A two-commodity flow formulation for the capacitated truck-and-trailer routing problem

Working Paper DPO-2017-02 (version 1, 24.03.2017)

Enrico Bartolini and Michael Schneider
{bartolini|schneider}@dpo.rwth-aachen.de
Deutsche Post Chair – Optimization of Distribution Networks
RWTH Aachen University, Germany

Abstract

In the capacitated truck-and-trailer routing problem (CTTRP), a limited fleet of capacitated trucks and trailers is available at a depot to serve a set of customers. Some of the customers cannot be reached by a truck pulling a trailer. Trucks are thus allowed to detach and park their trailer en route, then visit some customers without it, and then return back to pick up their trailer and continue their trip. In this paper, we propose a two-commodity flow formulation for the CTTRP, which uses two sets of flow variables that model the flow of goods carried by trucks pulling a trailer and by trucks alone, respectively. We describe some valid inequalities for strengthening the formulation and develop a branch-and-cut algorithm for solving it. In our computational experiments, we consider both the CTTRP and its special case in which a single capacitated truck pulling an uncapacitated trailer is available. We report results on instances derived from known benchmark sets featuring up to 30 customers. The results show that our branch-and-cut solves 30-customer instances with either a single vehicle or loose capacity constraints with very high success rate. Of the more tightly capacitated 30-customer instances, about two thirds can be solved.

Keywords: *vehicle routing, truck and trailer, two-commodity formulation, branch-and-cut*

Deutsche Post Chair - Optimization of Distribution Networks
RWTH AACHEN UNIVERSITY

1 Introduction

Truck-and-trailer routing problems (TTRPs) form a class of vehicle-routing problems (VRPs) that are characterized by the availability of non-autonomous trailers, which must be pulled by a truck. A *composite* vehicle consisting of a truck pulling a trailer permits to increase the load capacity but is subject to accessibility restrictions for composite vehicles at some of the customers, e.g., because the customers are located in a mountain area or in a city center with bans on large vehicles. To reflect such restrictions, TTRPs assume that a subset of the customers can only be served by a truck alone (so-called *truck customers*) whereas the remaining ones can be served by both a composite vehicle or by a truck alone (so-called *vehicle customers*). A number of real-world applications have been reported for TTRPs, like, e.g., milk collection (Caramia and Guerriero 2010a, Pasha et al. 2014), fuel oil delivery to private households (Drexl 2011), distribution of goods (Semet and Taillard 1993, Gerdessen 1996), postal mail delivery (Bodin and Levy 2000, Bode 2013), and container movement (Tan et al. 2006). Cuda et al. (2015) provide an up-to-date survey of TTRPs and review the solution methods proposed in the literature. TTRPs have also been discussed in surveys on location-routing problems (see, e.g., Nagy and Salhi 2007, Prodhon and Prins 2014).

In this paper, we consider the capacitated TTRP (CTTRP) introduced by Chao (2002), in which a limited number of capacitated trucks and a limited number of capacitated trailers are available at a central depot. The composite vehicles are allowed to park their trailer at any vehicle customer and to transfer load between between the truck and the trailer. From there, it is possible to perform a truck subtour serving a subset of customers, possibly inaccessible for trailers, and then return to the trailer and continue the trip. The objective is to minimize the transportation cost for serving all customers with a compatible vehicle of the given fleet while respecting the vehicle capacities.

The CTTRP is the most studied problem in the class of TTRPs, and several heuristics are available in the literature (see Chao 2002, Scheuerer 2006, Lin et al. 2009, Caramia and Guerriero 2010b, Villegas et al. 2011, 2013), of which the paper by Villegas et al. (2013) currently defines the state-of-the-art. To the best of our knowledge, no exact solution method for the CTTRP has yet been proposed in the literature, however, a few exact methods have been introduced for variants of the CTTRP and are described in the following.

Drexl (2011) study the generalized TTRP as a unified modeling framework for TTRPs with a fixed truck-trailer assignment. The problem considers additional transhipment locations at which trailers can be parked and load transfers take place, time windows at both customer and transhipment locations, and a heterogeneous fleet of vehicles with different fixed and distance-dependent costs. The authors devise an exact branch-and-price algorithm and several heuristic variants. The exact is able to consistently solve instances with up to 10 truck customer, 10 vehicle customers and 10 transshipment locations. Drexl (2014) study the VRP with trailers and transshipments, in which the the fixed truck-trailer assignment of the generalized TTRP is abandoned, i.e., a trailer can now be pulled by any compatible truck on the parts of its route, and any truck can perform a load transfer to any trailer. In addition, load dependent transfer times between trucks and trailers are considered. For this problem, the author proposes a branch-and-cut algorithm based on two compact formulations which build on a network representation of the problem involving $2 + 8n$ vertices, with n being the number of customers. Both formulations require the use of three-index variables to model the vehicle tours. The largest instances that are solved to optimality contain 8 customers, 8 transshipment locations, and 8 vehicles.

Parragh and Cordeau (2015) and Rothenbächer et al. (2016) both consider the CTTRP with time windows and develop column-generation-based solution methods based on a set-partitioning formulation. They report optimal solutions to instances with up to 100 customers if time windows are tight. Rothenbächer et al. (2016) also consider two real-world extensions: (i) load-dependent transfer times, and (ii) the option to collect double the amount at a customer every second day. The proposed method was also used to address the generalized TTRP instances of (Drexl 2011). Here, the number of istances that can be solved increases significantly. Finally, Belenguer et al. (2016) propose a branch-and-cut for the TTRP with satellite depots, in which a set of truck customers have to be served, and the trailer can be parked and load transfer can take place at a set of satellite depots. Thus, this variant is quite similar to two-echelon VRPs. The largest instance that the authors can solve features 100 customers and 10 satellite depots.

In this paper, we propose a new mathematical formulation for the CTTRP which is defined over a network with $2{+}n{+}n_c$ vertices, where n_c denotes the number of vehicle customer locations, and n is the total number of customers. We computationally assess the effectiveness of the new formulation by developing a branch-and-cut algorithm for solving it. We report results on a set of 96 instances with different characteristics featuring up to 30 customers, which are derived from the CTTRP benchmark sets of Chao (2002) and Lin et al. (2010). In the numerical experiments, we also consider a special case of the CTTRP that we call single-vehicle CTTRP, in which only one vehicle consisting of a capacitated truck pulling an uncapacitated trailer is available.

The remainder of this paper is organized as follows. In Section 2, we describe the CTTRP and introduce the main notation used in the paper. Section 3 presents the CTTRP formulation, and Section 4 introduces some classes of valid inequalities for strengthening its Linear Programming (LP) relaxation. In Section 5, we describe a branch-and-cut algorithm based on the new formulation and detail the separation procedures that it uses to detect violated inequalities. Section 6 provides a computational evaluation of our branch-and-cut algorithm, and some conclusions are drawn in Section 7.

2 Problem description and notation

The CTTRP can be defined on a complete undirected graph $G = (V_0, E)$ with vertex set $V_0 = \{0, ..., n\}$ and edge set $E = \{\{i, j\}, i \in V_0, j \in V_0, i < j\}$. The customer set $V = V_0 \setminus \{0\}$ is partitioned into the two subsets V^c and V^t: vertices in V^c correspond to n_c vehicle customers, vertices in V^t to n_t truck customers. Each customer $i \in V$ has a nonnegative demand q_i. Vertex 0 represents the depot at which a fleet of m_t trucks with capacity Q_t and m_c trailers with capacity Q_c are stationed. Trailers are non-autonomous vehicles and must be pulled by a truck. Each truck can either travel alone or pull at most one trailer, in which case it is called a composite vehicle with capacity $Q_t + Q_c$. Each edge $\{i, j\} \in E$ is associated with a cost c_{ij} which represents the cost for traveling directly from i to j (or from j to i) with a truck or with a composite vehicle. The goal of the CTTRP is to supply each customer i from the depot with a load q_i, using the available vehicles without exceeding their capacity and minimizing the transportation costs.

The trip of a vehicle starting from the depot, visiting a subset of customers, and finally returning back to the depot is called a route. The total load of the customers visited by a route cannot exceed the capacity of the corresponding vehicle (Q_t in case of a *truck route*, i.e., if the vehicle is a truck, or $Q_t + Q_c$ in case of a *vehicle route*, i.e., if the vehicle is a composite vehicle). A composite vehicle is allowed to detach its trailer during its vehicle route after visiting a vehicle customer k, then continue to visit a subset of customers (either truck customers, or vehicle customers, or both) with the truck alone, and then return back to vertex k to reattach the trailer before continuing its route. A route segment of a vehicle route which is traversed by the truck alone and starts and ends at a vehicle customer k is called *truck subtour* rooted at k. A vertex $k \in V^c$

which is the starting point of a truck subtour is called the *parking place* of that subtour. A truck subtour must correspond to a simple cycle in G, and the sum of loads of the customers on the subtour must not exceed the truck capacity Q_t.

Load transfers between truck and trailers are allowed, and thus, a vehicle route can include multiple subtours possibly rooted at the same parking place k. In principle, one could think of a truck route as a truck subtour rooted at vertex 0, however, it is necessary to distinguish between the two because the number of truck routes is limited by the number of available trucks whereas the number of truck subtours rooted at vertex 0 is not. However, the latter is restricted by the capacity of the composite vehicle whose truck is used to perform the truck subtours. This assumption seems artifical: In practice, if the depot is a parking place and no other constraints (e.g., time constraints) are considered, it stands to reason that a composite vehicle will always reload its truck upon returning to the depot. Thus, the load of the truck subtours rooted at the depot will not contribute to the load of any vehicle route. However, this is equivalent to say that the number of trucks is unlimited.

For these reasons, we assume that the depot cannot be a parking place, so that only truck routes can start from the depot, and the total number of truck routes starting from the depot is at most m_t. To simplify the exposition, we use the term "truck subtour" also to denote a truck route, and we use the respective parking place to distinguish truck routes from truck subtours, i.e., a truck route is a truck subtour rooted at 0.

3 Two-commodity flow formulation

This section describes a mathematical formulation of the CTTRP (Section 3.2), which is an extension of the two-commodity flow formulation of the CVRP (Baldacci et al. 2004). The formulation uses an extended graph (Section 3.1).

3.1 Extended graph $\mathcal{G}$

The extended graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined on an extended vertex set $\mathcal{V}$ that consists of (i) the vertex set V_0, (ii) a vertex $0'$, which is a copy of the depot 0 and represents the end vertex of all routes, and (iii) one copy i' of each vertex $i \in V^c$, which represents the end vertex of any truck subtour rooted at i. To simplify the notation, for any set $S \subseteq \mathcal{V} \setminus \{0, n+1\}$, we denote by $S_0 = S \cup \{0\}$, $S_{0'} = S \cup \{0'\}$, and by $S_{0,0'} = S \cup \{0, 0'\}$. The sum of demands of the customers in S is denoted by $q(S)$, i.e., $q(S) = \sum_{i \in S} q_i$. The edge set of the extended graph is defined as follows:

$$\mathcal{E} = E \cup \{\{i, j'\} : i \in V, j \in V_0^c, i \neq j\}.$$

Moreover, we denote by $\mathcal{E}_c$ the subset of edges that are incident with either two vehicle customers or with a vehicle customer and the depot:

$$\mathcal{E}_c = \{\{i, j\} \in E : i \in V_0^c, j \in V^c\} \cup \{\{i, 0'\} : i \in V_0^c\}.$$

With each edge $\{i, j\} \in E$ is associated the cost c_{ij}. For the other edges, we have (i) $c_{00'} = 0$, (ii) $c_{i0'} = c_{0i}$, $\forall i \in V$, and (iii) $c_{ij'} = c_{ij}$, $\forall i \in V$, $\forall j \in V^c$, $i \neq j$. Finally, we use the notation $\delta(S)$ to index the subset of edges in $\mathcal{E}$ that cross the vertex subset $S \subseteq \mathcal{V}$, and $\mathcal{E}(S, T)$ to index the subset of edges having one endpoint in S and the other in T for any two vertex subsets $S, T \subseteq \mathcal{V}$. In case S is a singleton $\{i\}$, we write $\delta(i)$ in place of $\delta(\{i\})$.

3.2 Mathematical formulation

The idea of two-commodity flow formulations is to model the flow of goods carried by a vehicle when traveling an edge $\{i, j\}$ by using two flow variables f_{ij} and f_{ji}. The first represents the load of the vehicle when traversing the edge and the latter the empty space. In our formulation, we use two distinct sets of flow variables to model the flow of goods and of empty space of the vehicle routes (called forward and backward f-flows, respectively) and those of the truck subtours (which we call forward and backward g-flows, respectively). More precisely, we use the following variables:

- Variables x_{ij}, $\forall\{i, j\} \in \mathcal{E}_c$, taking value 1 if a vehicle route traverses edge $\{i, j\} \in \mathcal{E}_c$, 0 otherwise
- Variables z_{ij}, $\forall\{i, j\} \in \mathcal{E}$, taking value 1 if a truck subtour traverses edge $\{i, j\} \in \mathcal{E}$, 0 otherwise
- Variables f_{ij} and f_{ji}, $\forall\{i, j\} \in \mathcal{E}_c$, representing the total load and empty space, respectively, on a composite vehicle traversing edge $\{i, j\} \in \mathcal{E}_c$
- Variables g_{ij} and g_{ji}, $\forall\{i, j\} \in \mathcal{E}$, representing the total load and empty space, respectively, on a truck traversing edge $\{i, j\} \in \mathcal{E}$
- Variables h_i representing the number of truck subtours rooted at $i \in V_0^c$. These variables are redundant, but we include them in the formulation to improve its readability.

An example of a feasible CTTRP solution on the extended graph $\mathcal{G}$ is shown in Figure 1. In this solution, two trucks of capacity 40 and two trailers of capacity 30 are used. In the figure, the start depot 0 and its copy $0'$ are represented as a black and a white square, respectively. Each vehicle customer i is represented as a black circle, and its copy i' as a white circle beside it. Truck customers are represented as triangles. The number in brackets on each edge displays the forward f-flow or g-flow on the edge (if non-zero), i.e., the load of the composite vehicle or the truck traversing that edge. The two numbers beside each vertex are, from left to right, the customer load and the customer number (in bold), respectively. Note that copy vertices (white circles) have no associated load.

The CTTRP can be modeled as mixed integer linear program as follows:

$$\min \sum_{\{i,j\} \in \mathcal{E}_c} c_{ij} x_{ij} + \sum_{\{i,j\} \in \mathcal{E}} c_{ij} z_{ij}$$

s.t.

$$\sum_{\{i,j\} \in \delta(i) \cap \mathcal{E}_c} (f_{ji} - f_{ij}) + \sum_{\{i,j\} \in \delta(i)} (g_{ji} - g_{ij}) = 2q_i + Q_t h_i, \;\; \forall i \in V^c \tag{1}$$

$$\sum_{j \in V^c} f_{0j} + \sum_{j \in V} g_{0j} = q(V) \tag{2}$$

$$\sum_{j \in V_{0'}^c} f_{j0} + \sum_{j \in V_{0'}} g_{j0} = m_c Q_c + m_t Q_t - q(V) \tag{3}$$

$$\sum_{j \in V_0^c} f_{0'j} + \sum_{j \in V_0} g_{0'j} = m_c Q_c + m_t Q_t \tag{4}$$

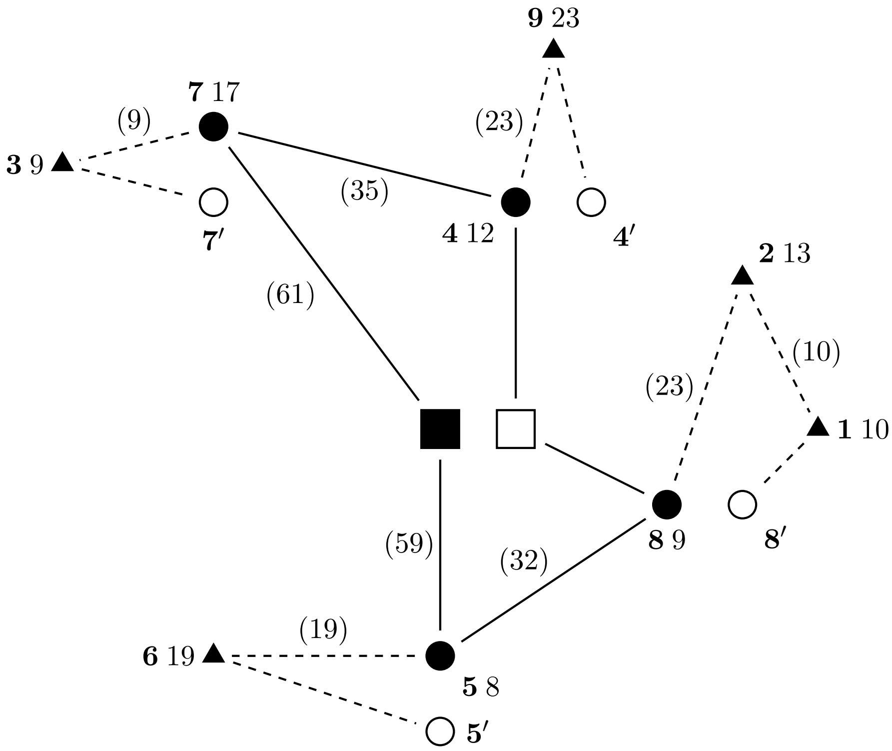

Figure 1: Example of a CTTRP solution in the extended graph.

$$f_{ij} + f_{ji} = (Q_t + Q_c)x_{ij}, \ \ \forall \{i,j\} \in \mathcal{E}_c \setminus \{0, 0'\} \tag{5}$$

$$\sum_{\{i,j\} \in \delta(i)} (g_{ji} - g_{ij}) = 2q_i, \ \ \forall i \in V^t \tag{6}$$

$$g_{ij} + g_{ji} = Q_t z_{ij}, \ \ \forall \{i,j\} \in \mathcal{E} \setminus \{0, 0'\} \tag{7}$$

$$\sum_{j \in V} g_{i'j} = h_i Q_t, \ \ \forall i \in V_0^c \tag{8}$$

$$\sum_{\{i,j\} \in \delta(i) \cap \mathcal{E}_c} x_{ij} + \sum_{\{i,j\} \in \delta(i)} z_{ij} = 2 + h_i, \ \ \forall i \in V^c \tag{9}$$

$$\sum_{\{i,j\} \in \delta(i)} z_{ij} = 2, \ \ \forall i \in V^t \tag{10}$$

$$\sum_{j \in V_{0'}^c} x_{0j} + h_0 \leq m_t \tag{11}$$

$$\sum_{j \in V_{0'}^c} x_{0j} \leq m_c \tag{12}$$

$$\sum_{\{i,j\} \in \mathcal{E}(S_k : V_{k'} \setminus S_k)} z_{ij} - h_k \geq 0, \ \ \forall S \subseteq V, \ \forall k \in V^c \tag{13}$$

$$\sum_{\{j,i'\} \in \delta(i') \setminus \{0,i'\}} z_{ji'} = h_i, \ \ \forall i \in V_0^c, \tag{14}$$

$$x_{ij} \in \{0,1\}, \ \ \forall \{i,j\} \in \mathcal{E}_c \tag{15}$$

$$f_{ij} \geq 0, f_{ji} \geq 0, \ \ \forall \{i,j\} \in \mathcal{E}_c \tag{16}$$

$$z_{ij} \in \{0,1\}, \ \ \forall \{i,j\} \in \mathcal{E} \tag{17}$$

$$g_{ij} \geq 0, g_{ji} \geq 0, \ \ \forall \{i,j\} \in \mathcal{E} \tag{18}$$

The objective function minimizes the total travel costs of the routes. Constraints (1)–(5), (7), (8) , (16), and (18) define a feasible forward and backward f-flow and g-flow . More precisely, constraints (1) define the total inflow minus the outflow at each vehicle customer vertex $i \in V^c$. If i is not a parking place, h_i must be zero, and the g-flow through i must be zero as well. Thus, in this case, constraints (1) state that the f-inflow minus the f-outflow must equal twice the demand of i. If i is a parking place of one or several truck subtours serving the customer subset S, then the f-inflow minus the f-outflow at i must equal $2q_i + 2q(S)$, i.e., twice the total load delivered to i and to its truck subtours by the composite vehicle. At the same time, the g-inflow minus the g-outflow of i must equal $h_iQ_t - 2q(S)$ because with respect to the truck subtours modeled by the g-flow, i plays the role of a depot for h_i subtours serving the customer set S. Equation (2) states that the total outflow from the start depot 0 must equal the total customer demand. Equation (3) imposes that the total inflow at the start depot 0 must equal the unused capacity of the entire vehicle fleet. Analogously, equation (4) states that the total outflow from the end depot $0'$ must equal the total capacity of the vehicle fleet. Equations (5) model the capacity constraints of the composite vehicles by imposing that the sum of forward and backward f-flows through each edge $\{i, j\}$ must either be equal to the total vehicle capacity if the edge is traversed by a composite vehicle, or be zero otherwise. Equations (6) and (7) are similar to equations (1) and (5), respectively. They model g-flow conservation and the capacity constraints of the trucks.

Equations (8) impose for each vehicle customer i that the total g-outflow from the end vertex i' of all the truck subtours rooted at i must equal the total capacity of those subtours (i.e., h_iQ_t). Together with (1) and (13), these constraints establish that each of the h_i truck subtours rooted at i has a forward g-flow from i to i' modeling the load of the truck, and a backward g-flow from i' to i modeling its empty space. Constraints (9) and (10) define the degree of each customer vertex, and constraints (11) and (12) impose an upper bound on the number of routes in a solution. The *subtour connectivity constraints* (13) forbid truck subtours to end at a different vertex than their parking place, and are described in more detail in Section 4. Finally, constraints (14) fix the value of the redundant variables h_i, and (15)–(18) define the domains of the variables.

4 Valid inequalities

In this section, we describe some classes of valid inequalities that we use to improve the LP relaxation of the formulation presented in Section 3. The separation procedures for solving the corresponding separation problem and a branch-and-cut algorithm based on the formulation strengthened by the described inequalities are detailed in Section 5.

4.1 Subtour connectivity constraints

The subtour connectivity constraints (13) are an adaptation of generalized subtour elimination constraints (see, e.g, Fischetti et al. 1995). They are used to eliminate solutions where the variables z_{ij} define a path starting from a vehicle customer i but ending at a different vertex than i'. To see that constraints (13) are valid, consider a vehicle customer $k \in V^c$. If k is the parking place of h_k truck subtours, then the variables z_{ij} must define h_k paths that all start at k, traverse a subset of vertices $S \subseteq V \setminus \{k\}$, and end at vertex k'. A necessary and sufficient condition for the existence of such paths is that for any vertex subset $S \subseteq V$ which contains k, there are at least h_k edges $\{i, j\}$ connecting S with $(V \setminus S) \cup \{k'\}$ such that $z_{ij} = 1$.

4.2 Rounded capacity constraints

Rounded capacity constraints are well-known inequalities for the capacitated VRP (CVRP, see Naddef and Rinaldi 2002). They impose a lower bound on the number of edges crossing each customer subset in any feasible solution. For any vertex subset $S \subseteq V$, let $r(S) = \left\lceil \frac{q(S)}{Q_c+Q_t} \right\rceil$, and $t(S) = \left\lceil \frac{q(S)}{Q_t} \right\rceil$ be lower bounds on the number of composite vehicles and trucks, respectively, which are needed to supply the customers in S. Then, we define the following two types of rounded capacity constraints:

Vehicle capacity cuts

$$\sum_{\{i,j\}\in\delta(S)} z_{ij} + \sum_{\{i,j\}\in\delta(S)\cap\mathcal{E}_c} x_{ij} \geq 2r(S), \qquad \forall S \subseteq V \tag{19}$$

Truck capacity cuts

$$\sum_{\{i,j\}\in\delta(S)} z_{ij} \geq 2t(S), \qquad \forall S \subseteq V^t \tag{20}$$

It is worth noting that vehicle capacity cuts (19) can be strengthened in case $r(S) = 1$ but $t(S) \geq 2$. Then, a single truck subtour is not enough to serve all customers in S. Therefore if S is not crossed by any vehicle route (i.e., if $\sum_{\{i,j\}\in\delta(S)\cap\mathcal{E}_c} x_{ij} = 0$), it must be crossed by at least two truck subtours (i.e., $\sum_{\{i,j\}\in\delta(S)} z_{ij}$ must equal at least four). Following this observation, (19) can be improved to obtain the following *lifted vehicle capacity cuts*:

$$\sum_{\{i,j\}\in\delta(S)} \frac{1}{t(S)} z_{ij} + \sum_{\{i,j\}\in\delta(S)\cap\mathcal{E}_c} x_{ij} \geq 2, \qquad \forall S \subseteq V : r(S) = 1 \text{ and } t(S) \geq 2 \tag{21}$$

To see that (21) are valid, consider any $S \subseteq V$ such that $r(S) = 1$ and $t(S) \geq 2$. In any feasible solution there are two cases: (i) at least $r(S) = 1$ vehicle routes cross S, in which case inequality (21) is satisfied, and (ii) no vehicle route crosses S, in which case at least $t(S)$ truck subtours must cross S, and thus the left hand side of (21) is at least $\frac{1}{t(S)} 2t(S) \geq 2$.

4.3 Co-circuit inequalities

Co-circuit inequalities are proposed in (Barahona and Grötschel 1986). They were adapted to the rural postman problem by Ghiani and Laporte (2000) and later to the TTRP with satellite depots by Belenguer et al. (2016). In our context, they are based on the following observation. Recall that all the vehicle routes and truck subtours are represented as paths from a starting vertex i to an ending vertex i' in the extended graph $\mathcal{G}$, and all edges $\{i,j\} \in \mathcal{E}$ can be traversed at most once in $\mathcal{G}$. For any vertex subset $S \subseteq V_0$, let S' be the vertex set obtained by adding to S all the copy vertices of the vehicle customers in S, i.e., $S' = S \cup \{i' \in \mathcal{V} : i \in S\}$, and let $\mathbb{S} = \{S' \subseteq \mathcal{V} : S \subseteq V\}$. In words, the set $\mathbb{S}$ collects all the vertex subsets of $\mathcal{V}$ such that for each vertex $i \in V_0^c$ either $i, i' \in S$, or $i, i' \notin S$.

Consider a set $S' \in \mathbb{S}$ and let $F \subseteq \delta(S')$ be a subset of edges crossing the set S'. If $|F|$ is odd, then the number of edges crossing S' in any feasible solution must be even. Indeed, any vehicle route or truck route entering S' must also leave it in order to finally reach the end depot. Moreover, any truck subtour departing from a vertex $i \in S$ must return back to i' which, by definition of S', also belongs to S'. Thus, because each edge in F can be traversed at most once, if all the $|F|$ edges in F are traversed, then at least one edge in $\delta(S') \setminus F$ must also be traversed because $|F|$ is odd. Therefore, the following inequalities, which we call *truck co-circuit inequalities*, are valid for the CTTRP

$$\sum_{\{i,j\}\in\delta(S')\setminus F} z_{ij} \geq \sum_{\{i,j\}\in F} z_{ij} - |F| + 1, \quad \forall S' \in \mathbb{S}, \quad \forall F \subseteq \delta(S'),\ |F|\ odd. \tag{22}$$

Similarly, let $\mathbb{S}^c = \{S' \subseteq \mathcal{V} : S \subseteq V_0^c\}$. Then, the following *vehicle co-circuit inequalities* are valid

$$\sum_{\{i,j\}\in(\delta(S')\cap\mathcal{E}_c)\setminus F} x_{ij} \geq \sum_{\{i,j\}\in F} x_{ij} - |F| + 1, \quad \forall S' \in \mathbb{S}^c, \quad \forall F \subseteq \delta(S') \cap \mathcal{E}_c,\ |F|\ odd. \tag{23}$$

5 Branch-and-cut algorithm and separation procedures

To assess the practical usefulness of the formulation presented in Section 3, we have implemented a branch-and-cut algorithm for solving it. We first solve the LP relaxation of formulation (1)–(18) strengthened by the inequalities described in Section 4 by means of a cutting plane algorithm. At this stage, the inequalities are separated in the order (13), (20), (19), (22), and (23). At most 20 violated cuts are added in each iteration. After each call to the separation algorithm, the problem is re-optimized.

The formulation resulting from the addition of the cuts found by the cutting plane algorithm is then solved by a branch-and-cut using the mixed integer linear programming solver IBM ILOG Cplex with default parameters settings. During preliminary experiments, we found that it is convenient to disable the separation of inequalities (23), (22), (19), (20), and (21) during the branch-and-cut phase. Therefore, only inequalities (13) (in addition to Cplex cuts) are separated by the branch-and-cut. Separation routines for detecting violated inequalities (13) are provided through the callback interface. In the remainder of this section, we describe how violated inequalities are separated by our algorithm.

Co-circuit inequalities We separate co-circuit inequalities (23) and (22) by means of a procedure similar to that used by Belenguer et al. (2016), which is based on an algorithm of Letchford et al. (2008) for separating blossom inequalities. We only describe the procedure for the separation of inequalities (22) because the method for separating inequalities (23) is very similar. The inequalities are separated in the following equivalent form:

$$\sum_{\{i,j\}\in\delta(S')\setminus F} z_{ij} + \sum_{\{i,j\}\in F} (1 - z_{ij}) \geq 1, \quad \forall S' \in \mathbb{S}, \quad \forall F \subseteq \delta(S'),\ |F|\ odd \tag{24}$$

The separation procedure uses the following observation. Once given a set $S' \in \mathbb{S}$, a set $F_{S'} \subseteq \delta(S')$ which minimizes the left hand side (lhs) of (24) with respect to a vector $\mathbf{z} \in \{0,1\}^{|\mathcal{E}|}$ is obtained simply by taking all edges $\{i,j\} \in \delta(S')$ with $z_{ij} > 0.5$. At this point, if $|F_{S'}|$ is even, then it suffices to either add or remove from $F_{S'}$ the edge $\{i,j\}$ yielding the smallest increase in the lhs of (24) to obtain the most violated constraint for the given S'. Following this observation the separation happens into two stages. First the min-cut tree of $\mathcal{G}$ (Gomory and Hu 1961) with respect to edge weights $u_{ij} = \min\{z_{ij}, 1 - z_{ij}\}$ is constructed. For each cut and corresponding vertex set S' obtained in this way, the weight of the cut equals the smallest lhs of an inequality (24) defined by S' and the corresponding $F_{S'}$ (where $|F_{S'}|$ is not necessarily odd). Second, the observation above is used to construct F with $|F|$ odd given $F_{S'}$ and to obtain a valid inequality defined by S and F. In our case, however, we have to impose the additional restriction $S' \in \mathbb{S}$. To do this, recall that each $S' \in \mathbb{S}$ is obtained from a set $S \subseteq V_0$ by adding to it the copy vertices of all vehicle customers (and depot) in S. Thus we first compute the reduced graph $G = (V_0, E)$ obtained from $\mathcal{G}$ by replacing each vertex pair i, i', such that $i \in V_0^c$ with the single vertex i, and by removing any edge $\{i, j'\}$ such that $j \in V_0^c$. The weight of each edge $\{i, j\} \in E$ is then set as follows:

$u_{ij} = \min\{(1 - z_{ij}) + (1 - z_{ji'}) + (1 - z_{ij'}), z_{ij} + (1 - z_{ji'}) + (1 - z_{ij'}), z_{ij} + z_{ji'} + (1 - z_{ij'}), z_{ij} + (1 - z_{ji'}) + z_{ij'}, (1 - z_{ij}) + z_{ji'} + z_{ij'}, z_{ij} + z_{ji'} + z_{ij'}\}$ if $i, j \in V_0^c$,

$u_{ij} = \min\{(1 - z_{ij}) + (1 - z_{ij'}), z_{ij} + z_{ij'}, (1 - z_{ij}) + z_{ij'}, z_{ij} + (1 - z_{ij'})\}$ if $i \in V^t$, and $j \in V_0^c$,

$u_{ij} = \min\{(1 - z_{ij}) + (1 - z_{ji'}), z_{ij} + z_{ji'}, (1 - z_{ij}) + z_{ji'}, z_{ij} + (1 - z_{ji'})\}$ if $j \in V^t$, and $i \in V_0^c$.

Indeed, consider for example an edge $\{i, j\} \in E$ with $i, j \in V_0^c$. This edge corresponds to the three edges $\{i, j\}$, $\{i, j'\}$, $\{j, i'\}$ of $\mathcal{G}$ which cross a set $S' \in \mathbb{S}$ if and only if $\{i, j\}$ crosses S. Thus, the contribution of the three variables z_{ij}, $z_{ij'}$ and $z_{ji'}$ to the lhs of an inequality (24) defined by a pair $(S', F_{S'})$ is either u_{ij} as defined above if $\{i, j\}$ crosses S, or zero otherwise. A similar reasoning applies to edges $\{i, j\}$ with either $i \in V_0^c$ or $j \in V_0^c$. Thus, we compute a min-cut tree with respect to weights u_{ij} in $\overline{\mathcal{G}}$, and for each resulting cut with weight less than 1 defined by a set $S \subseteq V_0$, we construct S' and $|F_{S'}|$, derive a corresponding valid inequality (24), and check if it is violated.

Subtour connectivity constraints Inequalities (13) can be separated in polynomial time as follows. Given a solution $(\mathbf{x}, \mathbf{z}, \mathbf{h})$, construct the support graph $G_{\mathbf{z}} = (\mathcal{V}_{\mathbf{z}}, \mathcal{E}_{\mathbf{z}})$ induced by $\mathbf{z}$, where $\mathcal{V}_{\mathbf{z}} \subseteq \mathcal{V} \setminus \{0, 0'\}$, and the edge set $\mathcal{E}_{\mathbf{z}}$ contains all edges $\{i, j\} \in \mathcal{E}$, $i, j \neq 0, 0'$, with $z_{ij} > 0$. Associate a capacity $w_{ij} = z_{ij}$ with each edge $\{i, j\} \in \mathcal{E}_{\mathbf{z}}$. Note that no inequality (13) defined by a vehicle customer k with $h_k = 0$ can be be violated. Consider a vertex $k \in V^c$ such that $h_k > 0$ and let $G_{\mathbf{z}}^k = (\mathcal{V}_{\mathbf{z}}^k, \mathcal{E}_{\mathbf{z}}^k)$ be the connected component of $G_{\mathbf{z}}$ which contains vertex k.

An inequality (13) defined by $k \in V^c$ is violated if and only if the maximum flow from k to k' in $G_{\mathbf{z}}^k$ is less than h_k. Indeed, if that is the case, then the minimum s, t-cut $(S, \bar{S})$ in $G_{\mathbf{z}}^k$ with $s = k$ and $t = k'$ has a weight $\sum_{\{i,j\} \in \mathcal{E}(S, \bar{S})} z_{ij}$ less than h_k, and thus the inequality (13) defined by S and k is violated. Obviously, if $h_k > 0$ but $k' \notin \mathcal{V}_{\mathbf{z}}^k$, then there is no need to compute a minimum cut as k and $S = \mathcal{V}_{\mathbf{z}}^k$ clearly define a violated inequality (13).

Rounded capacity constraints The separation of inequalities (20) is equivalent to the separation of classical rounded capacity constraints for a CVRP instance, in which the vehicles have a capacity of Q_t. Given a solution $(\mathbf{x}, \mathbf{z}, \mathbf{h})$, we work with a reduced graph $\overline{G}$ having vertex set V_0^t and only containing edges $\{i, j\} \in E$ with $i, j \in V_0^t$. The edges of $\overline{G}$ are associated with weights x_{ij}^* computed as $x_{ij}^* = z_{ij}$ if $i, j \in V^t$, and $x_{0i}^* = z_{0i} + \sum_{j \in V_0^c} z_{ij'}$ for all $i \in V^t$. We then use the separation package CVRPSEP (Lysgaard 2004) on the support graph induced on $\overline{G}$ by $\mathbf{x}^*$. Inequalities (19) are not separated in the same way as (20) because they are defined with respect to both $\mathbf{x}$ and $\mathbf{z}$, and thus, the degree of a vertex in the resulting support graph can be greater than two. Instead, we use the tabu search of Augerat et al. (1998), which starts with a vertex set S containing a single vertex $i \in V$ and iteratively adds or removes vertices to/from S to obtain a new set S'. For each set S, the algorithm computes the slack $\sigma(S) = 2r(S) - \sum_{\{i,j\} \in \delta(S)} z_{ij} - \sum_{\{i,j\} \in \delta(S) \cap \mathcal{E}_c} x_{ij}$ of the corresponding inequality (19) (or the slack of the corresponding lifted inequality (21) if $r(S) = 1$ and $t(S) \geq 2$) and returns the set S yielding the largest value $\sigma(S)$.

6 Computational experiments

This section describes the computational evaluation of the branch-and-cut algorithm described above. The algorithm was implemented in C and compiled with Visual Studio 2012 64-bit. All computational experiments were run on an Intel Core i7-3770 (CPU @ 3.40 GHz) with 16 GB of RAM. Cplex 12.7 was used as the MILP solver. The description of the benchmark instances is given in Section 6.1, the results on the instances are discussed in Section 6.2.

6.1 Description of the test instances

We derive four sets of instances by using two CTTRP benchmarks from the literature: (i) the *Chao* set containing 21 instances with up to 150 customers (Chao 2002), and (ii) the *Tai* set containing 36 instances with up to 150 customers (Lin et al. 2010). Both sets have originally been created from benchmark sets for the CVRP (Christofides et al. 1979, Rochat and Taillard 1995). For each CVRP instance, three CTTRP instances were obtained by designating 25%, 50% and 75% of the customers as truck customers and the remaining ones as vehicle customers. Instances of set *Chao* are characterized by uniformly distributed customers whereas in instances of set *Tai* customers are clustered.

Due to their size, the original *Chao* and *Tai* instances are not suitable for our branch-and-cut algorithm. Therefore, we create two new sets (called *Chao30* and *Tai30*) by extracting 30 customers from the original instances. More precisely, we use instances 1–12 in *Chao* and instances 1–12 in *Tai* as base instances. To generate the first instance, we take the first $n_t = \lceil 0.25 \cdot 30 \rceil$ truck customers and the first $30 - n_t$ vehicle customers from *Chao* (*Tai*) instance 1. In analogous fashion, we take the first $n_t = \lceil 0.5 \cdot 30 \rceil$ to generate the second instance from *Chao* (*Tai*) instance 2, and $n_t = \lceil 0.75 \cdot 30 \rceil$ to generate the third instance from *Chao* (*Tai*) instance 3. This is repeated with instances 4–12 of the original *Chao* and *Tai* set.

Both sets *Chao30* and *Tai30* consist of three subsets called A, B, and C. Instances of type A are obtained by setting the number of available vehicles and the vehicle capacities so as to keep a similar ratio between total customer demand $q(V)$ and total capacity $m_tQ_t + m_cQ_c$ as in the original *Chao* and *Tai* instances. Instances of type B are derived from the type A instances by increasing the capacity of the vehicles. Instances of type C are single-vehicle CTTRP instances, in which a single truck of capacity Q_t together with a trailer of capacity $Q_c = q(V)$ is available.

Finally, we create two additional sets of smaller instances called *Chao25* and *Tai25* by extracting the first 25 customers from the type A instances of sets *Chao30* and *Tai30*. The generated instances are available at the URL: https://www.dropbox.com/s/hxzuge7l62bye5r/CTTRP_DATA.zip?dl=0. In our computational experiments, the cost c_{ij} of each edge $\{i, j\}$ is set equal to $\frac{\lfloor 10000 Euc(i,j) \rfloor}{10000}$ where $Euc(i, j)$ denotes the Euclidean distance between i and j.

6.2 Computational results

Tables 1–4 report the results obtained by the branch-and-cut algorithm on the new instances within a time limit of 2 hours per instance. The first eight columns of these tables summarize the characteristics of the instances. They report an instance identifier (column Inst.), the number of vehicle customers (column n_c), the number of available trailers (column m_c) and their capacity (column Q_c), the number of available trucks (column m_t) and their capacity (columns Q_t), and the percentage ratio of customer demand and available vehicle capacity $\%q = 100 \cdot q(V)/(m_cQ_c + m_tQ_t)$ (column $\%q$). In the tables reporting results for instances of type C, the columns m_c, m_t and Q_c are not reported, and column $\%q$ is replaced by column q_t reporting the total demand of the truck customers.

In all tables, column ub reports the best upper bound found by the branch-and-cut algorithm. Columns $\%lb_0$, $\%lb_+$, $\%lb_c$, and $\%lb_*$ report the percentage ratio of the initial lower bound lb_0 (i.e., the optimal cost of the LP relaxation of the formulation without additional cuts), the percentage ratio of the lower bound lb_+ obtained after adding the cuts described in Section 4, the percentage ratio of the lower bound lb_c obtained by Cplex at the root node, and the percentage ratio of the final lower bound lb_* obtained by Cplex at termination. The percentage ratio of a lower bound lb is computed as $100 \cdot lb/ub$. Columns scc, rcc, and cci report the total number of subtour connectivity constraints (13), rounded capacity constraints (19, 20, 21), and co-circuit inequalities (22, 23) separated by the branch-and-cut algorithm. Finally, column t reports the total computing time (in seconds).

Tables 1–4 show that our branch-and-cut algorithm can solve all but one instance with 25 customers, and most of those with 30 customers of type B (21 out of 24). Instances of type A seem more difficult, and seven out of 24 are not solved within the time limit. This is probably due to the tighter capacity constraints. Finally, all the single-vehicle CTTRP instances, i.e., those of type C, are solved to optimality within about one hour of CPU time (and all but one even within 30 minutes). Overall, the algorithm solves 85 out of the 96 instances that we investigated.

The tables show that the lower bound lb_0 provided by the LP relaxation of our formulation can be rather weak. Considering only the instances with 25 customers, the percentage ratio of lb_0 can already be as low as $\sim 72\%$, and on the instances of type A (for all of which an optimal solution is known), the percentage ratio of lb_0 is on average 85.15%. However, the addition of rounded capacity constraints and co-circuit inequalities improves lb_0 significantly, especially on instances of type B and C. The addition of Cplex cuts improves further the lower bound at the root node. More precisely, considering the instances with 30 customers, the overall improvement in the root lower bound for instances *Chao30* (*Tai30*) is on average about 4.6% (13%) for type A instances, 5.8% (13.2%) for type B instances, and 6.3% (30.7%) for type C instances. It is interesting to note that for instances of sets *Tai25* and *Tai30*, the valid inequalities appear significantly more effective. This is possibly due to the clustered structure of these instances, which may make the rounded capacity constraints more effective. On the other hand, lb_0 is weaker for these instances and they also appear more difficult for the branch-and-cut. Indeed, the only unsolved instance with 25 customers, as well as eight out of the ten unsolved instances with 30 customers all belong to the sets *Tai25* and *Tai30*. Finally, recall that the instances of type C involve a single vehicle and no capacity constraint for the composite vehicle. These instances appear easier for the branch-and-cut, likely due to the structure of the single vehicle problem which is less constrained than the CTTRP.

7 Conclusions

We introduce a new mathematical formulation for the capacitated truck-and-trailer routing problem (CTTRP), which is an extension of the two-commodity flow formulation for the capacitated vehicle-routing problem, and we describe some valid inequalities for strengthening it. We develop a branch-and-cut algorithm based on the new formulation and evaluate it computationally on a set of CTTRP instances with up to 30 customers and diverse characteristics. Our computational results suggest that the branch-and-cut algorithm can solve instances with up to 30 customers and is particularly effective for instances with loose capacity constraints, or featuring a single composite vehicle with an uncapacitated trailer.

Table 1: Results on the instances of set *Chao25*

Inst.	n_c	m_c	Q_c	m_t	Q_t	%q	ub	%lb_0	%lb_+	%lb_c	%lb_*	scc	rcc	cci	t
1	19	2	100	3	100	84.40	340.6	84.27	89.57	90.30	100.0	6	22	87	35
2	13	2	100	3	100	84.80	366.3	83.70	87.21	89.87	100.0	28	8	44	26
3	6	2	100	3	100	84.20	373.2	84.18	85.48	92.36	100.0	77	13	23	33
4	19	2	100	4	100	82.17	355.6	85.43	89.80	90.07	100.0	18	66	68	163
5	12	2	100	4	100	86.17	378.1	85.49	87.84	90.83	100.0	13	24	40	18
6	6	2	100	4	100	80.83	401.0	84.31	85.69	90.97	100.0	28	5	21	31
7	18	2	100	2	150	68.20	324.6	87.31	93.00	93.07	100.0	2	26	46	15
8	12	2	150	1	150	76.67	353.6	85.45	88.92	91.22	100.0	34	20	37	26
9	7	2	100	2	150	69.00	364.2	86.48	88.12	94.14	100.0	16	9	39	6
10	20	2	100	3	100	66.40	374.0	84.73	88.79	89.86	100.0	4	39	93	58
11	12	2	100	3	100	66.40	389.7	85.49	86.72	90.10	100.0	67	5	39	68
12	6	2	100	2	100	66.40	401.3	84.92	85.85	90.88	100.0	86	4	35	51
Average								85.15	88.08	91.14	100.0				

Table 2: Results on the instances of set *Tai25*

Inst.	n_c	m_c	Q_c	m_t	Q_t	%q	ub	%lb_0	%lb_+	%lb_c	%lb_*	scc	rcc	cci	t
1	19	3	750	6	750	64.28	497.2	73.46	94.59	95.65	100.0	0	120	71	8
2	13	3	750	5	750	81.25	672.0	80.07	93.41	95.12	100.0	0	111	39	6
3	6	3	750	5	750	65.08	691.6	81.15	91.43	92.98	100.0	20	62	32	10
4	19	4	850	5	850	84.77	444.4	90.19	94.59	95.25	100.0	6	45	41	339
5	13	4	850	6	850	88.91	512.1	72.64	96.11	96.57	100.0	3	68	47	18
6	6	3	850	7	850	94.24	689.1	66.99	75.40	77.56	100.0	166	44	37	281
7	19	2	600	5	600	63.45	406.1	77.05	84.00	84.60	100.0	3	65	98	1197
8	13	2	600	5	600	64.05	413.4	78.47	84.57	85.98	100.0	82	32	56	49
9	5	2	600	5	600	63.98	402.3	83.32	87.44	90.28	100.0	44	21	10	20
10	19	3	850	5	850	75.34	706.9	83.98	94.28	97.80	100.0	0	91	66	7
11	13	2	850	5	850	80.00	738.9	73.78	90.37	91.54	100.0	6	59	46	117
12	5	3	850	5	850	61.32	820.3	71.05	83.67	87.93	94.43	276	51	28	tl
Average								77.68	89.16	90.94	99.54				

Table 3: Results on the instances of set *Chao30*

Chao30: Type A instances															
Inst.	n_c	m_c	Q_c	m_t	Q_t	$\%q$	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	2	100	3	100	97.00	386.9	84.20	88.89	89.37	100.0	34	34	91	1128
2	15	2	100	3	100	96.60	408.8	82.36	85.05	86.30	100.0	171	29	53	1219
3	7	2	100	3	100	96.20	421.8	84.33	85.88	88.78	100.0	27	10	44	46
4	23	2	100	4	100	95.67	402.9	86.95	89.43	91.10	100.0	7	41	77	253
5	15	2	100	4	100	98.50	460.2	81.30	83.30	85.20	97.44	115	25	48	tl
6	8	2	100	4	100	99.33	512.1	78.00	79.65	81.14	94.39	128	24	25	tl
7	22	2	100	2	150	86.40	385.4	87.56	90.30	91.46	100.0	6	39	84	48
8	15	1	150	2	150	92.67	389.6	87.35	89.71	90.82	100.0	13	21	64	14
9	8	2	100	2	150	87.00	444.1	84.84	87.15	94.05	100.0	60	9	52	7
10	22	2	100	3	100	81.00	432.5	82.43	86.87	87.33	100.0	146	38	73	1097
11	15	2	100	3	100	82.20	410.4	85.11	87.95	89.62	100.0	38	18	37	200
12	8	2	100	3	100	80.00	451.2	86.65	87.36	89.28	100.0	31	6	19	449
Average								85.18	87.86	89.81	99.32				

Chao30: Type B instances															
Inst.	n_c	m_c	Q_c	m_t	Q_t	$\%q$	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	2	150	2	150	80.83	374.5	82.74	86.48	88.30	100.0	165	25	91	524
2	15	2	150	2	150	80.50	379.9	83.70	85.95	88.41	100.0	237	11	65	581
3	7	2	150	2	150	80.17	387.1	84.72	86.05	93.32	100.0	4	7	28	5
4	23	2	150	2	200	82.00	356.8	88.84	91.47	92.01	100.0	15	28	72	55
5	15	2	150	2	200	84.43	395.0	84.58	86.74	90.95	100.0	17	17	63	30
6	8	2	150	2	200	85.14	417.0	85.86	86.43	96.14	100.0	38	8	27	8
7	22	2	150	2	200	61.71	372.5	87.70	93.25	93.49	100.0	89	36	80	223
8	15	2	150	2	200	59.57	374.7	89.07	91.90	93.67	100.0	3	21	94	21
9	8	2	150	2	200	62.14	387.2	88.33	91.69	98.28	100.0	15	22	33	6
10	22	2	150	2	200	57.86	377.4	89.59	94.43	94.81	100.0	23	27	49	25
11	15	2	150	2	200	58.71	350.7	93.73	96.82	96.90	100.0	4	30	69	3
12	8	2	150	2	200	57.14	373.1	96.77	98.27	98.79	100.0	6	19	16	4
Average								87.97	90.79	93.76	100.0				

Chao30: Type C instances												
Inst.	n_c	Q_t	q_t	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	150	93	356.3	82.96	88.58	88.96	100.0	97	24	121	786
2	15	150	224	371.6	80.80	84.91	85.59	100.0	212	23	72	670
3	7	150	341	386.5	79.71	84.43	86.33	100.0	17	15	29	102
4	23	150	113	351.7	87.66	90.32	90.84	100.0	7	11	59	321
5	15	150	276	383.8	81.96	88.57	89.89	100.0	31	21	52	131
6	8	150	422	435.8	75.19	77.35	83.02	100.0	37	11	20	196
7	22	200	128	365.1	85.78	91.31	91.41	100.0	34	36	119	214
8	15	100	217	419.5	77.11	84.97	85.48	100.0	8	20	75	106
9	8	200	326	408.0	82.00	86.63	91.81	100.0	19	13	36	16
10	22	100	93	411.5	80.72	85.18	85.46	100.0	109	19	117	1394
11	15	100	187	398.8	81.88	85.37	86.14	100.0	49	17	76	112
12	8	100	317	471.8	77.17	79.19	83.23	100.0	24	11	48	258
Average					81.08	85.57	87.34	100.0				

Table 4: Results on the instances of set *Tai30*

Tai30: Type A instances																
Inst.	n_c	m_c	Q_c	m_t	Q_t	$\%q$	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	3	750	6	750	90.16	743.3	75.71	89.25	91.02	97.03	0	24	87	tl
2	15	3	750	5	750	87.30	830.6	74.80	95.27	95.63	100.0	0	23	41	19
3	8	3	750	5	750	87.65	875.2	73.64	92.84	94.11	100.0	3	15	60	22
4	23	4	850	5	850	92.82	536.4	83.41	89.49	90.89	100.0	27	11	28	5467
5	15	4	850	6	850	91.86	526.0	75.78	95.71	97.38	100.0	10	21	72	72
6	8	3	850	8	850	93.93	806.7	68.53	71.71	80.65	100.0	36	11	42	6985
7	23	3	600	5	600	89.58	678.4	79.43	84.95	85.87	93.40	3	36	70	tl
8	15	3	600	5	600	89.58	699.1	81.25	85.52	86.38	97.60	19	20	68	tl
9	7	3	600	7	600	71.67	699.1	82.53	85.47	86.47	96.72	153	13	28	tl
10	23	3	850	5	850	91.66	743.1	88.85	97.85	99.39	100.0	1	19	105	70
11	15	2	850	5	850	92.32	788.3	77.79	91.42	94.33	100.0	9	17	64	498
12	7	3	850	5	850	91.32	939.4	73.98	84.03	89.72	94.67	212	11	69	tl
Average								78.83	90.16	91.93	98.13				

Tai30: Type B instances															
Inst.	n_c	m_c	Q_c	m_t	Q_t	$\%q$	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	3	1000	6	1000	67.62	566.5	85.49	98.40	98.77	100.0	10	85	81	4
2	15	3	1000	5	1000	65.48	638.9	82.96	91.94	93.39	100.0	10	72	51	11
3	8	3	1000	5	1000	65.74	655.5	82.70	89.94	91.51	100.0	0	27	53	6
4	23	4	1000	5	1000	78.90	467.2	84.60	89.72	91.36	100.0	3	62	45	1758
5	15	4	1000	6	1000	78.08	511.2	69.48	96.93	97.42	100.0	16	93	57	263
6	8	3	1000	8	1000	79.84	679.2	70.75	77.89	81.64	96.83	394	50	59	tl
7	23	3	900	5	900	59.72	544.0	81.20	94.98	95.29	100.0	0	67	124	77
8	15	3	900	5	900	59.72	568.2	82.45	92.74	93.61	100.0	63	32	51	23
9	7	3	900	7	900	47.78	568.7	83.56	92.93	94.64	100.0	2	17	26	13
10	23	3	1000	5	1000	77.91	729.2	82.16	94.45	95.80	100.0	0	61	58	4133
11	15	2	1000	5	1000	78.47	761.0	69.96	85.53	86.80	97.13	19	39	67	tl
12	7	3	1000	5	1000	77.63	815.1	75.02	86.75	89.60	99.14	32	68	52	tl
Average								80.26	92.21	93.47	99.66				

Tai30: Type C instances												
Inst.	n_c	Q_t	q_t	ub	$\%lb_0$	$\%lb_+$	$\%lb_c$	$\%lb_*$	scc	rcc	cci	t
1	23	750	1549	508.9	71.89	97.53	98.48	100.0	0	47	72	2
2	15	750	2288	711.3	65.02	94.04	95.48	100.0	6	39	48	5
3	8	750	3036	757.4	62.29	89.86	91.08	100.0	14	34	47	4
4	23	850	1867	310.6	59.52	71.70	73.14	100.0	10	45	55	3716
5	15	850	2478	333.9	61.87	95.82	97.84	100.0	0	88	24	1
6	8	850	4779	359.1	66.11	97.75	99.36	100.0	0	61	15	0
7	23	600	262	491.1	62.98	97.28	97.76	100.0	0	51	58	4
8	15	600	916	538.5	60.85	90.18	93.14	100.0	10	33	63	4
9	7	600	1358	585.4	57.58	88.82	90.99	100.0	80	36	22	15
10	23	850	1369	608.8	66.01	88.85	98.11	100.0	2	46	55	2
11	15	850	2986	667.6	62.69	87.51	97.14	100.0	13	41	49	29
12	7	850	4953	719.1	64.32	90.25	97.02	100.0	9	81	69	4
Average					63.43	90.80	94.13	100.0				

References

P. Augerat, J. M. Belenguer, E. Benavent, A. Corberan, and D. Naddef. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, 106:546–557, 1998.

R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52:723–738, 2004.

F. Barahona and M. Grötschel. On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory*, 40:40–62, 1986.

J. Belenguer, E. Benavent, A. Martínez, C. Prins, C. Prodhon, and J. Villegas. A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. *Transportation Science*, 50(2):735–749, 2016.

C. Bode. Lower bounds for park and loop delivery problems. Technical Report LM-2013-02, Chair of Logistics Management, Johannes Gutenberg University Mainz, Mainz, Germany, 2013. Available online at `http://logistik.bwl.uni-mainz.de/158.php`.

L. Bodin and L. Levy. Scheduling of local delivery carrier routes for the united states postal service. In M. Dror, editor, *Arc Routing: Theory, Solutions, and Applications*, pages 419–442. Kluwer, Boston, 2000.

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

M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61(7):1168–1180, 2010b.

I. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29(1): 33–51, 2002.

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial optimization*. Chichester, UK: Wiley, 1979.

R. Cuda, G. Guastaroba, and M. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015.

M. Drexl. Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem. *Journal of Quantitative Methods for Economics and Business Administration*, 12:5–38, 2011.

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

M. Fischetti, J. J. Salazar González, and P. Toth. The symmetric generalized traveling salesman polytope. *Networks*, 26(113–123), 1995.

J. Gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93(1):135–147, 1996.

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

R. E. Gomory and T. C. Hu. Multi-terminal network flows. *J. Soc. Indust. Appl. Math.*, 9:551–556, 1961.

A. N. Letchford, G. Reinelt, and D. O. Theis. Odd minimum cut sets and b-matchings revisited. *SIAM Journal on Discrete Mathematics*, 22:1480–1487, 2008.

S.-W. Lin, V. F. Yu, and S.-Y. Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5):1683–1692, 2009.

S. W. Lin, V. F. Yu, and S. Y. Chou. A note on the truck and trailer routing problem. *Expert Systems with Applications*, 37(899–903), 2010.

J. Lysgaard. A CVRPSEP package, 2004. URL `http://www.hha.dk/~lys/CVRPSEP.htm`.

D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, volume 9. SIAM Monographs on Discrete Mathematics and Applications, 2002.

G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177 (2):649 – 672, 2007.

S. Parragh and J.-F. Cordeau. Branch-and-price for the truck and trailer routing problem with time windows. Technical report, CIRRELT, Université de Montréal, Canada, 2015.

U. Pasha, A. Hoff, and A. Løkketangen. A hybrid approach for milk collection using trucks and trailers. *Annals of Management Science*, 2014.

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

Y. Rochat and E. D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

A.-K. Rothenbächer, M. Drexl, and S. Irnich. Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. Technical Report LM-2016-06, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany, 2016.

S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33 (4):894–909, 2006.

F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41(4):469–488, 1993.

K. Tan, Y. Chew, and L. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172(3):855–885, 2006.

J. Villegas, C. Prins, C. Prodhon, A. Medaglia, and N. Velasco. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9):1319–1334, 2011.

J. Villegas, C. Prins, C. Prodhon, A. Medaglia, and N. Velasco. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2):231–244, 2013.