: V_c)$ is traversed by the truck in second-level trips. For edges $\{i, j\} \in \gamma(V_c)$, x_{ij} takes binary values, and for edges $\{i, j\} \in E(V_d : V_c)$, variable x_{ij} may also take the value of 2 when there is a direct trip from a satellite depot $i \in V_d$ to a customer $j \in V_c$. For a given subset of edges $H \subseteq \gamma(V_2)$, $x(H) = \sum_{\{i,j\}\in H} x_{ij}$.

Finally, binary variable z_i ($i \in V_d$) indicates whether ($z_i = 1$) or not ($z_i = 0$) satellite depot i is selected to park the trailer. Using the notation given above, the STTRPSD is formulated as follows:

$$\min \left\{ \sum_{\{i,j\}\in\gamma(V_1)} c_{ij} y_{ij} + \sum_{\{i,j\}\in\gamma(V_2)} c_{ij} x_{ij} \right\} \tag{1}$$

$$\text{subject to } \quad x(\delta(j)) = 2, \quad \forall j \in V_c, \tag{2}$$

$$x(\delta(S)) \geq 2k(S), \quad \forall S \subseteq V_c, \tag{3}$$

$$\sum_{i\in I'} x_{ij} + 2x(\gamma(S \cup \{j, l\})) + \sum_{k\in V_d \backslash I'} x_{kl} \leq 2|S| + 3, \quad \forall j, l \in V_c, \quad \forall S \subseteq V_c \backslash \{j, l\},\ S \neq \varnothing,\ \forall I' \subset V_d, \tag{4}$$

$$\sum_{i\in I'} x_{ij} + 3x_{jl} + \sum_{k\in V_d\backslash I'} x_{kl} \leq 4, \quad \forall j, l \in V_c,\ \forall I' \subset V_d, \tag{5}$$

$$\sum_{i\in V_2\backslash S,\, j\in S_c} x_{ij} + k(S_c) y(\delta(S_d)) \geq 2k(S_c), \quad \forall S_c \subseteq V_c,\ \forall S_d \subseteq V_d,\ S = S_c \cup S_d, \tag{6}$$

$$y(\delta(i)) = 2z_i, \quad \forall i \in V_d, \tag{7}$$

$$x_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in \gamma(V_c), \tag{8}$$

$$x_{ij} \in \{0, 1, 2\}, \quad \forall \{i, j\} \in E(V_d: V_c), \tag{9}$$

$$y_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in \gamma(V_d), \tag{10}$$

$$y_{ij} \in \{0, 1, 2\}, \quad \forall \{i, j\} \in \delta(\{0\}), \tag{11}$$

$$z_i \in \{0, 1\}, \quad \forall i \in V_d. \tag{12}$$

The objective function (1) has two terms: the former is the total cost of the first-level trip and the latter is the total cost of the second-level trips. Constraints (2) state that all customers must be visited once and also enforce the continuity of second-level trips. Constraints (3) impose the capacity restrictions of second-level trips and ensure that they contain a satellite depot. Constraints (4) and (5), adapted from Belenguer et al. (2011), are called path-elimination constraints since they forbid inter-depot second-level trips (i.e., trips that begin at one satellite depot and end at another). Constraints (6), called connection constraints, state that each subset of customers $S_c \subseteq V_c$ must be connected to the main depot. These constraints as well as the path-elimination constraints are explained below. Constraints (7) state that the first-level trip must visit exactly once the satellite depots that are used. Constraints (8) to (12) define the variables.

The path-elimination constraints (4) can be shown to be valid as follows. Note that if neither j nor l are served with direct trips, $\sum_{i\in I'} x_{ij} + \sum_{k\in V_d\backslash I'} x_{kl} \leq 2$ and $x(\gamma(S \cup \{j, l\})) \leq |S| + 1$ because of the subtour-elimination constraints. This last inequality is satisfied with equality only if all of the customers in $S \cup \{j, l\}$ are visited consecutively in a trip; however, if this holds, then we have $\sum_{i\in I'} x_{ij} + \sum_{k\in V_d\backslash I'} x_{kl} \leq 1$ because otherwise the trip would be connected to two satellite depots, one in I' and the other in $V_d \backslash I'$, so (4) is satisfied. If one of the customers j or l, or both, are served with direct trips, then $\sum_{i\in I'} x_{ij} + \sum_{k\in V_d\backslash I'} x_{kl}$ is at most 3 or 4, respectively, but then $x(\gamma(S \cup \{j, l\}))$ is at most $|S|$ or $|S| - 1$, respectively, so the inequality is trivially satisfied. The path-elimination constraints (5) corresponding to the case where $S = \varnothing$ avoid paths connecting two different satellite depots and visiting only two customers, and they can be explained in a similar way. Note finally that the formulation allows paths connecting two different satellite depots and containing only one customer, such as $x_{ij} = 1$ and $x_{jk} = 1$, where $j \in V_c$ and $i, k \in V_d$. However, this solution will never appear since it is dominated by $x_{hj} = 2$, where h is i or k depending on which is the nearer satellite depot to customer j.

To see that the connection constraints (6) are valid, note that they are satisfied if $y(\delta(S_d)) \geq 2$. Moreover, if $y(\delta(S_d)) = 0$, this means that no satellite depot of S_d is used; however, then $\sum_{i\in V_2\backslash S,\, j\in S_c} x_{ij} = x(\delta(S_c)) \geq 2k(S_c)$ because of the capacity constraints, so the inequality is satisfied. Figure 2 illustrates a violated connection constraint.

Let us prove now that any nondominated integer solution of the formulation (1) to (12) defines an STTRPSD solution. Let $(\bar{x}, \bar{y}, \bar{z})$ be a nondominated solution of the above formulation, and let $G_{\bar{x}} = (V_2, E_{\bar{x}})$ be an undirected graph where $E_{\bar{x}}$ contains $\bar{x}_e$ copies of edge $e \in \gamma(V_2)$. It is obvious that every node $j \in V_c$ has degree two in $G_{\bar{x}}$. Furthermore, because of constraints (4) and (5), there is no path connecting two different satellite depots, which also implies that the degree of every node $i \in V_d$ is even. Therefore, $G_{\bar{x}}$ can be decomposed into a set of cycles each containing a satellite depot, and the total demand of each cycle does not exceed the capacity Q_v because of the capacity constraints (3). Let us now denote by $G_{\bar{y}} = (V_1, E_{\bar{y}})$ an undirected graph where $E_{\bar{y}}$ contains

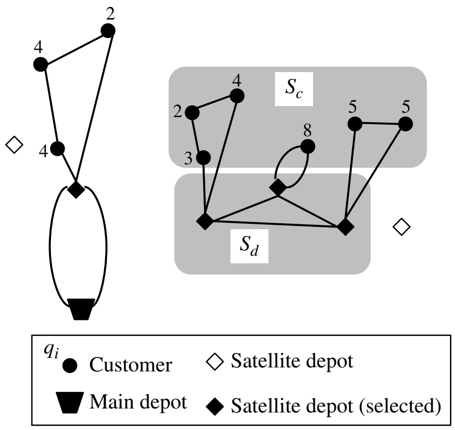

Figure 2 Example of the Violation of a Connection Constraint ($Q_v = 10$)

$\bar{y}_e$ copies of edge $e \in \gamma(V_1)$. Because of constraints (7), every node of $G_{\bar{y}}$ has degree zero or two, so $G_{\bar{y}}$ can be decomposed into a set of cycles. Furthermore, because of the connection constraints (6), these cycles contain all of the satellite depots $i \in V_d$ that are used by any second-level trip, and they are all connected to the main depot. If the triangular inequality holds, we may assume that $G_{\bar{y}}$ consists of only one cycle, so the degree of the main depot is always two.

Finally, let us remark that a tighter formulation can be obtained by substituting $k(S)$ in (3) and (6) by the better bound $r(S)$. However, computing $r(S)$ involves solving the BPP, which is an NP-hard problem; for this reason, we will use $k(S)$ unless otherwise stated.

3. Valid Inequalities

In this section we introduce several families of constraints that can be used to strengthen the LP relaxation of the STTRPSD formulation (1) to (12). Some of these families have been adapted to the STTRPSD from other vehicle routing problems, such as the multidepot multiple traveling salesman problem (MDMTSP; Benavent and Martínez 2013) and the location routing problem (LRP; Belenguer et al. 2011), whereas others have been specifically developed for this problem.

Some of these constraints are in fact *optimality cuts*, which means that they are not valid for the STTRPSD but they do not eliminate all of the optimal solutions. If the costs satisfy the triangular inequality, it is easy to see that there are always optimal solutions for which the following property holds: there are no two second-level trips based at the same satellite depot and visiting customers whose total demand is less than or equal to Q_v. The reason is that if two such trips appear in any solution they can be merged into a single trip without increasing the cost. This property was introduced for the LRP by Belenguer et al. (2011), and it is called the *TI-property*.

3.1. Co-Circuit Constraints

The co-circuit constraints are based on the following property of any STTRPSD solution: the number of times that the edges of an edge-cut are used, by the first- and/or second-level trips, must be even. The co-circuit constraints for the STTRPSD have the following structure:

$$x(\delta(S_c)\backslash F_c) \geq x(F_c) - |F_c| + 1,$$
$$S_c \subset V_c,\ F_c \subseteq E(S_c\colon V_c\backslash S_c)\colon |F_c| \quad \text{is odd}, \tag{13}$$

$$y(\delta(S_d)\backslash F_d) \geq y(F_d) - |F_d| + 1,$$
$$S_d \subset V_d,\ F_d \subseteq E(S_d\colon V_d\backslash S_d)\colon |F_d| \quad \text{is odd}. \tag{14}$$

We now discuss the validity of the co-circuit constraints with x variables (13). Given a feasible STTRPSD solution $(\bar{x}, \bar{y}, \bar{z})$, note that all of the variables associated with F_c are binary. If $\bar{x}(F_c) < |F_c|$, constraints (13) are trivially satisfied because the right-hand side (RHS) is negative or zero. When $\bar{x}(F_c) = |F_c|$, since $|F_c|$ is odd, at least one edge from $\delta(S_c)\backslash F_c$ must be used, and therefore the constraint is also satisfied. Note that F_c cannot contain any edge incident with satellite depots because the associated x-variables are not binary, so the inequality would not be valid in that case. The same arguments can be used for the co-circuit constraints with y variables (14).

3.2. Satellite Depot Cuts

This family of constraints is valid for STTRPSD solutions satisfying the TI-property. If two customers i and j have a joint demand that does not exceed Q_v, then we cannot serve both customers in two direct trips departing from the same satellite depot. Generalizing this observation to a subset of customers S such that $D(S) \leq Q_v$ produces the following constraints:

$$x(E(\{i\}\colon S)) \leq (|S|+1)z_i,\quad i \in V_d,\ S \subseteq V_c\colon D(S) \leq Q_v. \tag{15}$$

If $z_i = 0$, constraints (15) are trivially satisfied because in that case $x_{ij} = 0\ \forall\, j \in V_c$. To analyze the validity of constraints (15) when $z_i = 1$, we let $S' \subseteq S$ be the subset of customers served by direct trips from satellite depot i. On one hand, if $|S'|$ is even all of the customers can be matched in pairs, creating a lower-cost solution, and then $|S'| = 0$. On the other hand, if $|S'|$ is odd $|S'| - 1$ customers can be matched to produce a lower-cost solution and only one customer is served by a direct trip. Therefore, the cardinality of $|S'|$ is at most 1. If there are no direct trips then $x(E(\{i\}\colon S)) \leq |S|$, and constraint (15) is trivially satisfied. If there is a direct trip serving one customer, say, j, then $x(E(\{i\}\colon S\backslash\{j\}) \leq |S| - 1$ and $x_{ij} = 2$, and constraint (15) is satisfied.

3.3. Depot-Degree Constraints

These constraints are also valid only for STTRPSD solutions satisfying the TI-property

$$x(\gamma(S)) + x(E(\{i\}\colon S)) - 2z_i \le |S| - 1,\quad i \in V_d,\ S \subseteq V_c\colon D(S) \le Q_v. \tag{16}$$

Let $(\bar{x}, \bar{y}, \bar{z})$ be a feasible solution of the problem. If satellite depot i is not used, $\bar{z}_i = 0 = \bar{x}(E(\{i\}\colon S))$ and constraint (16) is trivially satisfied.

Let us now assume that $\bar{z}_i = 1$. Since the TI-property holds and $D(S) \le Q_v$, then there is at most one second-level trip rooted at satellite depot i serving only customers in S. Let $k = \bar{x}(E(\{i\}\colon S))$. There are two possibilities:

(i) There is one trip rooted at satellite depot i serving only customers in S. If this trip serves all of the customers in S, then $\bar{x}(\gamma(S)) + \bar{x}(E(\{i\}\colon S)) = \bar{x}(\gamma(S \cup \{i\})) = |S| + 1$ and we are done. Otherwise, there are $k-2$ paths starting at i and visiting some customers in S and some customers in $V_c \backslash S$, so $\bar{x}(\gamma(S)) = |S| - k + 1$; since $k = \bar{x}(E(\{i\}\colon S))$, the inequality is satisfied.

(ii) If there is no trip rooted at satellite depot i serving only customers in S, then $\bar{x}(E(S\colon V_c \backslash S)) \ge k$. Hence, $\bar{x}(\gamma(S)) = |S| - k$ and the inequality is also satisfied.

3.4. CVRP Constraints

There are several polyhedral studies of the CVRP that present families of valid inequalities that can be used to strengthen the linear relaxation of different CVRP formulations. Moreover, some of the most successful methods for the solution of the CVRP are based on cutting plane approaches (Augerat et al. 1995; Lysgaard, Letchford, and Eglese 2004). Note that if all of the satellite depots are shrunk into a single depot, the inequalities of the CVRP can be used to strengthen the LP relaxation of the STTRPSD. Specifically, in the branch-and-cut, we employ the CVRP combs described in Lysgaard, Letchford, and Eglese (2004).

3.5. Combs with Satellite Depots

Inequalities derived for the STTRPSD from CVRP combs have the property that all of the satellite depots are in the same part of the comb structure because all of them have been shrunk into a single depot. This is a great drawback because satellite depots can be far from each other, thus making the resulting inequality ineffective. Contardo, Cordeau, and Gendron (2013) introduced new comb inequalities in the context of the LRP in which depot locations can be in different teeth. However, their approach mainly relies on the capacity of the depots, which is unlimited in the STTRPSD. Benavent and Martínez (2013) introduced the so-called multidepot comb inequalities for the MDMTSP, where the depots can be in different parts of the comb. They introduced two types of multidepot combs: H-combs where some depots are in the handle of the comb, and T-combs for which every tooth of the comb contains at least one depot. These inequalities are valid for the STTRPSD, where the role of the depots is played by satellite depots. However, these inequalities are too weak because they do not take into account customer demands and truck capacity. In what follows we introduce multidepot combs that consider the truck capacity. Two types of capacitated multidepot combs are defined, depending on which part of the comb contains satellite depots. For the sake of clarity, the proofs of the validity of the inequalities of this subsection can be found in the appendix.

3.5.1. Capacitated H-Comb Inequalities. A capacitated H-comb is defined by a vertex set $H \subset V_c \cup V_d$, $H \cap V_d \ne \varnothing$, called the handle, and $t \ge 1$ vertex sets $T_i \subseteq V_c$, $i = 1, \ldots, t$, called teeth. We denote $H'_i = T_i \cap H$ and $T'_i = T_i \backslash H$ for $i = 1, \ldots, t$. The teeth must satisfy $T'_i \ne \varnothing$ and $H'_i \ne \varnothing$ for $i = 1, \ldots, t$; and $T_i \cap T_j = \varnothing$, $\forall\, i, j = 1, \ldots, t, i \ne j$. Then the capacitated H-comb inequality is

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \ge \sum_{i=1}^{t} \big(2k(T'_i) + 2k(H'_i) - 1\big) + 1. \tag{17}$$

Proposition 1. *Capacitated H-comb inequalities (17) are valid for the STTRPSD if t is odd.*

Let us consider two disjoint and nonempty subsets of customers, S_1 and S_2; we will say that S_1 and S_2 are *incompatible* if $k(S_1 \cup S_2) = k(S_1) + k(S_2)$. It can easily be shown that $k(S_1) + k(S_2) - 1 \le k(S_1 \cup S_2) \le k(S_1) + k(S_2)$; if $k(S_1 \cup S_2) = k(S_1) + k(S_2)$, then no vehicle can be saved if we allow the customers in $S_1 \cup S_2$ to be served together instead of using different vehicles for the customers in S_1 and S_2. We will consider only capacitated H-combs for which T'_i and H'_i are *compatible*; i.e., $k(T_i) = k(T'_i) + k(H'_i) - 1$ for all $i = 1, \ldots, t$. It is also shown in the appendix (see Remark 1) that the existence of a tooth $T_{i'}$ such that $T'_{i'}$ and $H'_{i'}$ are incompatible leads to an inequality that can be generated by adding capacity constraints so it is redundant. Note that the inequality for the case where $t = 1$ is also valid and not redundant if $V_d \backslash H \ne \varnothing$ holds. This contrasts with comb inequalities for the CVRP or TSP, which are redundant in the case $t = 1$.

Additionally, if the teeth are pairwise incompatible, the inequality can be greatly improved by adding $t-1$ to the RHS, and the condition that t must be odd can be removed.

Proposition 2. *If all of the teeth of a capacitated H-comb are pairwise incompatible, the following inequality is valid for the STTRPSD:*

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq \sum_{i=1}^{t} \big(2k(T_i') + 2k(H_i')\big). \quad (18)$$

Finally, a tighter inequality than (17) is obtained if we use the bound $r(S)$ instead of $k(S)$. Let

$$g(H, T_1, \ldots, T_t) = \sum_{i=1}^{t} \big(r(T_i') + r(H_i') + r(T_i)\big).$$

Proposition 3. *If $g(H, T_1, \ldots, T_t)$ is odd, then the following inequality is valid for the STTRPSD:*

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq g(H, T_1, \ldots, T_t) + 1. \quad (19)$$

3.5.2. Capacitated T-Comb Inequalities. We now consider combs with satellite depots in the teeth. Let $H \subset V_c$ be the handle, and let $T_i \subset V_c \cup V_d$, for $i = 1, \ldots, t$, $t \geq 1$, be the teeth. We write $H_i' = T_i \cap H$, for $i = 1, \ldots, t$, and $H^c = H \setminus \bigcup_{i=1}^{t} T_i$. The following conditions must be satisfied: $H_i' \neq \varnothing$, $T_i \cap V_d \neq \varnothing$ for $i = 1, \ldots, t$, and $H^c \neq \varnothing$. Then the capacitated T-comb inequality is

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq 2r(H^c) + \sum_{i=1}^{t} 2r(H_i'). \quad (20)$$

Proposition 4. *Capacitated T-comb inequalities (20) are valid for the STTRPSD.*

4. Branch-and-Cut

The integer programming formulation described in §2 and the valid inequalities of §3 are used in a branch-and-cut algorithm to solve the STTRPSD exactly. At each node of the branch-and-cut tree, we solve an LP that contains a given set of constraints and separation procedures search for inequalities violated by the current LP optimal solution. We then add the violated constraints, and the LP is reoptimized. The cycle of optimization and identification of violated constraints iterates until no more violated inequalities are found; this is called the cutting-plane phase. The next step is to use the strong branching strategy of CPLEX (IBM-ILOG 2010) to explore the branch-and-cut tree. A description of the elements of the proposed branch-and-cut algorithm follows.

4.1. Initial Linear Relaxation

The initial LP that is solved at the root node of the branch-and-cut tree includes the objective function (1), customer-degree constraints (2), variable bounds (8) to (12), and the following additional (simple) constraints: (i) Capacity constraints (3) for $S = V_c$; (ii) a set of constraints to ensure that if a trailer point is not visited, it does not have any departing second-level trips: $x_{ij} \leq 2z_i$, $\forall\, i \in V_d$, $\forall\, j \in V_c$; (iii) the complementary constraints of the previous ones; i.e., if a trailer point is visited in the first-level trip, it must have departing second-level trips: $x(\delta(\{i\})) \geq 2z_i$, $\forall\, i \in V_d$; (iv) a constraint to state that the first-level trip begins and ends at the main depot: $y(\delta(\{0\})) = 2$; and (v) a set of so-called logic constraints of the orienteering problem (Fischetti, Salazar-González, and Toth 1998): $y_{ij} \leq z_i$, $\forall\, i, j \in V_d$.

4.2. Separation Procedures

Let $(\bar{x}, \bar{y}, \bar{z})$ be the current LP optimal solution at a given iteration of the cutting-plane phase. In the separation procedures that follow, the weighted graph $G[\bar{x}]$ induced by edges $\{i, j\}$ with $\bar{x}_{ij} > 0$ will be called the support graph. Likewise, $G[\bar{x}, \bar{y}]$ is the support graph induced when the variables $\bar{y}_{ij} > 0$ are also considered. In $G[\bar{x}]$ the weight of the edges $e \in \gamma(V_2)$ are given by $\bar{x}_e$. Additionally, in $G[\bar{x}, \bar{y}]$ the weight of the edges in $\gamma(V_1)$ are given by $\bar{y}_e$.

Several exact and heuristic procedures are employed to separate each family of constraints. Some of them are based on separation procedures for the CVRP (Augerat et al. 1995; Lysgaard, Letchford, and Eglese 2004), the MDMTSP (Benavent and Martínez 2013), and the LRP (Belenguer et al. 2011). In addition, for specific STTRPSD constraints it was necessary to design and implement several new separation procedures.

4.2.1. Capacity and Depot-Degree Constraints. Violated capacity constraints (3) are sought with the separation procedures developed by Augerat et al. (1995). These procedures comprise (i) a heuristic that verifies the violation of these constraints for each connected component of the support graph $G[\bar{x}]$ and those of $G[\bar{x}] \setminus V_d$, (ii) a shrinking heuristic that iteratively shrinks the endpoints of edges with $\bar{x}_{ij} \geq 1$ and verifies the violation of the capacity constraint of the resulting super-node, and (iii) the tabu search procedure of Augerat et al. (1998). All of these heuristics generate subsets $S \subseteq V_c$ that have also been used to check the violation of the depot-degree constraints (16) for each satellite depot $i \in V_d$. We also use heuristic separation procedures for the capacity constraints, developed by Lysgaard, Letchford, and Eglese (2004) for the CVRP.

4.2.2. Path-Elimination Constraints. The abovementioned shrinking heuristic is also used to separate the path-elimination constraints (4–5): when a super-node is connected to more than one satellite depot, the violation of the corresponding constraint is checked. Furthermore, Belenguer et al. (2011) developed an exact polynomial algorithm to separate path-elimination constraints in the LRP. It can be adapted as follows to separate such constraints in the STTRPSD.

Given a pair of customers $\{j, l\}$, constraint (4) can be decomposed into two terms. The first, $\sum_{i\in I'} x_{ij} + \sum_{k\in V_d\setminus I'} x_{kl}$, depends only on the selection of I'; the second, $2x(\gamma(S \cup \{j, l\}))$, depends on the selection of S. Given an LP solution $(\bar{x}, \bar{y}, \bar{z})$, the first term is maximized when $I' = \{i \in V_d: \bar{x}_{ij} \geq \bar{x}_{il}\}$. Now consider the selection of S to maximize the second term. Adding the degree constraints for $S \cup \{j, l\}$, we obtain $2x(\gamma(S\cup\{j, l\})) + x(\delta(S\cup\{j, l\})) = 2(|S| + 2)$. Therefore, maximizing $2x(\gamma(S\cup\{j, l\}))$ is equivalent to minimizing $x(\delta(S \cup \{j, l\}))$, so the goal is to look for the set S' including j and l that minimizes $\bar{x}(\delta(S'))$. This is found by solving a maximum-flow problem on an auxiliary graph $G'[\bar{x}]$ that is built by adding to $G[\bar{x}]$ a source node s connected to j and l with edges of infinite capacity and a sink node t connected with all satellite depots with edges of infinite capacity, and considering the weights of all of the edges in $G[\bar{x}]$ as capacities. After we solve the maximum-flow problem from s to t in $G'[\bar{x}]$, the set S' is composed of the nodes reachable from s when the arcs in the minimum-cut are eliminated. Then given the sets $I' \subset I$ and $S = S'\setminus\{i, j\}$ that maximize the two terms of constraint (4), it is easy to check their violation. This procedure is applied to each pair of customers $\{j, l\}$ contained in each connected component of the support graph $G[\bar{x}]$.

4.2.3. Connection Constraints. For the connection constraints (6) we apply four different separation procedures. Since the connection constraints are in some ways similar to the capacity constraints, the first separation procedure is a heuristic that reuses the subsets $S_c \subseteq V_c$ generated during the capacity constraint separation where the capacity constraint is met. Given S_c, a greedy interchange heuristic looks for a subset $S_d \subseteq V_d$ maximizing the violation of the associated connection constraint. The second separation heuristic verifies the violation of the connection constraint for each connected component of the support graph $G[\bar{x}, \bar{y}]$ and of $G[\bar{x}, \bar{y}]\setminus\{0\}$, respectively.

The third heuristic is a simple tabu search (with only short-term memory) that solves the following optimization problem:

$$\min \zeta(S) = \sum_{i\in V_2\setminus S,\, j\in S_c} \bar{x}_{ij} + k(S_c)\bar{y}(\delta(S_d)) - 2k(S_c), \quad \text{for } S = S_d \cup S_c.$$

During the execution of the tabu search, when a given subset S' with $\zeta(S') < 0$ is found, the corresponding connection constraint is violated.

Finally, we have implemented an exact polynomial algorithm that identifies violated connection constraints in the special case where $k(S_c) = 1$. Let $H(X, A)$ be a directed graph with node set $X = V \cup \{\alpha\}$, where α is a dummy source node, and with arc set A that contains

- two reverse arcs (i, j) and (j, i) for each edge $e = \{i, j\} \in \gamma(V_1) \cup \gamma(V_2)$, each with capacity y_{ij} (x_{ij}) for $e \in \gamma(V_1)$ $(e \in \gamma(V_2))$;
- arcs (i, j) with capacity x_{ij}, for each $i \in V_c$, $j \in V_d$; and
- arcs (α, i) with capacity 2, for each $i \in V_c$.

Then we compute the maximum flow from the dummy node α to the main depot in $H(X, A)$. Let S' be the set of nodes that are reachable from the dummy node after we remove the arcs in the minimum cut. It is easy to show that if the maximum flow is smaller than 2, then the connection constraint for $S_c = S' \cap V_c$ and $S_d = S' \cap V_d$ is violated; otherwise, no connection constraint is violated for a customer subset $S_c \subseteq V_c$ such that $k(S_c) = 1$.

4.2.4. Co-Circuit Constraints. The separation procedures for the co-circuit constraints (13) and (14) are similar, so only those for (13) are described.

Ghiani and Laporte (2000) developed a procedure to separate these constraints for the undirected rural postman problem. It is based on the following idea. Note that the co-circuit constraints can be rewritten as

$$\sum_{e\in\delta(S)\setminus F} x_e + \sum_{e\in F}(1 - x_e) \geq 1,$$
$$S \subseteq V_c,\, F \subseteq E(S\colon V_c\setminus S)\colon |F| \quad \text{is odd.} \tag{21}$$

For a given $S \subseteq V_c$ the left-hand side (LHS) of (21) is the minimum if $F = \{e \in E(S\colon V_c\setminus S)\colon \bar{x}_e \geq 0.5\}$. When $|F|$ is even, either one edge from F is removed or one edge from $\delta(S)\setminus F$ is added to F; we choose the operation that produces the smallest increase in the LHS of (21). The co-circuit constraint associated with S and F will be violated if the LHS is less than 1; otherwise constraint (21) for this set S is not violated for any odd edge subset F.

We use a heuristic algorithm that applies the above procedure for each set $S = \{v\}\colon v \in V_c$. Furthermore, we use an exact procedure based on the separation of blossom inequalities, developed by Letchford, Reinelt, and Theis (2004), for the separation of the co-circuit constraints. This procedure builds the minimum-cut tree (Gomory and Hu 1961) T of a graph G where the nodes are $V_c \cup V_d$ and the edges have weights defined as $w_e = \min\{\bar{x}_e, 1 - \bar{x}_e\}$. Let S^f be the subset of nodes defined by the minimum-weight cut associated with edge f of T. The weight of the cut is equal to the LHS of (21). If the weight of a given edge f is less than 1, the procedure described above is applied to S^f.

4.2.5. Satellite Depot Cuts. A greedy heuristic is used for the separation of the satellite depot cuts (15).

For each open satellite depot $i \in V_d$: $\bar{z}_i > 0$, it adds iteratively to S the customer

$$j^* = \underset{j \in V_c \setminus S:\, \bar{x}_{ij} - \bar{z}_i > 0}{\arg\max} \{\bar{x}_{ij} - \bar{z}_i\}$$

provided the total demand of S does not exceed Q_v.

4.2.6. CVRP Combs and Capacitated Multi-Depot Combs. The heuristic procedures of Lysgaard, Letchford, and Eglese (2004) are used to separate the CVRP combs by previously shrinking the satellite depots into a single depot.

The separation procedures for capacitated H-combs and T-combs are inspired by those developed by Benavent and Martínez (2013) for the MDMTSP. Violated H-combs are identified as follows. First, we build the subgraph G_ϵ of $G[\bar{x}]$ that contains all edges incident with at least one satellite depot and also those edges $e \in \gamma(V_2)$ such that $\epsilon < x_e < 1 - \epsilon$, where we use $\epsilon = 0.3$. Then each connected component of G_ϵ that contains at least one satellite depot is considered as a candidate for the handle of an H-comb. For each candidate handle, say, H, several candidate sets of teeth are generated. We then consider the first candidate set of teeth. Each tooth is a pair of end-nodes of an edge of G_ϵ with one end-node in H and the other not in H; if two such edges share the same end-node, these edges are discarded and the common incident node is removed from H if it belonged to H or added to H otherwise. Then we grow the node sets of the candidate teeth as follows. Let $T_1, \ldots, T_t$ be the set of candidate teeth at one iteration. Then one of the following operations is carried out: one node is added to a tooth, two adjacent nodes are added to a tooth, or the node sets of two teeth are added to (removed from) H. All possible operations are checked, and the one producing the smallest increase in $\bar{x}(\delta(H')) + \sum_{i=1}^{t'} \bar{x}(\delta(T_i))$ is selected, where H' and t' are, respectively, the new handle and the number of teeth after the operation. We then build and check an H-comb inequality as follows. First, we discard the teeth T_i for which H_i' and T_i' are not compatible; then we discard the teeth for which $\bar{x}(\delta(T_i)) \geq 2k(H_i') + 2k(T_i') - 1$. If the number of teeth that remains are even, we recover the tooth with the smallest $\bar{x}(\delta(T_i)) - 2k(H_i') - 2k(T_i') + 1$ and check the resulting comb inequality (17). We use the same set of candidate teeth to build and check inequality (18). In this case, the teeth that are discarded are those for which $\bar{x}(\delta(T_i)) \geq 2k(H_i') + 2k(T_i')$ holds. The resulting teeth are ordered by decreasing demand. Let t be the smallest number such that tooth $t+1$ is compatible with tooth t. Then we check inequality (18) for the current handle H and the first t teeth in the list.

In the branch-and-cut we use the weaker version of the T-comb inequalities (20) that results from substituting $r(\cdot)$ with $k(\cdot)$. The procedures for separating weak T-combs start by generating a list of candidate teeth, each containing at least one satellite depot, as follows. Every satellite depot is used as the starting node set for a tooth. This node set, say, S, is iteratively enlarged by adding a new node (a customer or another satellite depot); the node added at each iteration is the one that produces the smallest increase in $\bar{x}(\delta(S))$. Of the node sets generated in this way, we store those for which $\bar{x}(\delta(S)) < 2$ in a pool that we denote S_T. This pool is used to generate T-combs that are checked for violation via the following procedures. First, we generate T-combs with only one tooth by building, for each tooth in the pool, say, T, a sequence of candidate handles. We start with $H = T \cap V_c$, and we repeatedly add the customer node that produces the smallest increase in $\bar{x}(\delta(H))$, until no new customer node can be added. This procedure is repeated for each $i \in T \cap V_c$, by starting the sequence of handles with $H = T \setminus \{i\} \cap V_c$. The second procedure first selects a maximal subset of pairwise disjoint teeth from the pool S_T that are in the same connected component of $G[\bar{x}]$. Let $T_1, \ldots, T_t$ be such a set of teeth; then, as above, we generate a sequence of candidate handles, starting with all of the customer nodes of the connected component and removing at each iteration one, or two, adjacent nodes from the handle, trying to keep $\bar{x}(\delta(H))$ as small as possible. For each generated set of teeth, $T_1, \ldots, T_t$, and handle H, we build a T-comb inequality by discarding those teeth, $i = 1, \ldots, t$, for which H_i' and H^c are not compatible, and we check for violation. It is easy to see that if $k(H_i' \cup H^c) = k(H_i') + k(H^c)$, then we obtain a strongest weak T-comb inequality by discarding tooth T_i.

4.3. Separation Strategy

At each node of the branch-and-cut tree we use a cutting-plane scheme in which, at each iteration, we use the separation procedures to find cuts that strengthen the LP relaxation. The strategy used when calling the separation procedures is important for the performance of a branch-and-cut algorithm.

The strategy we use in the root node consists of four phases, where each phase is executed only if the preceding phases have not been able to find a given number of violated constraints. In the first phase we call the connected components and the shrinking heuristics for the capacity (3), path elimination (4), connection (6), and depot-degree (16) constraints.

The second phase consists of running sequentially the tabu search heuristic for the capacity constraints; the connected components heuristic for the connection constraints; and the Lysgaard, Letchford, and Eglese (2004) heuristics for the capacity constraints, where each heuristic is called only if the number of violated constraints found so far is less than $|V_c|/2$.

In the third phase we call the following procedures sequentially whenever the number of violated constraints found so far is less than $|V_c|/4$: the greedy heuristic for the satellite depot constraints (15); the heuristic procedure for the co-circuit constraints (13–14); the tabu search heuristic for the connection constraints; the heuristic for T-combs (20) with one tooth; the heuristic for H-combs (17–18) with small teeth (the demand of each tooth is not greater than Q_v); and the heuristics of Lysgaard, Letchford, and Eglese (2004) for the CVRP combs.

The fourth phase consists of calling sequentially the following procedures in an order that is randomly changed at each iteration: the H-comb heuristic (without any restriction on the size of the teeth), the T-comb heuristic (without any restriction on the number of teeth), the exact procedures for the co-circuit constraints (13–14), the exact procedure for the connection constraints with $k(S_c) = 1$, and the exact procedure for the path-elimination constraints. Each procedure in this phase is called only if the number of violated constraints found so far is less than $|V_c|/4$.

The strategy for the other nodes of the branch-and-cut tree is simpler. It consists of executing the first phase and then, if the number of violated constraints found so far is less than $|V_c|/2$, calling the connected-components heuristic for the connection constraints.

5. Computational Experiments

The experiments of this section were run using a computer with an Intel Core i7-3770 processor running at 3.4 GHz under Windows 7 (64 bits) with 16 GB of RAM. We used the test bed of 32 instances described by Villegas et al. (2010) (available at http://hdl.handle.net/1992/1124). The instances have between 25 and 200 customers and five to 20 satellite depots. For each problem size there are two levels of truck capacity and two types of instances: one with clustered customers (type *c*) and one with randomly distributed customers (type *rd*). Each instance is named with the convention n-p-$(0.001 \times Q_v)$-t, where t indicates the instance type. We implemented the branch-and-cut algorithm in Visual C++, and it calls CPLEX 12.4 for the solution of the LPs. We used a running time of three hours for the branch-and-cut and, for each instance, we used the cost of the best known solution reported by Villegas et al. (2010) as an initial upper bound.

Tables 1 and 2 show the results obtained with the branch-and-cut algorithm. They present the value of the initial upper bound (UB), the best lower bound (LB) obtained by the branch-and-cut algorithm, the gap between UB and LB ($Gap = (UB - LB/UB) \times 100$), the number of nodes in the branch-and-cut tree (*Nodes*), the number of cuts found (*No. of Cuts*), and the CPU time broken down into the time for the solution of the linear relaxations (*LP*) the separation procedures (*Sep.*) and the computation of the upper bound (*Heur.*). The overall computation time is also shown (*Overall*). The last three columns of the tables show the lower bound found at the root node (*LB0*), the gap between UB and LB0 (*Gap0*), and the CPU time for the root node (*Time0*). All of the CPU times are in seconds. The values in bold indicate proven optima, i.e., LB is equal to the cost of the best known solution. In those instances where the branch-and-cut tree were not completely explored because of the time limit, LB indicates the best proven lower

Table 1 Results of the Branch-and-Cut Algorithm for STTRPSD Instances with up to 50 Customers

						Time (s)						
Name	*UB*	*LB*	*Gap* (%)	*Nodes*	*No. of cuts*	*LP*	*Sep.*	*Heur.*[a]	*Overall*	*LB0*	*Gap0* (%)	*Time0* (s)
25-10-1-c	**386.45**	**386.45**	0.00	17	436	2.07	1.33	132.00	135.40	371.45	3.88	0.86
25-10-1-rd	**573.96**	**573.96**	0.00	22	362	1.73	0.75	138.00	140.48	551.91	3.84	1.00
25-10-2-c	**380.86**	**380.86**	0.00	19	290	1.81	0.84	126.00	128.65	368.87	3.15	0.87
25-10-2-rd	**506.37**	**506.37**	0.00	36	303	2.54	0.69	126.00	129.23	479.16	5.37	0.51
25-5-1-c	**405.46**	**405.46**	0.00	26	197	0.94	0.67	114.00	115.61	387.54	4.42	0.84
25-5-1-rd	**584.03**	**584.03**	0.00	8	192	0.59	0.45	126.00	127.04	565.43	3.19	0.39
25-5-2-c	**374.79**	**374.79**	0.00	12	190	0.58	0.53	114.00	115.11	363.80	2.93	0.44
25-5-2-rd	**508.48**	**508.48**	0.00	13	146	0.76	0.42	102.00	103.18	487.51	4.12	0.28
50-10-1-c	**387.83**	**387.83**	0.00	211	655	5.60	1.00	822.00	828.60	374.16	3.52	0.89
50-10-1-rd	**811.28**	**811.28**	0.00	1,124	1,011	43.38	3.00	720.00	766.38	758.29	6.53	2.95
50-10-2-c	**367.01**	**367.01**	0.00	12	356	1.70	1.42	750.00	753.12	364.65	0.64	1.58
50-10-2-rd	**731.53**	**731.53**	0.00	715	1,072	32.62	2.56	630.00	665.18	678.81	7.21	2.26
50-5-1-c	**583.07**	**583.07**	0.00	6	604	1.01	1.48	708.00	710.49	558.41	4.23	1.25
50-5-1-rd	**870.51**	**870.51**	0.00	190	813	6.54	2.06	612.00	620.60	832.55	4.36	0.90
50-5-2-c	**516.98**	**516.98**	0.00	10	453	1.40	0.89	570.00	572.29	493.55	4.53	2.07
50-5-2-rd	**766.03**	**766.03**	0.00	338	569	9.17	1.53	588.00	598.70	733.98	4.18	1.83
Average			0.00			7.03	1.23	398.63	406.88		4.13	1.18

[a]Total computing time for 10 runs of the metaheuristic of Villegas et al. (2010) that found the best upper bound, on a computer with an Intel Pentium D 945 processor at 3.4 GHz with 1,024 MB of RAM running under Windows XP Professional.

Table 2 Results of the Branch-and-Cut Algorithm for Large Instances of the STTRPSD

						Time (s)						
Name	*UB*	*LB*	*Gap* (%)	*Nodes*	*No. of cuts*	*LP*	*Sep.*	*Heur.*[a]	*Overall*	*LB0*	*Gap0* (%)	*Time0* (s)
100-10-1-c	**614.02**	**614.02**	0.00	9,219	2,419	1,287.39	47.11	3,480.00	4,814.50	576.99	6.03	19.13
100-10-1-rd	1,275.76	1,214.23	4.82	18,765	2,516	10,744.66	55.38	2,292.00	13,092.04	1,136.18	10.94	13.82
100-10-2-c	**547.44**	**547.44**	0.00	57	1,502	25.97	24.49	3,894.00	3,944.46	518.20	5.34	21.33
100-10-2-rd	1,097.28	1,075.00	2.03	32,980	2,611	10,711.15	88.89	2,604.00	13,404.04	987.07	10.04	20.84
100-20-1-c	**642.61**	**642.61**	0.00	1,095	2,496	328.30	28.14	2,892.00	3,248.44	618.36	3.77	44.23
100-20-1-rd	1,143.10	1,093.01	4.38	10,212	2,436	10,743.89	56.16	3,012.00	13,812.05	1,007.52	11.86	41.20
100-20-2-c	**581.56**	**581.56**	0.00	443	2,176	224.13	55.10	3,600.00	3,879.23	555.30	4.52	51.15
100-20-2-rd	1,060.75	1,033.02	2.61	16,686	3,468	10,693.91	106.13	2,946.00	13,746.04	941.57	11.24	32.64
200-10-1-c	822.52	784.43	4.63	3,462	4,279	10,635.02	165.05	11,544.00	22,344.07	743.31	9.63	100.06
200-10-1-rd	1,761.10	1,613.60	8.38	1,694	4,931	10,686.44	113.68	8,340.00	19,140.12	1,546.93	12.16	150.07
200-10-2-c	714.33	691.29	3.23	8,198	3,304	10,656.33	143.75	11,682.00	22,482.08	640.00	10.41	162.97
200-10-2-rd	1,445.94	1,398.05	3.31	5,603	3,013	10,663.55	136.52	9,288.00	20,088.07	1,311.14	9.32	83.44
200-20-1-c	909.46	883.85	2.82	19,515	4,848	10,535.49	264.59	13,746.00	24,546.08	842.25	7.39	89.95
200-20-1-rd	1,614.18	1,510.18	6.44	2,577	3,757	10,661.88	138.22	12,282.00	23,082.10	1,424.23	11.77	203.41
200-20-2-c	815.51	807.84	0.94	13,073	4,784	10,614.12	185.97	11,730.00	22,530.09	784.34	3.82	68.37
200-20-2-rd	1,413.32	1,346.35	4.74	1,736	4,016	10,613.52	186.61	12,084.00	22,884.13	1,274.25	9.84	250.80
Average			3.02			8,114.11	112.24	7,213.50	15,439.85		8.63	84.59

[a]Total computing time for 10 runs of the metaheuristic of Villegas et al. (2010) that found the best upper bound, on a computer with an Intel Pentium D 945 processor at 3.4 GHz with 1,024 MB of RAM running under Windows XP Professional.

bound for the corresponding instance, i.e., the least lower bound of all of the active nodes.

As can be seen in Table 1, all of the instances with up to 50 customers were solved optimally, and the running times are rather small. However, the gaps at the root node are quite large (4.13% on average) and the gap is less than 1% (0.64%) in only one instance (50-10-2-c), whereas the number of nodes in the branch-and-cut tree are small. Large gaps at the root node can be explained because the variables associated with the use of the satellite depots (the z-variables) usually have small values in the fractional solutions of the linear relaxation. We tried running a partial branch-and-cut where only the z-variables were declared integer. The lower bounds show an average gap of only 0.68% on these instances, with an average CPU time of 33.12 seconds, which shows that branching on the z-variables is effective. It is worth noting that the best solutions found by multistart evolutionary local search (Villegas et al. 2010) were optimal in all of these problems.

Table 2 summarizes the results of the branch-and-cut algorithm for large instances. As can be seen, four 100-customer instances were solved optimally, and for one 200-customer instance (200-20-2-c) the final gap is small (0.94%). For the remaining large instances the branch-and-cut produced large gaps and did not find feasible solutions. Additional developments will be needed to solve these instances.

Note that all of the 100-customer instances that could be solved are of type c (clustered). Furthermore, for the same size and truck capacity, the rd instances have in general larger gaps and longer running times than the c instances. This suggests that instances with randomly distributed customers are more difficult.

We performed an additional experiment to evaluate the contribution of each family of valid inequalities to the effectiveness of the branch-and-cut algorithm. For this purpose we solved the root node with variants of the code, including different sets of valid inequalities. The base version of the code includes only the separation procedures for the capacity (3), path-elimination (4–5), and connection constraints (6), i.e., the constraints of the initial formulation. The other five variants were obtained by adding to the base version the separation procedures for each family of inequalities. For each variant, Table 3 reports the gap with respect to the upper bound ($Gap_{\text{version}} = (UB - LB_{\text{version}}/UB) \times 100$) and the closed gap with respect to the lower bound of the base version ($\textit{Closed gap}_{\text{version}} = (LB_{\text{version}} - LB_{\text{base}}/UB - LB_{\text{base}}) \times 100$).

As can be seen in Table 3, each family of constraints reduces the average gap of the base version. The largest improvement was obtained with the capacitated multidepot combs (CMD combs), followed by the co-circuit and depot-degree constraints. Note that for three instances the closed gap for the depot-degree constraints are negative; this is because the separation procedures are in fact heuristic algorithms. When all of the constraints are used, the average gap of the lower bound in the root node is 6.38% (as can be deduced from the results shown in Tables 1 and 2), which is better than the average gap of 8.77% obtained with the base version.

Table 3 **Contribution of Each Family of Valid Inequalities to the Lower Bound at the Root Node**

Instance name	Base version	Co-circuit		Satellite depot cuts		Depot-degree		CVRP combs		CMD combs	
	Gap	*Gap*	*Closed gap*	*Gap*	*Closed gap*	*Gap*	*Closed gap*	*Gap*	*Closed gap*	*Gap*	*Closed gap*
25-10-1-c	5.89	5.61	4.83	5.80	1.54	5.66	3.90	5.87	0.28	5.41	8.22
25-10-1-rd	8.04	5.51	31.50	6.81	15.27	5.44	32.34	8.04	0.00	6.78	15.66
25-10-2-c	9.13	9.02	1.16	8.93	2.12	8.67	5.01	9.13	0.00	8.56	6.25
25-10-2-rd	7.74	6.99	9.68	7.71	0.35	7.68	0.82	7.74	0.00	6.28	18.85
25-5-1-c	9.86	9.08	7.87	9.80	0.62	9.42	4.44	9.86	0.00	8.42	14.56
25-5-1-rd	4.97	4.58	7.77	4.82	2.96	4.97	0.00	4.92	0.91	4.36	12.12
25-5-2-c	9.60	9.56	0.43	9.56	0.49	7.80	18.82	9.60	0.00	9.14	4.79
25-5-2-rd	6.79	6.58	3.06	6.77	0.25	6.79	0.00	6.76	0.46	6.44	5.10
50-10-1-c	4.91	4.42	9.86	4.78	2.60	4.29	12.64	4.84	1.33	4.13	15.73
50-10-1-rd	9.02	8.36	7.34	8.49	5.90	8.71	3.39	8.69	3.62	8.10	10.15
50-10-2-c	3.76	2.26	39.86	2.94	21.76	2.72	27.51	2.86	24.01	2.06	45.07
50-10-2-rd	8.79	7.94	9.69	8.70	0.95	8.75	0.45	8.30	5.53	7.91	10.02
50-5-1-c	6.97	6.66	4.45	6.93	0.62	5.56	20.20	6.94	0.37	6.60	5.32
50-5-1-rd	5.49	5.34	2.67	4.97	9.42	5.35	2.55	5.31	3.24	5.37	2.27
50-5-2-c	9.67	9.29	4.01	9.56	1.20	7.45	22.96	9.67	0.06	9.25	4.40
50-5-2-rd	6.22	5.46	12.18	6.05	2.81	5.77	7.25	5.20	16.49	6.08	2.31
100-10-1-c	8.00	7.62	4.68	7.58	5.23	6.89	13.81	7.77	2.90	7.70	3.78
100-10-1-rd	12.40	12.12	2.28	11.52	7.10	12.01	3.19	12.23	1.40	12.35	0.44
100-10-2-c	7.42	6.70	9.73	7.11	4.12	7.05	5.02	7.14	3.72	6.79	8.46
100-10-2-rd	11.60	11.01	5.02	11.21	3.31	11.67	−0.62	11.43	1.42	11.41	1.58
100-20-1-c	7.10	4.76	32.93	5.65	20.40	5.25	26.08	5.49	22.59	4.93	30.54
100-20-1-rd	13.61	12.33	9.37	12.89	5.27	13.09	3.78	13.16	3.27	12.35	9.23
100-20-2-c	5.78	4.89	15.35	5.53	4.21	5.70	1.41	5.60	3.06	5.02	13.18
100-20-2-rd	12.56	11.81	5.94	12.42	1.14	12.70	−1.13	12.19	2.96	11.99	4.58
200-10-1-c	10.45	10.34	1.09	9.74	6.81	10.45	0.03	10.38	0.71	10.21	2.26
200-10-1-rd	14.36	14.26	0.67	13.03	9.26	13.23	7.82	14.17	1.33	14.25	0.72
200-10-2-c	11.71	11.06	5.48	10.98	6.24	11.20	4.34	11.20	4.33	11.29	3.54
200-10-2-rd	10.69	10.42	2.54	9.87	7.69	10.42	2.53	10.50	1.81	10.57	1.13
200-20-1-c	9.18	8.67	5.58	8.72	5.03	8.43	8.15	8.96	2.38	8.65	5.83
200-20-1-rd	13.17	12.88	2.20	12.33	6.37	12.55	4.71	12.88	2.24	12.52	4.97
200-20-2-c	4.93	4.44	9.95	4.71	4.45	4.50	8.72	4.74	3.91	4.44	9.83
200-20-2-rd	10.80	10.37	4.04	10.52	2.61	10.88	−0.69	10.58	2.08	10.37	3.97
Average	8.77	8.14	8.54	8.33	5.25	8.16	7.79	8.50	3.64	8.12	8.90

6. Conclusions and Future Work

In this paper we have proposed a new formulation and several valid inequalities for the single truck and trailer routing problem with satellite depots. In particular, the capacitated multidepot comb inequalities take the truck capacity into account and allow the satellite depots to be in different parts of the comb. These inequalities could be useful in other routing problems involving capacitated vehicles and several depots. We have developed two new exact polynomial algorithms to separate the path-elimination constraints and a special case of the connection constraints, and we have designed a number of heuristic separation procedures for all of the families of constraints.

All of these procedures have been embedded within a branch-and-cut algorithm capable of solving all of the tested instances with up to 50 customers and 10 satellite depots, and four instances with 100 customers and up to 20 satellite depots. However, further research is required to develop an exact method able to cope with larger real-life instances. Future work could include finding new valid inequalities specific to the STTRPSD, adapting other constraints from the CVRP, and improving the overall branch-and-cut scheme.

Acknowledgments

The first and second authors acknowledge support from the Generalitat Valenciana [Project GVPROMETEO2013-049], the first and third authors also thank the Ministerio de Ciencia e Innovación [Project DPI2011-24977]. The second author received support from the Ministerio de Economía y Competitividad [Project MTM2012-36163-C06-02]. Project MTM2012-36163-C06-02 is funded in cooperation with the Fondo Europeo de Desarrollo Regional (FEDER). The fourth author acknowledges the Centre of Operations Research, Management Science, and Information Systems (CORMSIS) for their support. The fourth, fifth, and sixth authors were partially supported by the Champagne-Ardenne Regional Council, and the sixth author also thanks the Administrative Department of Science, Technology, and Innovation-Colciencias. This work was done while the sixth author was a joint Ph.D. student at the Centro para la Optimización y Probabilidad Aplicada (COPA)–Universidad de los Andes and the Laboratoire d'Optimisation des Systèmes Industriels (LOSI)–Université de Technologie de Troyes.

Appendix. Proofs of Validity of Capacitated H- and T-Comb Inequalities

Given a subset of customers S and a feasible STTRPSD solution, we say that $(v_1, \ldots, v_r)$ is a maximal path in S if (i) $v_1, \ldots, v_r \in S$; (ii) it is a path followed by a vehicle in the solution; and (iii) the edges incident with the end nodes, say, $\{u, v_1\}$ and $\{v_r, w\}$, satisfy $u \notin S$ and $w \notin S$.

Proof of Proposition 1. Let us consider any feasible solution for the STTRPSD. For every tooth T_i, $i = 1, \ldots, t$ define

d_i: number of maximal paths of the solution in H'_i;
f_i: number of maximal paths of the solution in T'_i;
c_i: number of edges of the solution with one endpoint in H'_i and the other in T'_i.

Then $x(\delta(H)) \geq \sum_{i=1}^t c_i$, and since each of the c_i edges joins one maximal path in H'_i with another in T'_i, $x(\delta(T_i)) = 2d_i + 2f_i - 2c_i$, for $i = 1, \ldots, t$. Therefore,

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq \sum_{i=1}^{t} (2d_i + 2f_i - c_i). \tag{22}$$

Note that the number of maximal paths of the solution in T_i is exactly $d_i + f_i - c_i$, so $d_i + f_i - c_i \geq k(T_i) \geq k(T'_i) + k(H'_i) - 1$; on the other hand, $d_i \geq k(H'_i)$ and $f_i \geq k(T'_i)$. Adding these last three inequalities, we obtain $2d_i + 2f_i - c_i \geq 2k(T'_i) + 2k(H'_i) - 1$ for $i = 1, \ldots, t$. Then combining these inequalities with (22) we have $x(\delta(H)) + \sum_{i=1}^t x(\delta(T_i)) \geq \sum_{i=1}^t (2k(T'_i) + 2k(H'_i) - 1)$. Since t is odd, the RHS of this inequality is odd, whereas the LHS is even for every solution, so we can add 1 to the RHS and the inequality (17) results. □

Remark 1. Let us assume that one tooth, say, i', satisfies $k(T_{i'}) = k(T'_{i'}) + k(H'_{i'})$. Consider then the comb that results by removing tooth i' but leaving the handle unchanged; since the number of teeth is now even, we cannot add 1 to the RHS of the resulting inequality as at the end of the preceding proof. If we add to the obtained inequality the capacity constraint $x(\delta(T_{i'})) \geq 2k(T'_{i'}) + 2k(H'_{i'})$, we also obtain (17) so, in this case, it is redundant.

The case t=1 leads to a redundant inequality if $V_d \backslash H = \varnothing$. Indeed, the capacity constraints imply the following:

$$x(\delta(H)) \geq 2k(T'_1) \geq k(T'_1) + 1 \quad \text{and}$$
$$x(\delta(T_1)) \geq 2k(T_1) \geq k(T_1) + k(H'_1) \geq k(T'_1) + k(H'_1) - 1 + k(H'_1);$$

adding these two inequalities gives inequality (17) for $t = 1$. Note, on the other hand, that the inequality for $t = 1$ is also valid and not redundant if $V_d \backslash H \neq \varnothing$ because, in this case, there are solutions for which $x(\delta(H)) = 0$.

Proof of Proposition 2. Note that inequalities (17) and (18) are identical if $t = 1$, so (18) is true for $t = 1$. We assume that (18) is true for combs with at most $t - 1$ teeth, and we prove it for a comb with t teeth. Let us consider any feasible solution for the STTRPSD. We consider two cases.

1. Assume that there is at least one tooth, say, T_r, for which the number of trips visiting its customers is at least $k(T'_r) + k(H'_r)$. Note that for the other teeth this number is at least $k(T_i) \geq k(T'_i) + k(H'_i) - 1$. Let us denote by P the inequality (18) for the current comb and by P' the inequality for the comb that results by removing tooth T_r but leaving the handle unchanged. By the induction hypothesis, the LHS of P' for the solution is at least $\sum_{i=1, i \neq r}^t (2k(T'_i) + 2k(H'_i))$. Because the LHS of P for the solution is equal to the LHS of P' plus $x(\delta(T_r))$, and this is at least $2k(T'_r) + 2k(H'_r)$, the inequality (18) for P follows.

2. Assume now that the number of trips visiting the customers of T_i is $k(T'_i) + k(H'_i) - 1$ for all $i = 1, \ldots, t$. Let T_i and T_j be any two teeth, and let α be the number of trips that visit customers in both of these teeth. Then the number of different trips that visit customers of $T_i \cup T_j$ is $k(T'_i) + k(H'_i) + k(T'_j) + k(H'_j) - 2 - \alpha$. Since T_i and T_j are incompatible, the number of trips visiting customers of $T_i \cup T_j$ is at least $k(T_i \cup T_j) = k(T_i) + k(T_j) = k(T'_i) + k(H'_i) + k(T'_j) + k(H'_j) - 2$, which implies that $\alpha = 0$. This means that there is no trip of the solution that visits customers from two different teeth. On the other hand, since the number of trips visiting customers of a tooth T_i is $k(T'_i) + k(H'_i) - 1$, at least one of these trips will visit customers of both T'_i and H'_i (otherwise, the number of trips would be at least $k(T'_i) + k(H'_i)$). This trip will cross at least twice the edge cutset $\delta(H)$. Because this is true for every tooth and the trips are different for each tooth, we conclude that $x(\delta(H)) \geq 2t$, and since $\sum_{i=1}^t x(\delta(T_i)) \geq \sum_{i=1}^t (2k(T'_i) + 2k(H'_i) - 1)$, we are done. □

Proof of Proposition 3. The proof of this proposition is similar to that of Proposition 1. We simply substitute the inequalities $d_i + f_i - c_i \geq k(T_i) \geq k(T'_i) + k(H'_i) - 1$, $d_i \geq k(T'_i)$, and $f_i \geq k(H'_i)$ in the proof of Proposition 1 with $d_i + f_i - c_i \geq r(T_i)$, $d_i \geq r(T'_i)$ and $f_i \geq r(H'_i)$. We thus obtain $x(\delta(H)) + \sum_{i=1}^t x(\delta(T_i)) \geq \sum_{i=1}^t (r(T'_i) + r(H'_i) + r(T_i)) = g(H, T_1, \ldots, T_t)$, and since $g(H, T_1, \ldots, T_t)$ is odd, (19) follows. □

Before proving Proposition 4 we need to prove the following lemma.

Lemma 1. *Capacitated T-comb inequality (20) is valid for $t = 1$.*

Proof of Lemma 1. Recall that inequality (20) for $t = 1$ is

$$x(\delta(H)) + x(\delta(T_1)) \geq 2r(H^c) + 2r(H'_1).$$

Let $\bar{x}$ be any feasible solution and let us denote by α the number of trips visiting at least one customer of H^c and at least one of H'_1. Furthermore, let β be the number of trips that visits at least one customer of H^c and no customer of H'_1; similarly, let λ be the number of trips that visits at least one customer of H'_1 and no customer in H^c. Note that any trip that visits customers either in H^c or in H'_1 contributes to the LHS of (6) in at least two units, whereas any trip visiting customers of both H^c and H'_1 contributes to that LHS in at least four units. Therefore,

$$\bar{x}(\delta(H)) + \bar{x}(\delta(T_1)) \geq 4\alpha + 2\beta + 2\lambda$$
$$= 2(\alpha + \beta) + 2(\alpha + \lambda) \geq 2r(H^c) + 2r(H'_1),$$

where the last inequality follows because $\alpha + \beta$ is the number of trips visiting customers in H^c and this number is at least $r(H^c)$. Similarly, $\alpha + \lambda \geq r(H'_1)$. □

Proof of Proposition 4. We may assume without loss of generality that there is at least one satellite depot in $V_d \backslash \bigcup_{i=1}^t T_t$. This is because if a feasible solution not satisfying (20) existed for an instance with $V_d \backslash \bigcup_{i=1}^t T_t = \varnothing$, the same solution would also violate (20) for an instance with the same set of customers and a set of satellite depots containing an additional satellite depot in $V_d \setminus \bigcup_{i=1}^{t} T_t$.

We prove the T-comb inequality (20) by induction on the number of teeth t. The above lemma proves the case $t = 1$. We assume that the inequality is valid for a number of teeth less than t, and we will prove the inequality for t teeth.

Let us now consider a T-comb inequality P with t teeth, and let P' be the comb that results by removing the last tooth T_t and whose handle is $H_0 = H \setminus T_t$

$$x(\delta(H_0)) + \sum_{i=1}^{t-1} x(\delta(T_i)) \geq 2r(H^c) + \sum_{i=1}^{t-1} 2r(H_i'). \qquad (23)$$

Let $\bar{x}$ be a feasible solution of the STTRPSD. The LHS of the comb inequality P for this solution can be written as

$$\bar{x}(\delta(H)) + \sum_{i=1}^{t} \bar{x}(\delta(T_i)) = \bar{x}(\delta(H_0)) + \sum_{i=1}^{t-1} \bar{x}(\delta(T_i)) + \bar{x}(\delta(T_t)) + \bar{x}(\delta(H_t')) - 2\bar{x}(E(H_0\colon H_t')). \qquad (24)$$

By induction, we know that $\bar{x}(\delta(H_0) + \sum_{i=1}^{t-1} \bar{x}(\delta(T_i)) = 2r(H^c) + \sum_{i=1}^{t-1} 2r(H_i') + h'$ where $h' \geq 0$ is the slack for which the solution satisfies the P' inequality. Let $b_1 = \bar{x}(\delta(T_t))$, $b_2 = \bar{x}(\delta(H_t'))$, and $b_3 = \bar{x}(E(H_0\colon H_t'))$. We must prove that $b_1 + b_2 - 2b_3 \geq 2r(H_t') - h'$. Because the number of trips visiting customers of H_t' is at least $r(H_t')$, the solution contains at least $r(H_t')$ maximal paths in H_t' that belong to different trips. In what follows we analyze the incidence of each of these maximal paths in the value of $b_1 + b_2 - 2b_3$ and of h'. We distinguish several cases depending on the two edges that connect the extremes of the path with nodes in $V_2 \setminus H_t'$:

1. The two edges belong to $E(H_t'\colon T_t \setminus H_t')$. This case increments $b_1 + b_2 - 2b_3$ by $0 + 2 - 0 = 2$.

2. One edge is in $E(H_t'\colon T_t \setminus H_t')$ and the other in $E(H_t'\colon H_0)$. Then $b_1 + b_2 - 2b_3$ is incremented by $1 + 2 - 2 = 1$, but that trip contains a path starting at a node in T_t and ending at a node not in T_t, so finally the trip must cross another time $\delta(T_t)$, and the total increment resulting from this trip will be at least 2.

3. One edge is in $E(H_t'\colon T_t \setminus H_t')$ and the other in $E(H_t'\colon V_2 \setminus (H \cup T_t))$. Then $b_1 + b_2 - 2b_3$ is incremented by at least $1 + 2 - 0 = 3$.

4. One edge is in $E(H_t'\colon H_0)$ and the other in $E(H_t'\colon V_2 \setminus (H \cup T_t))$. Then $b_1 + b_2 - 2b_3$ is incremented by $2 + 2 - 2 = 2$.

5. Both edges are in $E(H_t'\colon V_2 \setminus (H \cup T_t))$, which increments $b_1 + b_2 - 2b_3$ by at least $2 + 2 - 0 = 4$.

6. Both edges are in $E(H_t'\colon H_0)$, which increments $b_1 + b_2 - 2b_3$ by $2 + 2 - 4 = 0$. Note that this trip goes out from H_0, visits a set of customers outside the comb P', and then enters H_0 again. Let $v_1, v_2, \ldots, v_{r-1}, v_r$ be this section of the trip, where $v_1, v_r \in H_0$, and $v_2, v_{r-1} \in V_2 \setminus (H_0 \cup \bigcup_{i=1}^{t-1} T_i)$. The existence of this trip implies that the slack h' is two units larger for the following reason. Let us consider another feasible solution in which customers in section $v_2, \ldots, v_{r-1}$ form a new trip that is directly connected to a satellite depot outside the comb P', and the two customers v_1, v_r are connected with an edge to close the original trip. Then the LHS of (23) for this solution is at least two units lower than the LHS for the original one, which implies that the original solution satisfies the inequality with a slack of at least two units.

Overall, we have seen that each of the $r(H_t')$ trips adds at least two units to $b_1 + b_2 - 2b_3$, or, as in the last case, the slack h' is at least two units greater. Therefore, $b_1 + b_2 - 2b_3 \geq 2r(H_t') - h'$, and we are done. □

References

Augerat P, Belenguer JM, Benavent E, Corberán Á, Naddef D (1998) Separating capacity constraints in the CVRP using tabu search. *Eur. J. Oper. Res.* 106(2–3):546–557.

Augerat P, Belenguer JM, Benavent E, Corberán Á, Naddef D, Rinaldi G (1995) Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report RR-949-M, ARTEMIS-IMAG, Université Joseph Fourier, Grenoble, France.

Baldacci R, Mingozzi A, Roberti R, Wolfler Calvo R (2013) An exact algorithm for the two-echelon capacitated vehicle routing problem. *Oper. Res.* 61(2):298–314.

Belenguer JM, 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.

Benavent E, Martínez A (2013) Multi-depot multiple TSP: A polyhedral study and computational results. *Ann. Oper. Res.* 207(1): 7–25.

Caramia MC, Guerriero F (2010a) A heuristic approach for the truck and trailer routing problem. *J. Oper. Res. Soc.* 61(7):1168–1180.

Caramia MC, Guerriero F (2010b) A milk collection problem with incompatibility constraints. *Interfaces* 40(2):130–143.

Chao IM (2002) A tabu search method for the truck and trailer routing problem. *Comput. Oper. Res.* 29(1):33–51.

Contardo C, Cordeau J-F, Gendron B (2013) A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optim.* 10(4):263–295.

Contardo C, Hemmelmayr VC, Crainic TG (2012) Lower and upper bounds for the two-echelon capacitated location-routing problem. *Comput. Oper. Res.* 39(12):3185–3199.

Crainic TG, Mancini S, Perboli G, Tadei R (2011) Multi-start heuristics for the two-echelon vehicle routing problem. Merz P, Hao JK, eds. *Evolutionary Computation in Combinatorial Optimization, EvoCOP 2011*, Vol. 6622. Lecture Notes Comput. Sci. (Springer, Berlin), 179–190.

Crainic TG, Mancini S, Perboli G, Tadei R (2013) GRASP with path relinking for the two-echelon vehicle routing problem. Di Gaspero L, Schaerf A, Stutzle T, eds. *Advances in Metaheuristics*, Vol. 53. Oper. Res./Comput. Sci. Interfaces Series (Springer, New York), 113–125.

Cuda R, Guastaroba G, Speranza M (2015) A survey on two-echelon routing problems. *Comput. Oper. Res.* 55:185–199.

Derigs U, Pullmann M, Vogel U (2013) Truck and trailer routing—Problems, heuristics and computational experience. *Comput. Oper. Res.* 40(2):536–546.

Drexl M (2007) On some generalized routing problems. Doctoral dissertation, Faculty of Business and Economics, Rheinisch-Westfaelische Technische Hochschule, Aachen University, Aachen, Germany.

Drexl M (2011) Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem. *Revista Métodos Cuantitativos Econom. Empresa* 12:5–38.

Drexl M (2014) Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. *Networks* 63(1): 119–133.

Fischetti M, Salazar-González JJ, Toth P (1998) Solving the orienteering problem through branch-and-cut. *INFORMS J. Comput.* 10(2):133–148.

Ghiani G, Laporte G (2000) A branch-and-cut algorithm for the undirected rural postman problem. *Math. Programming* 87(3): 467–481.

Gomory RE, Hu TC (1961) Multi-terminal network flows. *J. Soc. Indust. Appl. Math.* 9(4):551–556.

Gonzalez-Feliu J, Perboli G, Tadei R, Vigo D (2007) The two-echelon capacitated vehicle routing problem. Technical report DEIS OR.INGCE 2007/2(R), DEIS, 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.

Hoff A (2012) Milk collection in western Norway using trucks and trailers. *ODYSSEUS 2012, 5th Internat. Workshop Freight Transportation Logist., Mykonos, Greece*, 180–183.

Hoff A, Løkketangen A (2007) A tabu search approach for milk collection in western Norway using trucks and trailers. *TRISTAN VI: Sixth Triennial Sympos. Transportation Anal., Phuket, Thailand.*

IBM-ILOG (2010) CPLEX 12.1. User's Manual. ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf.

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

Laporte G (2007) What you should know about the vehicle routing problem. *Naval Res. Logist.* 54(8):811–819.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Sci.* 43(4):408–416.

Letchford AN, Reinelt G, Theis DO (2004) A faster exact separation algorithm for blossom inequalities. Nemhauser GL, Bienstock D, eds. *Integer Programming and Combinatorial Optimization*, Vol. 3064. Lecture Notes Comput. Sci. (Springer, Berlin), 196–205.

Lin SW, Yu VF, Chou SY (2009) Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Comput. Oper. Res.* 36(5):1683–1692.

Lin SW, Yu VF, Chou SY (2010) A note on the truck and trailer routing problem. *Expert Systems Appl.* 37(1):899–903.

Lin SW, Yu VF, Lu CC (2011) A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems Appl.* 38(12):15244–15252.

Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Programming* 100(2):423–445.

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.

Pureza V, Morabito R, Reimann M (2012) Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *Eur. J. Oper. Res.* 218(3):636–647.

Ralphs TK (2003) Parallel branch and cut for capacitated vehicle routing. *Parallel Comput.* 29(5):607–629.

Santos FA, da Cunha AS, Mateus GR (2013) Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. *Optim. Lett.* 7(7):1537–1547.

Scheuerer S (2006) A tabu search heuristic for the truck and trailer routing problem. *Comput. Oper. Res.* 33(4):894–909.

Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N (2009) GRASP/Evolutionary local search hybrids for a truck and trailer routing problem. *VIII Metaheuristic Internat. Conf. (MIC 2009), Hamburg, Germany.*

Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N (2010) GRASP/VND and multistart evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engrg. Appl. Artificial Intelligence* 23(5):780–794.

Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N (2011) A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Comput. Oper. Res.* 38(9):1319–1334.

Villegas JG, Prins C, Prodhon C, Medaglia AL, Velasco N (2013) A matheuristic for the truck and trailer routing problem. *Eur. J. Oper. Res.* 230(2):231–244.