Drexl, Michael

Article

Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem

Revista de Métodos Cuantitativos para la Economía y la Empresa

Provided in Cooperation with:
Universidad Pablo de Olavide, Sevilla

Suggested Citation: Drexl, Michael (2011) : Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem, Revista de Métodos Cuantitativos para la Economía y la Empresa, ISSN 1886-516X, Universidad Pablo de Olavide, Sevilla, Vol. 12, pp. 5-38

This Version is available at:
https://hdl.handle.net/10419/59097

REVISTA DE MÉTODOS CUANTITATIVOS PARA LA ECONOMÍA Y LA EMPRESA (12). Páginas 5–38.
Diciembre de 2011. ISSN: 1886-516X. D.L: SE-2927-06.
URL: http://www.upo.es/RevMetCuant/art.php?id=51

Branch-and-Price and Heuristic Column Generation for the Generalized Truck-and-Trailer Routing Problem

Drexl, Michael
Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz
Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg
Correo electrónico: drexl@uni-mainz.de

ABSTRACT

The generalized truck-and-trailer routing problem (GTTRP) constitutes a unified model for vehicle routing problems with trailers and a fixed lorry-trailer assignment. The GTTRP is a generalization of the truck-and-trailer routing problem (TTRP), which itself is an extension of the well-known vehicle routing problem (VRP). In the GTTRP, the vehicle fleet consists of single lorries and lorry-trailer combinations. Some customers may be visited only by a single lorry or by a lorry without its trailer, some may also be visited by a lorry-trailer combination. In addition to the customer locations, there is another relevant type of location, called transshipment location, where trailers can be parked and where a load transfer from a lorry to its trailer can be performed.

In this paper, two mixed-integer programming (MIP) formulations for the GTTRP are presented. Moreover, an exact solution procedure for the problem, a branch-and-price algorithm, and heuristic variants of this algorithm are described. Computational experiments with the algorithms are presented and discussed. The experiments are performed on randomly generated instances structured to resemble real-world situations and on TTRP benchmark instances from the literature. The results of the experiments show that instances of realistic structure and size can be solved in short time and with high solution quality by a heuristic algorithm based on column generation.

Keywords: vehicle routing; transshipment; fleet planning; elementary shortest path problem with resource constraints.
JEL classification: C610.
MSC2010: 90B06; 90C11; 90C90.

Artículo recibido el 22 de febrero de 2011 y aceptado el 8 de julio de 2011.

Branch-and-Price y generación heurística de columnas para el problema generalizado de rutas de trenes de carretera

RESUMEN

El problema generalizado de rutas de trenes de carretera (*generalized truck-and-trailer routing problem*, GTTRP) constituye un modelo unificado para problemas de rutas de vehículos con remolques y asignación fija camión-remolque. El GTTRP es una generalización del *truck-and-trailer routing problem* (TTRP), que es una extensión del conocido problema de rutas de vehículos (*vehicle routing problem*, VRP). En el GTTRP, la flota de vehículos consiste en camiones sin remolque (camiones solos) y trenes de carretera. Algunos clientes pueden ser visitados exclusivamente por un camión solo o un camión sin su remolque, otros pueden ser visitados también por un tren de carretera. Además de las ubicaciones de los clientes hay otro tipo de localización llamada ubicación de trasbordo. Allí los remolques pueden ser aparcados, y es posible efectuar un trasbordo de carga desde un camión a su remolque.

En este trabajo se presentan dos modelos de programación lineal entero mixto (MIP). Además, se describen un algoritmo exacto branch-and-price y variantes heurísticas de este algoritmo. Se presentan y analizan estudios computacionales con los algoritmos. Se usan problemas generados aleatoriamente, diseñados para semejar situaciones reales, y problemas TTRP de la literatura. Los resultados muestran que, utilizando un algoritmo heurístico basado en generación de columnas, se pueden resolver problemas de estructura y tamaño real en poco tiempo y con solución de alta calidad.

Palabras clave: rutas de vehículos; trasbordo; planificación de flotas; problema del camino más corto con limitaciones de recursos.
Clasificación JEL: C610.
MSC2010: 90B06; 90C11; 90C90.

1 Introduction

The generalized truck-and-trailer routing problem (GTTRP) is a generalization of the truck-and-trailer routing problem (TTRP) (a term introduced by [8]), which itself is an extension of the well-known vehicle routing problem (VRP) ([10], [43], [19]). The GTTRP constitutes a unified model for vehicle routing problems with trailers and a fixed lorry-trailer assignment.

The GTTRP can be described as follows. There is a set of *customers* with a known, *deterministic supply of a single good*, and a set of *transshipment locations* used for *parking* and/or *load transfer*. Visiting a transshipment location only incurs the distance-dependent cost for the resulting detour. We assume that both the parking and load transfer poses no additional cost. All customers and all transshipment locations may have a *time window* associated with them. Each customer must be visited exactly once. The customers' supplies must be collected and transported to a central *depot*. To this end, a *fleet of heterogeneous vehicles with limited loading capacity* is available.

There are four main criteria in which the vehicles differ: First of all, the fleet is comprised of *single lorries* and of *lorry-trailer combinations* (*LTCs*) (see Figure 1).

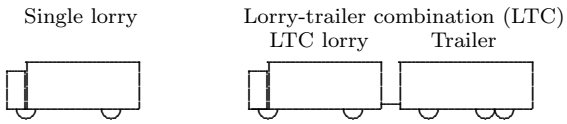

Figure 1: GTTRP fleet

There is a fixed assignment of an LTC lorry to its trailer (and *vice versa*), i.e., each trailer may be pulled by only one lorry, and only this lorry may perform a load transfer into this trailer. Second, the vehicles may have different *loading capacities*: Evidently, single lorries have fewer loading capacity than lorry-trailer combinations. In addition, also vehicles of the same type (that is, single lorries or LTCs) may have different capacities. For example, a single lorry may have two axles, limiting the loading capacity (payload) to approximately 10 tons, another may have three axles, leading to a loading capacity of about 15 tons (see also Section 5.2). For LTCs, two capacities are relevant, the lorry capacity and the trailer capacity. Third, because of different sizes and total weights, the vehicles may have different fixed and distance-dependent *costs*. Fourth, the vehicles may be subject to *accessibility constraints*, i.e., they may not be allowed to visit some customers. Such constraints are mostly due to limited manoeuvring space on customer premises or to bad road conditions, especially in winter. The driving speeds are assumed to be identical for all lorries, whether or not they pull a trailer, and whether or not they carry load. An LTC lorry need not use its trailer at all and may simply leave it at the depot. Also, it can decouple and re-couple its trailer at any transshipment location. All vehicles are initially located at the depot, and all vehicles must return to the depot at the end of their tour. That is, a trailer cannot be left behind at a transshipment location by its lorry, it must

also be pulled back to the depot. Each vehicle may perform at most one tour. If, in a solution, a vehicle is not used, the vehicle's fixed cost is not added to the objective function value of the solution. This is for two reasons. On the one hand, for a tactical planning problem, where the fleet size and mix are to be determined, not using a vehicle in a solution means that the vehicle need not be acquired or rented, so that the fixed cost is actually not incurred. On the other hand, for an operative planning problem, not using a vehicle in a solution means that the vehicle is available for other purposes. Moreover, in both a tactical and an operative setting, using fixed vehicle costs is a way to reduce the overall number of vehicles used, an objective often pursued in vehicle routing problems.

Customers which can only be visited by a lorry without a trailer are called *lorry customers*. The other customers, which can be visited by a single lorry, an LTC lorry, or an LTC, are called *trailer customers*. Split collection is not allowed, so there are no customers with a supply exceeding the capacity of the largest single or LTC lorry (or LTC, in the case of trailer customers).

The single lorries are used as in a 'usual' VRP. The LTCs are used as follows. To visit lorry customers, the trailer is parked at a transshipment location and the LTC lorry does some collecting. It then returns to its trailer and either transfers its load to the trailer and continues collecting, or it re-couples the trailer, parks it elsewhere, decouples and does some more collecting before transferring load. When a trailer customer is visited by an LTC, the supply of the customer can directly be loaded completely or partly into the trailer. In essence, the trailers are used as mobile depots and extend the capacity of their lorries. A load transfer between two lorries or between two trailers is not allowed. This means that single lorries never visit transshipment locations.

Figure 2 depicts a possible route plan with one single lorry and one LTC. The single lorry route is evident. The LTC lorry starts at the depot with its trailer, visits a trailer customer, goes on to a transshipment location, decouples the trailer, visits two customers, returns to the transshipment location, re-couples the trailer, pulls it to a second transshipment location, decouples it again (and perhaps performs a load transfer), visits two more customers, returns to the second transshipment location, re-couples the trailer and goes back to the depot. Two transshipment locations in the centre of the figure are not used.

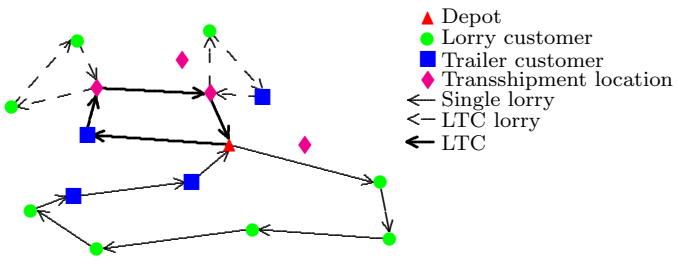

Figure 2: Example GTTRP route plan

The task is to devise tours minimizing the sum of fixed and distance-dependent, that is, variable, cost over all vehicles (some of which may not be needed), such that the complete supply of all customers is collected and delivered to the depot while respecting vehicle capacities, accessibility constraints, and location time windows.

To see when and why it is sensible to use LTCs, imagine that the supply of the customers visited by the LTC lorry exceeds the capacity of the largest single lorry. Then, without LTCs, two single lorry tours are necessary to serve these customers, so that overall, three lorries (and, consequently, three drivers) are required instead of two as in the above figure. This will increase overall fixed costs. Moreover, in general, the overall distance travelled will increase as well, thus increasing variable costs.

The GTTRP is of considerable practical relevance. Actual and potential applications are fuel oil delivery to private households, food distribution to supermarkets ([42]), raw milk collection at farmyards ([6], [44], [18], [39]), intermodal container transportation ([22]), and the so-called park-and-loop problem encountered in postal delivery ([2], [4]). Moreover, many location-routing problems (LRPs, see [28], pp. 169 ff., [11], pp. 339 ff., [33], [34]) can be modelled as GTTRPs.

The problem just described is denoted *generalized* TTRP, because it generalizes the TTRPs considered in the literature (see next section) in the following ways:

- Variable costs for the trailers and fixed costs for the vehicles are considered.
- Time windows are considered.
- Trailer customer locations as well as pure transshipment locations can be used for parking trailers and performing load transfers.

No other paper cited below considers these three aspects simultaneously. In particular, the presence of pure transshipment locations adds an additional degree of freedom, because visiting a pure transshipment location is optional. Hence, a selection has to be made as to which transshipment locations to use. In most variants of the VRP or the VRP with time windows (VRPTW) documented in the literature, with the notable exception of location-routing problems, such a selection component is not present.

The remainder of the paper is structured as follows. In the next section, the existing literature is reviewed. In Section 3, a mixed-integer programming (MIP) model for the GTTRP, based on binary arc flow and continuous resource variables, is developed. In Section 4, branch-and-price algorithms based on a path flow reformulation of this model are presented. Computational results with implementations of these algorithms are presented and analyzed in Section 5. The paper ends with a brief summary and a research outlook in Section 6.

2 Literature Review

The practical relevance of the problem is not adequately reflected by the existing literature. The following is a brief review of the few contributions there are. [6], [44], and [18] consider only one transshipment location per trailer. [6] develops a heuristic sequential solution approach for this type of problem (clustering of customers, determination of one transshipment location per trailer, routing). [44] proceeds similarly. [18] presents several construction heuristics and intra- and inter-tour exchange improvement procedures. Also, the paper mentions an unpublished working paper containing an MIP formulation.

[42], [41], [8], and [39] (see also [40]) allow that trailers be parked several times at different locations. However, they equate the potential trailer parking locations with the trailer customers. This means that a trailer can be parked at any trailer customer location but nowhere else, and that only one trailer can be parked at such a customer location, because the corresponding lorry is then assumed to service the customer. [42] consider time windows and heterogeneous lorries, but identical trailers. [42] and [41] also consider accessibility constraints for the lorries. [8] and [39] limit the number of available vehicles and the length or duration of a tour. The term 'Truck-and-Trailer Routing Problem' (TTRP) was coined by [8] and is also used by [39], [40], [20], [21], [31], [32]. [39] extends the approaches of [41] and [8] and considers a multi-period and a multi-depot version of the problem. [32] consider the case of an unlimited number of vehicles. Contrary to all other papers, [20] and [21] consider the situation where the transshipment locations are separate locations and do not coincide with the trailer customer locations. All authors solve their respective problems by sophisticated heuristic procedures, mostly tabu search and simulated annealing. [41] and [39] also give mathematical programming formulations of the problems they consider (0-1 IP models), but they do not solve their models with an exact method.

The present paper is based on the technical report [14]. In the next subsection, a network representation for the GTTRP is presented. Interestingly, a similar network was used in [29] for the solution of a location-routing problem.

3 A Mixed-Integer Programming Model for the GTTRP

3.1 Network Representation

The subsequent formulation is based on a 'time-space-operation' network $D = (V, A)$. Each vertex in V corresponds to a location in space, an absolute and/or relative period of time, and a type of operation. $L{:=}\{Depot\} \cup L_C \cup L_T$ is the set of relevant real-world locations. $L_C := L_{C_L} \cup L_{C_{LT}}$ is the set of customer locations, which is partitioned into L_{C_L}, the set of lorry customers, and $L_{C_{LT}}$, the set of trailer customers. L_T is the set of *pure* transshipment locations. For each $i \in V$, $loc_i \in L$ denotes the location corresponding to i, $[a_i, b_i] \subseteq [0, T]$, is

the arrival time window of i, where T is the length of the planning horizon, and s_i is the supply of vertex i (which is zero for depot and transshipment vertices). V is comprised as follows. There is one start depot vertex o and one end depot vertex d, both with a time window of $[0, T]$. For each customer, there is one customer vertex. Let $V_C := V_{C_L} \cup V_{C_{LT}}$ be the set of customer vertices, where V_{C_L} is the set of lorry customers, and where $V_{C_{LT}}$ is the set of trailer customers. Following [8] and [39], it is assumed that the locations corresponding to trailer customers can also be used for parking and transshipment operations. Hence, for each trailer customer location and for each (pure) transshipment location l, there are $n_{TS} \geqq 3$ vertices v_l^{dec}, $v_l^{trans,1}$, ..., $v_l^{trans,n_{TS}-2}$, v_l^{coup}. Let V_{dec} be the set of decoupling vertices, let V_{trans} be the set of transfer vertices, and let V_{coup} be the set of coupling vertices. Let $V_I := V_{dec} \cup V_{trans} \cup V_{coup}$ be the set of transshipment vertices. The idea behind this separation of transshipment locations and processes is the following: At a vertex in V_{dec}, an LTC lorry may decouple its trailer, and it may also perform a load transfer. Decoupling vertices can be reached only with a trailer and left without a trailer. At a vertex in V_{trans}, an LTC lorry may perform a load transfer to its trailer. Transfer vertices can be reached and left only without a trailer. At a vertex in V_{coup}, it may re-couple its trailer, and it may again perform a load transfer. Coupling vertices can be reached only without a trailer and left only with a trailer. It is assumed that each vertex is visited by each vehicle at most once. If an LTC wants to use transshipment location l, the LTC must first visit v_l^{dec}. The trailer then moves in time to the vertices $v_l^{trans,1}$, ..., $v_l^{trans,n_{TS}-2}$, v_l^{coup}, while the LTC lorry visits customers and finally re-couples the trailer at v_l^{coup}. The LTC lorry need not visit any of the transfer vertices of l. The LTC lorry must not visit any other vertex in V_I before having re-coupled its trailer at v_l^{coup}. The formulation below takes this into account.

$F := F_L \cup F_{LT}$ denotes the set of vehicles. F_L is the set of single lorries, which do not have a trailer. F_{LT} is the set of lorry-trailer combinations. For all $k \in F$, q_k^{total} is the total capacity of a vehicle, i.e., it is the capacity of single lorry k, or, respectively, the capacity of LTC lorry k and its trailer. For all $k \in F_{LT}$, q_k^{lorry} is the capacity of LTC lorry k, and $q_k^{trailer}$ is the capacity of LTC lorry k's trailer. All vehicles are initially at the start depot vertex o and end their tour at the end depot vertex d. Single lorries are allowed, in principle, to visit lorry and trailer customer vertices, and LTC lorries may, in principle, visit all vertices of D. Trailers can only reach the transshipment vertices and the trailer customers. However, as mentioned above, there may be additional accessibility constraints for certain vehicles at certain customers (e.g., trailer customers with too much supply for a single lorry).

It is assumed that each LTC uses each transshipment location at most once. Thus, for each trailer, there are at most n_{TS} transshipment operations at each transshipment location. As for the 'correct' choice of n_{TS}, note that in the worst case, a *feasible* solution to the model may exist only when, for each trailer, there are as many intermediate and coupling vertices as there are lorry customers, because it may be necessary to perform a load transfer after each visit to a lorry customer. Consequently, an *optimal* solution to the model may only be an optimal solution to an underlying problem instance when there are as many opportunities for a load transfer *at*

the right transshipment location as there are lorry customers. Depending on the instance data, better values for n_{TS} can be computed. However, as will become clear in the next section, in the branch-and-price algorithm, the numerical value of n_{TS} need not be fixed in advance; rather, this value is determined during the solution of the pricing problems.

The arc set consists of the following arcs:

- (o, d)
- (o, v_c) and (v_c, d) for all customer vertices $v_c \in V_C$
- (o, v_l^{dec}) for all transshipment locations $l \in L_{C_{LT}} \cup L_T$
- (v_l^{coup}, d) for all transshipment locations $l \in L_{C_{LT}} \cup L_T$
- $(v_c, v_{c'})$ for all customer vertices $v_c, v_{c'} \in V_C$ with $c \neq c'$
- (v_l^{dec}, v_c) for all transshipment locations $l \in L_{C_{LT}} \cup L_T$ and all customer vertices $v_c \in V_C$
- $(v_l^{trans,i}, v_c)$ for all transshipment locations $l \in L_{C_{LT}} \cup L_T$, all $i \in \{1, \ldots, n_{TS} - 2\}$ and all customer vertices $v_c \in V_C$
- (v_l^{coup}, j) for all transshipment locations $l \in L_{C_{LT}} \cup L_T$ and all vertices $j \in (V_{dec} \setminus \{v_l^{dec}\}) \cup V_{C_{LT}}$
- (i, j) for all trailer customer vertices $i \in V_{C_{LT}}$ and all transshipment vertices $j \in V_I$
- (i, j) for all lorry customer vertices $i \in V_{C_L}$ and all transfer and coupling vertices $j \in V_{trans} \cup V_{coup}$

For each arc $(i, j) \in A$, τ_{ij}^{tr} is the traversal time. τ_{ij}^{tr} includes the service time at vertex i. For all $k \in F$, let V^k (A^k) be the set of vertices (arcs) that can be reached (traversed) by vehicle k. For all subsets of V (A) defined hereafter, let the superscript k denote the intersection of the respective subset with V^k (A^k). In particular, $A_{LT}^k := A^k \setminus (\{(i, j) \in A : i \in V_{C_L} \cup V_{dec}\} \cup \{(i', j') \in A : j' \in V_{C_L} \cup V_{coup}\})$ is the set of arcs LTC lorry k can traverse with its trailer attached. $c_{ij}^{lorry,k}$ denotes the cost of traversing arc (i, j) with lorry $k \in F$ (for LTC lorries, without its trailer attached), and $c_{ij}^{trailer,k}$ denotes the *additional* cost of pulling LTC lorry k's trailer over arc (i, j). On arcs emanating from the start depot o, except for the arc (o, d), the fixed vehicle cost is included in $c_{oj}^{lorry,k}$, respectively, $c_{oj}^{trailer,k}$.

Similar to [8] and [39], a fixed time for load transfer is assumed, independent of the actual amount of load transferred. This is a sensible assumption when the setup cost is the determining factor for the duration of a transshipment operation.

Figure 3 visualizes the subnetworks for the different vehicle types. To keep the figure clear and concise, there is only one arc for each arc type present in the subnetwork of the respective vehicle type. For example, in the LTC subnetwork, one arc from the left trailer customer to the right one is depicted to indicate that LTCs can freely move from any trailer customer to any other trailer customer (unless capacity, time window, or accessibility constraints prohibit this). The absence of an arc from the right trailer customer to the left one in the figure does not mean that there is no such arc in the network.

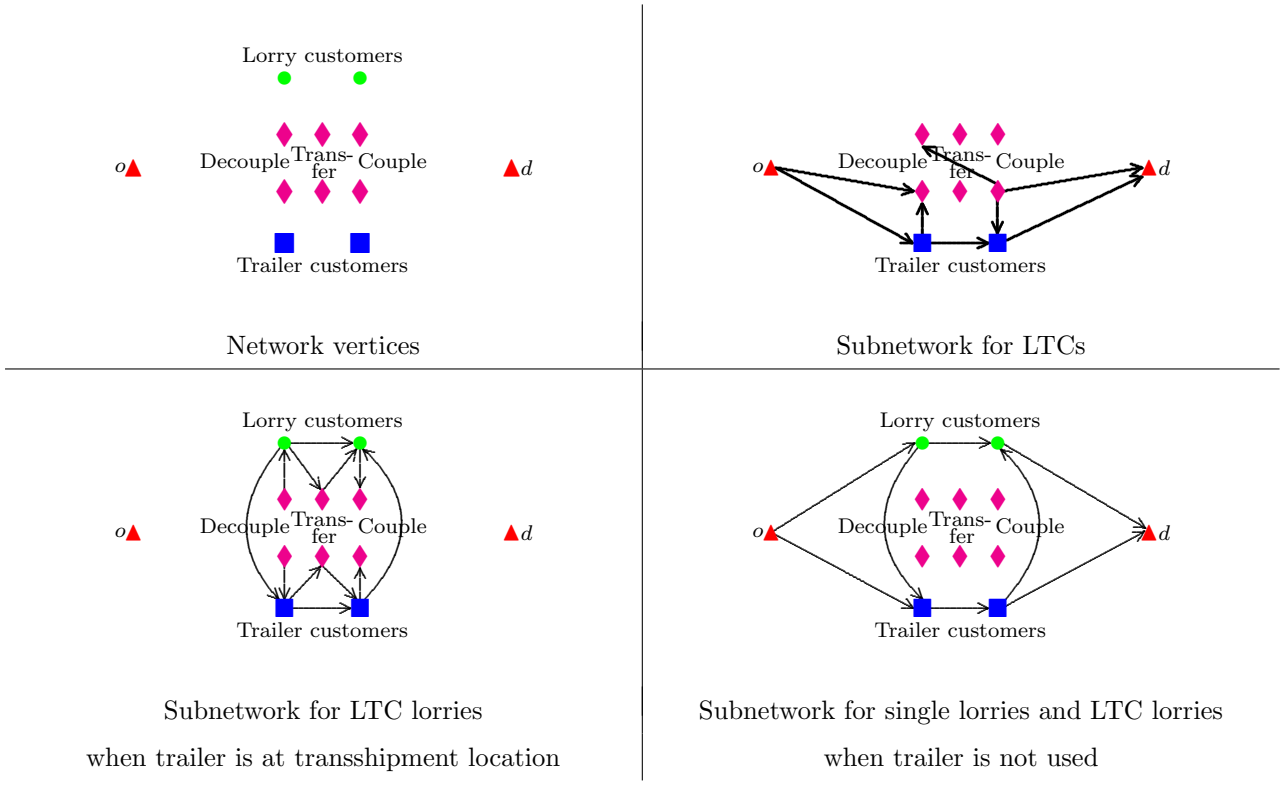

Figure 3: Network structure

3.2 Formulation

The subsequent formulation uses the following variables:

- $x_{ij}^k \in \{0,1\} \; \forall \; k \in F, (i,j) \in A^k$.
  $x_{ij}^k = \begin{cases} 1, \text{lorry } k \text{ traverses arc } (i,j) \\ 0, \text{otherwise} \end{cases}$
- $y_{ij}^k \in \{0,1\} \; \forall \; k \in F_{LT}, (i,j) \in A_{LT}^k$.
  $y_{ij}^k = \begin{cases} 1, \text{LTC lorry } k \text{ traverses arc } (i,j) \textit{ with} \text{ its trailer attached} \\ 0, \text{LTC lorry } k \text{ does } \textit{not} \text{ traverse arc } (i,j) \text{ with its trailer attached} \end{cases}$
- $l_i^{coll,k} \in \mathbb{R}_+ \; \forall \; k \in F, i \in V^k$.
  The total amount of customer supplies that lorry k has collected when reaching vertex i, *before* k starts its service at i.
- $l_i^{trans,k} \in \mathbb{R}_+ \; \forall \; k \in F_{LT}, i \in V^k$.
  The total amount of customer supplies that LTC lorry k has transferred to its trailer when reaching vertex i, or, equivalently, the load of LTC lorry k's trailer when k reaches vertex i (with or without its trailer attached), *before* k starts its service at i.
- $t_i^k \in \mathbb{R}_+ \; \forall \; k \in F, i \in V^k$.
  The point in time when lorry k starts its service at vertex i.

The resulting model is:

(GTTRP):

$$\min \sum_{k \in F} \sum_{(i,j) \in A^k} c_{ij}^{lorry,k} x_{ij}^k + \sum_{k \in F_{LT}} \sum_{(i,j) \in A_{LT}^k} c_{ij}^{trailer,k} y_{ij}^k \text{ subject to} \tag{1}$$

$$\sum_{k \in F} \sum_{(h,i) \in A^k} x_{hi}^k = 1 \quad \forall\, i \in V_C \tag{2}$$

$$\sum_{(h,i) \in A^k} x_{hi}^k \leqq 1 \quad \forall\, k \in F_{LT}, i \in V_{dec} \tag{3a}$$

$$\sum_{(v_l^{coup},j) \in A_{LT}^k} x_{v_l^{coup} j}^k - \sum_{(h,v_l^{dec}) \in A_{LT}^k} x_{h v_l^{dec}}^k = 0 \quad \forall\, k \in F_{LT}, l \in L_{C_{LT}} \cup L_T \tag{3b}$$

$$\sum_{(h,i) \in A^k} x_{hi}^k - \sum_{(v_l^{coup},j) \in A_{LT}^k} x_{v_l^{coup} j}^k \leqq 0 \quad \forall\, k \in F_{LT}, l \in L_{C_{LT}} \cup L_T, i \in V_{trans}, loc_i = l \tag{3c}$$

$$\sum_{(i,d) \in A^k} x_{id}^k = 1 \quad \forall\, k \in F \tag{3d}$$

$$\sum_{(i,d) \in A_{LT}^k} y_{id}^k = 1 \quad \forall\, k \in F_{LT} \tag{3e}$$

$$\sum_{(h,i) \in A^k} x_{hi}^k - \sum_{(i,j) \in A^k} x_{ij}^k = 0 \quad \forall\, k \in F, i \in V^k \setminus \{o,d\} \tag{3f}$$

$$\sum_{(h,i) \in A_{LT}^k} y_{hi}^k - \sum_{(i,j) \in A_{LT}^k} y_{ij}^k = 0 \quad \forall\, k \in F_{LT}, i \in V_{C_{LT}}^k \tag{3g}$$

$$y_{ij}^k \leqq x_{ij}^k \quad \forall\, k \in F_{LT}, (i,j) \in A_{LT}^k \setminus \{(o,d)\}, i \notin V_{coup}, j \notin V_{dec} \tag{4a}$$

$$x_{hi}^k = y_{hi}^k \quad \forall\, k \in F_{LT}, i \in V_{dec}, (h,i) \in A_{LT}^k \tag{4b}$$

$$x_{ij}^k = y_{ij}^k \quad \forall\, k \in F_{LT}, i \in V_{coup}, (i,j) \in A_{LT}^k \tag{4c}$$

$$l_i^{trans,k} \leqq l_i^{coll,k} \quad \forall\, k \in F_{LT}, i \in V^k \tag{5a}$$

$$l_i^{coll,k} - l_i^{trans,k} \leqq q_k^{lorry} \quad \forall\, k \in F_{LT}, i \in V^k \tag{5b}$$

$$x_{ij}^k = 1 \Rightarrow l_i^{coll,k} + s_i \leqq l_j^{coll,k} \quad \forall\, k \in F, (i,j) \in A^k \tag{5c}$$

$$x_{ij}^k = 1 \Rightarrow l_i^{trans,k} \leqq l_j^{trans,k} \quad \forall\, k \in F_{LT}, (i,j) \in A^k \tag{5d}$$

$$x_{ij}^k = 1 \Rightarrow l_i^{trans,k} \geqq l_j^{trans,k} \quad \forall\, k \in F_{LT}, i \in V_{C_L}, (i,j) \in A^k \tag{5e}$$

$$x_{ij}^k = 1 \wedge y_{ij}^k = 0 \Rightarrow l_j^{trans,k} \leqq l_i^{trans,k} \quad \forall\, k \in F_{LT}, i \in V_{C_{LT}}, (i,j) \in A^k \tag{5f}$$

$$y_{ij}^k = 1 \Rightarrow l_j^{trans,k} \leqq l_i^{trans,k} + s_i \quad \forall\, k \in F_{LT}, i \in V_{C_{LT}}, (i,j) \in A_{LT}^k \tag{5g}$$

$$x_{ij}^k = 1 \Rightarrow t_i^k + \tau_{ij}^{tr} \leqq t_j^k \quad \forall\, k \in F, (i,j) \in A^k \tag{6a}$$

$$t_{v_l^{dec}}^k \leqq t_{v_l^{trans,1}}^k \quad \forall\, k \in F_{LT}, l \in L_{C_{LT}} \cup L_T \tag{6b}$$

$$t_{v_l^{trans,i}}^k \leqq t_{v_l^{trans,i+1}}^k \quad \forall\, k \in F_{LT}, i \in \{1, \ldots, n_{TS} - 3\}, l \in L_{C_{LT}} \cup L_T \tag{6c}$$

$$t_{v_l^{trans,n_{TS}-2}}^k \leqq t_{v_l^{coup}}^k \quad \forall\, k \in F_{LT}, l \in L_{C_{LT}} \cup L_T \tag{6d}$$

$$t^k_{v^{coup}_l} \leqq T \sum_{(h,v^{dec}_l)\in A^k_{LT}} x^k_{hv^{dec}_l} \quad \forall\, k \in F_{LT}, l \in L_{C_{LT}} \cup L_T \tag{6e}$$

$$x^k_{ij} \in \{0,1\} \quad \forall\, k \in F, (i,j) \in A^k \tag{7a}$$

$$y^k_{ij} \in \{0,1\} \quad \forall\, k \in F_{LT}, (i,j) \in A^k_{LT} \tag{7b}$$

$$0 \leqq l^{coll,k}_i \leqq q^{total}_k \quad \forall\, k \in F, i \in V^k \tag{7c}$$

$$0 \leqq l^{trans,k}_i \leqq q^{trailer}_k \quad \forall\, k \in F_{LT}, i \in V^k \tag{7d}$$

$$a_i \leqq t^k_i \leqq b_i \quad \forall\, k \in F, i \in V^k \tag{7e}$$

The objective function (1) minimizes total fixed and variable costs over all lorries and all trailers.

Seen from a column generation perspective, constraints (2) are *linking* constraints affecting more than one vehicle within each single constraint, (3) and (4) are *non-linking* or *independent* flow constraints, (5) and (6) are non-linking constraints specifying the update of the load and time resource variables respectively, and (7) determine the ranges of the variables.

With regard to contents, the constraints have the following meaning: (2) are the customer covering constraints stating that each customer must be visited exactly once. (3a)–(3c) make sure that each transshipment location is used at most once by each $k \in F_{LT}$, and that only intermediate and coupling vertices of those transshipment locations are used whose decoupling vertex is visited. (3d) and (3e) require that each vehicle reach the end depot (possibly via the arc (o,d)). (3f) and (3g) are flow conservation constraints: They ensure that if a vehicle enters a vertex (except for the start and the end depot vertex), the vehicle also leaves this vertex. (4a)–(4c) are the constraints linking the routing of the trailer to that of its lorry. Note that it is possible that an LTC lorry does not use its trailer, in which case the latter moves directly from o to d. (5a) require that the load of the trailer be at most equal to the total amount of supply collected. (5b) require that the amount of supply collected be not greater than the lorry capacity and the amount of supply already transferred to the trailer. Without constraints (5a), $l^{trans,k}_i$ could always be set equal to the trailer capacity. Because of constraints (5b), $l^{coll,k}_i$ would then be bounded from above only by the overall vehicle capacity, and not by the lorry capacity and the amount of supply already transferred. This would illicitly enlarge the lorry capacity. A lorry could then park its trailer before doing any collecting and collect as much supply as the overall vehicle capacity permits, without having to transfer any load. (5c) are the load update constraints. The total amount of load collected increases (at least) by the supply of each vertex visited. Similarly, constraints (5d)–(5g) are for the update of the transshipment variables, respectively, the trailer load variables. (5d) state that the trailer load is non-decreasing. (5e) state that the trailer load is non-increasing at lorry customer vertices. (5f) state that at trailer customer vertices, the trailer load of an LTC k can only increase if the vertex is visited by the lorry *and* its trailer. (5g) state that at a trailer customer vertex, the trailer load does not increase by more than the customer's supply. Constraints (6a) are the constraints for the update of the timing variables: The overall time en route for a vehicle after

traversing an arc increases (at least) by the travel time along the arc. (6b)–(6e) are needed to make sure that the vertices of a transshipment location are visited in the correct order, i.e., that decoupling vertices of a transshipment location are visited *before* their corresponding coupling vertex. Without these constraints, it would be possible to visit decoupling vertex $v_{l_1}^{dec}$, go to coupling vertex $v_{l_2}^{coup}$, then to decoupling vertex $v_{l_2}^{dec}$ and then to $v_{l_1}^{coup}$. Note that if an LTC lorry does not visit a transshipment intermediate vertex for some l, the time variable for this vertex can be fixed appropriately; it is not required to have a visiting time of zero for non-visited vertices.

To improve readability, constraints (5c)–(6a) are written as logical implications. However, these can easily be linearized using a Big-M technique without introducing additional variables (see [46]). Hence, the convex hull of all points fulfilling (2)–(7) is a bounded polyhedron.

The only existing formulations for the TTRP, by [41] and [39], are conceptually different from the above formulation. [39] uses binary three-index arc flow variables similar to the above y_{ij}^k variables, and five-index variables indicating whether a certain lorry traverses a certain arc on the nth subtour starting at a certain trailer customer or at the depot, where n is the number of customers (in the worst case, as many single lorry (sub-)tours starting at a trailer customer or at the depot are necessary as there are customers). [41] uses similar variables, but requires that at most one subtour originate at each trailer customer, and hence the second variable type has only four indices in his model. Neither author uses resource variables; rather, both formulations are pure 0-1 IPs.

In the GTTRP, several lorry-trailer combinations may use the same transshipment location. Hence, the GTTRP formulation (1)–(7), using the arc set defined in Section 3.1, is a relaxation of the TTRP models from the literature. To make sure that only the lorry which visits a certain trailer customer vertex h uses the corresponding location l for parking and load transfer, the arc set can be restricted by removing all arcs entering the corresponding decoupling vertex i, except for the arc (h, i).

4 Branch-and-Price Algorithms for the GTTRP

For capacitated vehicle routing problems (with time windows), many authors have devised formulations and solution approaches based on the integer programming version of Dantzig-Wolfe decomposition, i.e., branch-and-price. Standard references focused on the methodology include [3], [45], and [30]. [12] present a unified model for the solution of time-constrained vehicle routing and scheduling problems by branch-and-price. In this section, a reformulation of (1)–(7) for use in a branch-and-price algorithm is developed. The subsequent exposition follows [12].

The usual decomposition approach for VRPs can also be applied to the GTTRP. The linking constraints, i.e., the customer covering constraints (2), go in the master program, the non-linking constraints (3)–(7) define the sub- or pricing problems.

4.1 The Master Program

The model presented in Section 3 uses arc flow variables for each relevant combination of lorry or trailer k and arc (i, j). Branch-and-price approaches for vehicle routing problems use path flow variables in the master program: There is one variable for each feasible o-d-path of each vehicle. As for VRPs without trailers, also for the GTTRP, each feasible tour of a single lorry is an elementary path from the start depot vertex to the end depot vertex through the respective subnetwork. This is also true for the LTC lorries. However, the movements of a trailer in the network D of Section 3 do not constitute a path: There are no trailer flow variables leaving decoupling vertices or entering coupling vertices. This, though, is only to avoid unnecessary variables and constraints. In the real world, the itinerary of a single lorry or a trailer resulting from a solution to the above formulation is an elementary path, too, whereas the itinerary of an LTC lorry using its trailer is not elementary; it will contain cycles starting and ending at the transshipment locations where the trailer is parked.

It follows from the flow decomposition theorem (see, for example, [1], p. 80 f.) that the extreme points of the convex hull of all points fulfilling the pricing problem constraints (3)–(7) correspond to lorry paths from o to d in D, some of which may be non-elementary, because D is not acyclic. For LTCs, constraints (3d)–(4c) additionally imply that the extreme points represent a path-like structure consisting of an o-d-path for the lorry and one or more paths for its trailer. The union of these trailer paths is a subset of the lorry path.

These extreme points are described by flow and resource vectors

$$(x_p^k, y_p^k, l_p^{coll,k}, l_p^{trans,k}, t_p^k) = (x_{ijp}^k, y_{ijp}^k, l_{ip}^{coll,k}, l_{ip}^{trans,k}, t_{ip}^k) \quad \forall\ k \in F, p \in P^k, (i,j) \in A^k, \tag{8a}$$

where P^k is the set of extreme points for single lorry or LTC k, and where, for simplicity of notation, in the following

$$y_p^k = y_{ijp}^k := 0 \quad \forall\ k \in F_L,$$

and

$$l_p^{trans,k} = l_{ip}^{trans,k} := 0 \quad \forall\ k \in F_L.$$

Any solution satisfying (3)–(7) can be expressed as a convex combination of these extreme points:

$$x_{ij}^k = \sum_{p \in P^k} x_{ijp}^k \lambda_p^k \quad \forall\ k \in F, (i,j) \in A^k \tag{8b}$$

$$x_{ij}^k \in \{0,1\} \quad \forall\ k \in F, (i,j) \in A^k \tag{8c}$$

$$y_{ij}^k = \sum_{p \in P^k} y_{ijp}^k \lambda_p^k \quad \forall\ k \in F_{LT}, (i,j) \in A_{LT}^k \tag{8d}$$

$$y_{ij}^k \in \{0,1\} \quad \forall\ k \in F_{LT}, (i,j) \in A_{LT}^k \tag{8e}$$

$$l_i^{coll,k} = \sum_{p \in P^k} l_{ip}^{coll,k} \lambda_p^k \quad \forall\ k \in F, i \in V^k \tag{8f}$$

$$l_i^{trans,k} = \sum_{p \in P^k} l_{ip}^{trans,k} \lambda_p^k \quad \forall\ k \in F_{LT}, i \in V^k \tag{8g}$$

$$t_i^k = \sum_{p \in P^k} t_{ip}^k \lambda_p^k \quad \forall\ k \in F, i \in V^k \tag{8h}$$

$$\sum_{p \in P^k} \lambda_p^k \leqq 1 \quad \forall\ k \in F \tag{8i}$$

$$\lambda_p^k \geqq 0 \quad \forall\ k \in F, p \in P^k \tag{8j}$$

The integer master program (IMP) is then:

$$\sum_{k \in F} \sum_{p \in P^k} \left(\sum_{(i,j) \in A^k} c_{ij}^{lorry,k} x_{ijp}^k \right) \lambda_p^k + \sum_{k \in F_{LT}} \sum_{p \in P^k} \left(\sum_{(i,j) \in A_{LT}^k} c_{ij}^{trailer,k} y_{ijp}^k \right) \lambda_p^k \rightarrow \min \tag{9a}$$

subject to

$$\sum_{k \in F_L} \sum_{p \in P^k} \left(\sum_{i \in V_C^l} \sum_{(h,i) \in A^k} x_{hip}^k \right) \lambda_p^k = 1 \quad \forall\ l \in L_C \tag{9b}$$

$$\sum_{p \in P^k} \lambda_p^k \leqq 1 \quad \forall\ k \in F \tag{9c}$$

$$\lambda_p^k \geqq 0 \quad \forall\ k \in F, p \in P^k \tag{9d}$$

$$\lambda_p^k \in \{0,1\} \quad \forall\ k \in F, p \in P^k \tag{9e}$$

In general, it is not true that binary restrictions on the original x_{ij}^k arc flow variables can be replaced by binary restrictions on the λ_p^k path flow variables. However, this statement holds for the above GTTRP reformulation, since the definition of the original master problem solution space, i.e., the constraint set (2), involves only the x_{ij}^k variables. Hence, binary requirements on the path variables are equivalent to binary requirements on the x_{ij}^k variables, see [12], p. 75. Moreover, if the x_{ij}^k variables are binary, the y_{ij}^k variables will be binary, too. This is because for any path of an LTC lorry k represented by the values of the x_{ij}^k variables, the values of the pertinent y_{ij}^k variables are unequivocally determined: If the path visits transshipment vertices, all partial paths ending at a decoupling vertex or starting at a coupling vertex have all y_{ij}^k variables equal to one, and all partial paths between a decoupling and its corresponding coupling vertex have all y_{ij}^k variables equal to zero. Paths visiting only customers and at least one lorry customer trivially have all y_{ij}^k variables equal to zero, and paths visiting only trailer customers have all y_{ij}^k variables equal to one, unless the total demand of the customers on the path does not exceed the lorry capacity, in which case the corresponding trailer will not be used.

4.2 The Pricing Problems

As it is the case for all vehicle routing problems, also the pricing problems for the GTTRP are (elementary) shortest path problems with resource constraints ((E)SPPRCs) (see [24]). When the fleet is heterogeneous, in each pricing step, a pricing problem must be solved for each vehicle. Because of the dual prices coming from the master problem, the pricing problem networks usually have negative cost cycles. This makes the problem $\mathcal{NP}$-hard in the strong sense (ibid., p. 38). There is a considerable amount of literature on the solution of (E)SPPRCs. It is beyond the scope of this paper to give a detailed review, but some important contributions are [16], [37], [38], [36], [5], [7], [25].

The standard solution technique for (E)SPPRCs is a labelling algorithm based on dynamic programming. In principle, such an algorithm works similar to a labelling algorithm for shortest path problems without resource constraints, e.g., the Dijkstra algorithm (see [1], pp. 93 ff.). The *basic concepts* used in such an algorithm are the following (see [24]). A *resource* is an arbitrarily scaled one-dimensional quantity that can be determined or computed at the vertices of a directed walk in a network. Examples are cost, time, load, or the information 'Is LTC lorry k currently pulling its trailer?'. The value of a resource at a vertex is stored in a *resource variable*. An arbitrarily scaled resource is *constrained* if there is at least one vertex in the network where the associated resource variable must not take all possible values. A cardinally scaled resource is constrained if there is at least one vertex in the network with a finite upper or lower bound on the value of the resource. The *resource window* of a nominally scaled resource r at a vertex is the set of allowed values of r at this vertex. The resource window of a cardinally scaled resource r at a vertex i is the interval $[lb_i^r, ub_i^r] \subseteq]-\infty, +\infty[$. A *resource extension function* (*REF*) is defined on each arc in a network for each resource considered. An REF for a resource r maps the set of all possible vectors of resource values at the tail of an arc to the set of possible values of r at the head of the arc. More precisely, let $R := (\sigma^1, \ldots, \sigma^{|R|})^T$ be a vector of (values of) resource variables. Then, an REF for a resource r is a function $f^r : A \times \mathbb{R}^{|R|} \mapsto \mathbb{R}$. For simplicity, let $f_{ij}^r(R) := f^r(((i,j), R))$. An REF for a cardinally scaled resource r indicates (lower bounds on) the consumptions of r along the arcs. When seeking a path from an origin vertex o to a destination vertex d, partial paths from o to a vertex $i \neq d$ are *extended* along all arcs (i,j) emanating from i to create new, extended paths. For each o-i-path, there is a corresponding *label l resident at i* that stores the values of all resource variables at i for its path, along with the information on how it was created: the arc (h,i) over which i was reached and (a reference to) the label of the o-h-path whose extension along (h,i) yielded the o-i-path (this makes it easy to reconstruct the path corresponding to a label). Initially, there is exactly one label corresponding to the path (o). W.l.o.g., the values of the resource variables of the initial label are all set to the lower bounds of their respective resource windows at o. A *label l* is *feasible* iff the value of each resource variable in l is within the resource window of its respective resource. If a label is not feasible, it is discarded. An *extension* of a path/label along an arc (i,j) is *feasible* iff the resulting label at j is feasible. An *h-j-path* is *feasible* iff, for each arc (i,i') in the path, a

feasible label at i exists which can be extended along (i, i') to a feasible label at i'. To keep the number of labels as small as possible, it is very important to perform a *dominance procedure* to eliminate feasible but unnecessary labels. A label l_1 *dominates* a label l_2 iff both reside at the same vertex, the value of the resource variable of each nominally scaled resource in l_1 is equal to the corresponding value in l_2, the value of the resource variable of each cardinally scaled resource in l_1 is 'better' (less or greater, depending on the resource) than or equal to the corresponding value in l_2, and the value of the resource variable of at least one cardinally scaled resource in l_1 is strictly 'better' than the corresponding value in l_2. Dominated labels are discarded as well.

The resources used in the labelling algorithm for vehicle k are:

- an unconstrained, cardinally scaled resource for cost
- one cardinally scaled resource for each of the three resource variables used in (1)–(7), i.e., for collected load, transferred load, and time
- one cardinally scaled *visitation counter* resource for each customer
- one nominally scaled auxiliary resource for each trailer

In the following, these resources are described in detail. The auxiliary resource is presented first, because it is needed to describe the REFs for cost and load.

The auxiliary resource is primarily needed for the correct modelling of the routing logic at transshipment locations. It must be considered that, on its itinerary, an LTC lorry must visit the decoupling, transfer, and coupling vertices of each transshipment location in the correct order (or not at all), and that, after an LTC lorry has visited the decoupling vertex of transshipment location l, it must not visit any decoupling, transfer or coupling vertex of any other transshipment location before having visited the coupling vertex of l. To this end, one nominally scaled resource r^{tp} is used to model the trailer logic. r^{tp} ('trailer position') records the current position of the trailer by means of a resource variable σ_i^{tp}, which indicates the position of the trailer when the LTC lorry reaches vertex i. σ_i^{tp} is either zero, meaning that the LTC lorry is currently pulling its trailer, or equal to the vertex number of the decoupling vertex where the trailer was parked. The corresponding REF is $f^{r^{tp}}$ with

$$f_{ij}^{r^{tp}}(\sigma_i^{tp}) = \begin{cases} \sigma_i^{tp}, j \in V_C \cup V_{trans} \cup \{d\} \\ j, j \in V_{dec} \\ 0, j \in V_{coup} \end{cases} \qquad \forall\ (i,j) \in A, \tag{10}$$

and the resource windows at a vertex j are as shown in Table 1.

Resource window	for
$\{0\}$	$j \in \{o, d\} \cup V_{dec}$
$\{1, \ldots, \lvert V \rvert\}$	$j \in V_{C_L}$
$\{0, \ldots, \lvert V \rvert\}$	$j \in V_{C_{LT}}$
$\{i\}$	$j \in V_{trans} \cup V_{coup}, i \in V_{dec}, loc_j = loc_i$

Table 1: Resource windows for r^{tp} at vertex j

Additionally, r^{tp} serves the following purposes:

- It is used to model the load transfer logic at trailer customer and transshipment locations (see below).
- It is used for the update of the visitation counter resources (see below).
- It does not only determine the feasibility of an extension of a path along an arc, it is also relevant for (exact) dominance considerations, as a label can only dominate another label if their respective current trailer positions are equal (see below).
- It is used to consider the trailer costs in the pricing problems for LTCs, i.e., to express the y_{ij}^k variables: If LTC lorry k reaches vertex j via arc (i, j) with its trailer attached, the $c_{ij}^{trailer,k}$ costs are incurred. This is registered in the pricing problems by checking whether $\sigma_j^{tp} = 0$.

For each of the resource variables used in formulation (1)–(7), there is one constrained resource. For the total amount of customer supplies that lorry k has collected when reaching vertex i, the resource r^{coll} with resource variable σ_i^{coll} and REF $f^{r^{coll}}$ with

$$f_{ij}^{r^{coll}}(\sigma_i^{coll}) = \sigma_i^{coll} + s_i \quad \forall\ (i,j) \in A \tag{11}$$

is used. $f_{ij}^{r^{coll}}$ models constraints (5c) and sets σ_j^{coll} to the lowest nonnegative value fulfilling these constraints. The resource window at a vertex j is $[0, q_k^{total}]$.

For the total amount of customer supplies that LTC lorry k has transferred when reaching vertex i, the resource r^{trans} with resource variable σ_i^{trans} and REF $f^{r^{trans}}$ with

$$f_{ij}^{r^{trans}}(\sigma_i^{trans}, \sigma_i^{coll}, \sigma_i^{tp}) =$$

$$\begin{cases} \sigma_i^{trans}, & j \in V_{C_L} \vee (j \in V_{C_{LT}} \wedge \sigma_i^{tp} \neq i) \\ \sigma_i^{trans} + \min\{s_i, q_k^{trailer} - \sigma_i^{trans}\}, & j \in V_{C_{LT}} \wedge \sigma_i^{tp} = i \\ \sigma_i^{trans} + \min\{\sigma_i^{coll} - \sigma_i^{trans}, q_k^{trailer} - \sigma_i^{trans}\}, & j \in V_I \end{cases} \qquad \forall\ (i,j) \in A \tag{12}$$

is used. $f_{ij}^{r^{trans}}$ models constraints (5a), (5b), and (5d)–(5g) and sets σ_j^{trans} to the highest possible value that fulfils these constraints. The resource window at a vertex j is $[0, q_k^{trailer}]$.

This means that an LTC lorry always transfers as much load as possible to its trailer (its complete load or an amount of load equal to the residual capacity of its trailer, whichever is less). This is a sensible stipulation, as it was assumed above that the time needed for a load transfer is fixed, independent of the actual amount of load transferred. If this assumption is abandoned, there is a trade-off between transferring few load and saving time on the one hand, and transferring much load and gaining lorry capacity on the other hand. In the former case, it is possible to visit more customers before the end of their time windows, but the lorry capacity may be insufficient to do so. In the latter case, it is possible to visit more customers before having to perform the next load transfer, but perhaps it is no longer possible to arrive at some customers before the end of their time windows. Correctly treating such a trade-off is non-trivial. It is beyond the scope of this paper to elaborate further on this issue. The reader is referred to [23].

For the point in time when lorry k begins its service at vertex i, the resource r^{time} with resource variable σ_i^{time} and REF $f^{r^{time}}$ with

$$f_{ij}^{r^{time}}(\sigma_i^{time}) = \sigma_i^{time} + \tau_{ij}^{tr} \quad \forall\ (i,j) \in A \tag{13}$$

is used. $f_{ij}^{r^{time}}$ models constraints (6a) and sets σ_j^{time} to the lowest nonnegative value fulfilling these constraints. The resource window at a vertex j is $[a_j, b_j]$.

The branch-and-price algorithm in the present paper solves the elementary SPPRC in each pricing step. For the 'usual' VRP or VRPTW, to solve the ESPPRC by dynamic programming, it is necessary to have a binary visitation counter resource for each customer (vertex). This is also the case for the GTTRP. In the course of the algorithm, for a label l residing at a vertex v, this resource is set to zero for a customer c if it is not or no longer possible to extend l such that c is visited after v on any extension of l. This will mostly be the case because c has already been visited, but possibly also because of time window, capacity and accessibility constraints. To be precise, for each $v_c \in V_C$, there is one binary resource r^{visit,v_c} with resource variable $\sigma_i^{v_c}$ at each vertex $i \in V$ and REF $f^{r^{visit,v_c}}$ with

$$f_{ij}^{r^{visit,v_c}}(\sigma_i^{v_c}, \sigma_j^{coll}, \sigma_j^{trans}, \sigma_j^{time}, \sigma_j^{tp}) =$$

$$\begin{cases} 0 & , v_c = i \vee \text{ not enough capacity } \vee \text{ not enough time} \\ \sigma_i^{v_c}, & \text{otherwise} \end{cases} \qquad \forall\ (i,j) \in A \tag{14}$$

where $\sigma_j^{v_c} = 0$ means that customer v_c cannot be visited any more, and where 'not enough capacity' means

$$\begin{aligned} & \sigma_j^{coll} + s_j > q_k^{total} \\ \vee\ & (v_c \neq j \wedge \sigma_j^{coll} + s_j + s_{v_c} > q_k^{total}) \\ \vee\ & (\sigma_j^{tp} \neq j \wedge \sigma_j^{coll} - \sigma_j^{trans} + s_j > q_k^{lorry}) \\ \vee\ & (v_c \neq j \wedge \sigma_j^{tp} \neq j \wedge \sigma_j^{coll} - \sigma_j^{trans} + s_j + s_{v_c} > q_k^{lorry}) \end{aligned}$$

and 'not enough time' means

$$\sigma_j^{time} + \tau_{jv_c}^{tr} > b_{v_c}, \text{ where } \tau_{hh}^{tr} := 0 \; \forall \; h \in V.$$

The resource window for vertex j at vertex j is $[1, 1]$, and the resource window for a vertex $j' \neq j$ at vertex j is $[0, 1]$.

When the pricing problems are solved by dynamic programming, it is not necessary to introduce more than one intermediate vertex in the LTC subnetworks. It is sufficient to have only one, and to allow multiple visits to this vertex. Visitation counters are maintained only for the customers, not for the transshipment vertices. Cycle elimination can be performed with visitation counters only for the customers, as every possible cycle in the subnetworks (as well as in the network corresponding to the compact formulation) contains a customer. It is then also possible to use a transshipment location more than once, which may be relevant depending on the time windows. This is an additional degree of freedom compared to the compact formulation. Indeed, when the pricing problems are solved by dynamic programming, it would be possible to use only one vertex per transshipment location. Although this would reduce the subnetwork size for the LTC subproblems, the number of possible labels, which determines the difficulty of an (E)SPPRC, would remain the same.

Finally, there is one unconstrained resource r^{cost} measuring the cost. The corresponding resource variable is σ_i^{cost}, and the REF is $f^{r^{cost}}$ with

$$f_{ij}^{r^{cost}}(\sigma_i^{cost}) = \sigma_i^{cost} + \tilde{c}_{ij}^k \quad \forall \; (i,j) \in A, \tag{15}$$

where $\tilde{c}_{ij}^k$ is the reduced cost of traversal of arc (i, j) for vehicle k. Introducing dual variables

$$(\alpha^T, \gamma^T) := (\alpha^1, \ldots, \alpha^{|L_C|}, \gamma^1, \ldots, \gamma^{|F|}) \tag{16}$$

for constraints (9b) and (9c), the objective function of the pricing problem for vehicle k can be written as

$$\sum_{(i,j)\in A^k} c_{ij}^{lorry,k} x_{ij}^k + \sum_{(i,j)\in A_{LT}^k} c_{ij}^{trailer,k} y_{ij}^k - \sum_{l\in L_C} \sum_{i\in V_C^l} \sum_{(h,i)\in A^k} \alpha^l x_{hi}^k - \gamma^k, \tag{17}$$

and the reduced costs of the arcs in the pricing problem for a lorry-trailer combination k can be stated as in Table 2. The reduced cost of the arcs in the pricing problem for a single lorry k are as shown in Table 3.

$\tilde{c}_{ij}^k$	for
$c_{ij}^{lorry,k} - \alpha^l$	$(i,j) \notin A_{LT}^k, j \in V_C^k, loc_j = l$
$c_{ij}^{lorry,k} + c_{ij}^{trailer,k} \max(1 - \sigma_j^{tp}, 0) - \alpha^l$	$(i,j) \in A_{LT}^k, j \in V_{C_{LT}}^k, loc_j = l$

(continued on next page)

(continued from previous page)

$\tilde{c}_{ij}^k$	for
$c_{ij}^{lorry,k} + c_{ij}^{trailer,k}$	$j \in V_{dec}$
$c_{ij}^{lorry,k}$	$j \in V_{trans} \cup V_{coup}$
$c_{id}^{lorry,k} + c_{id}^{trailer,k} - \gamma^k$	$j = d$

Table 2: Reduced cost of arcs in the GTTRP pricing problem for a lorry-trailer combination k

$\tilde{c}_{ij}^k$	for
$c_{ij}^{lorry,k} - \alpha^l$	$(i,j) \in A^k, j \in V_C^k, loc_j = l$
$c_{id}^{lorry,k} - \gamma^k$	$j = d$

Table 3: Reduced cost of arcs in the GTTRP pricing problem for a single lorry k

With the above resource specifications, a feasible label l_1 dominates a feasible label l_2 if and only if

- both reside at the same vertex i,
- $\sigma_i^{tp}(l_1) = \sigma_i^{tp}(l_2)$,
- $\sigma_i^{cost}(l_1) \leqq \sigma_i^{cost}(l_2)$,
- $\sigma_i^{coll}(l_1) \leqq \sigma_i^{coll}(l_2)$,
- $\sigma_i^{trans}(l_1) \leqq \sigma_i^{trans}(l_2)$,
- $\sigma_i^{time}(l_1) \leqq \sigma_i^{time}(l_2)$,
- $\sigma_i^{v_c}(l_1) \geqq \sigma_i^{v_c}(l_2)\ \forall\ v_c \in V_C$,
- and at least one of the above inequalities is strict,

where $\sigma_i^r(l)$ denotes the value of the resource variable σ_i^r for a label l resident at vertex i.

The value of the cost resource of a label l resident at the end depot vertex d indicates the reduced cost of the path represented by the label. The reduced cost of a path p is

$$\tilde{c}_p^k(\alpha,\gamma) = \sum_{(i,j)\in A^k} c_{ij}^{lorry,k} x_{ijp}^k + \sum_{(i,j)\in A_{LT}^k} c_{ij}^{trailer,k} y_{ijp}^k - \sum_{l\in L_C} \left(\sum_{i\in V_C^l} \sum_{(h,i)\in A} x_{hip}^k \right) \alpha^l - \gamma^k. \quad (18)$$

If a path p is feasible and if (18) is negative, a new column corresponding to p can be added to the restricted master problem. The path itself is recursively reconstructed from the l, because each label stores its direct predecessor arc and predecessor label. As explained in Section 4.1, the trailer (sub)path(s) of a path of an LTC lorry is/are also unequivocal and can easily be

reconstructed, so, the objective function coefficient of a new column in the restricted master problem can be determined efficiently, too.

The network D for the compact formulation (1)–(7) contains an arc (o, d) that can be traversed by a trailer without being pulled by its lorry to model the fact that an LTC lorry may also be used without its trailer. This must be taken into account by solving a pricing problem ESPPRC for each LTC, for each single lorry, and for each LTC lorry for which there is no single lorry with identical properties. However, if several vehicles are identical, the resulting symmetry in the problem can be exploited. This can be done as described in [12], p. 84 ff.

5 Computational Experiments

5.1 Algorithms

Extensive studies of possible set-ups for branch-and-price algorithms were performed. In particular, many computational experiments were performed to determine useful strategies for the solution of the pricing problems. The aim was to find successful procedures for the exact as well as the heuristic solution of the pricing problems, and, hence, the overall problem.

For standard VRP(TW)s, it has proved useful for computational purposes to perform heuristic pricing, i.e., to solve the ESPPRC heuristically as long as negative reduced cost paths are found, and to solve the ESPPRC exactly only when heuristic procedures fail to find any more negative reduced cost paths (see, for example, [37]). In view of this, the following exact approach turned out to be acceptable. The pricing problems are solved in three stages:

(i) Initially, the ESPPRC is solved heuristically considering the visitation counters only in the REFs (to ensure that only elementary paths are returned), but completely disregarding the visitation counters in the dominance procedure. In this way, fewer labels are created, because the dominance procedure becomes much stronger (indeed, it becomes too strong, and not all undominated negative reduced cost paths will be found: It is possible that a path is now dominated by another path although both cover different customers). This first stage is performed for each vehicle class on a reduced subnetwork where only the m_s shortest arcs in the forward star of each vertex are considered ($m_s = 3$ for instances with less than 15 customers, and $m_s = 5$ otherwise).

(ii) If no negative reduced cost path is found, the ESPPRC is solved heuristically (i.e., ignoring the visitation counters in the dominance step as before) for each vehicle class on the entire network.

(iii) Only if this also does not yield a negative reduced cost path, the ESPPRC is solved exactly (i.e., with the visitation counter resources of all customers considered in the dominance step) for each vehicle class on the entire network.

By performing each stage for each vehicle class, emphasis is put on quickly returning many negative reduced cost columns. To exactly solve the pricing problems, also bounded bidirectional dynamic programming as described in [38] was used. This yielded moderately better results than the unidirectional approach for the larger instances of the test bed described below. Incremental state space augmentation as described in [38] and [5] was also tried, but did not lead to good results. For more details on experiments with exact algorithms see [13].

The following branching strategy was used:

(i) branch on the number of tours
(ii) branch on an aggregated arc flow variable (i.e., aggregated over all vehicles or vehicle classes) for an arc whose head and/or tail is a customer vertex
(iii) branch on an arc flow variable for a vehicle or vehicle class for an arc whose head and/or tail is a customer vertex

It is sufficient to consider the x_{ij}^k variables for the branching decisions. As described in Section 4.1, once these are all binary, the path variables will be binary as well. It is already sufficient that all x_{ij}^k variables where i or j are customer vertices are integral. This is because each customer is visited exactly once. This also holds if there are identical pricing problems, again because of the constraint that each customer is visited exactly once.

For the solution of the GTTRP with a heuristic approach based on branch-and-price or column generation, also many different settings were investigated. These settings were combinations of heuristic solutions to the pricing problems. Many different heuristic solution strategies for ESPPRCs are described, for example, in [37], p. 207 f. Most of the heuristic strategies tried yielded small computation times, but a quite bad solution quality. The following three strategies (henceforth referred to as H-1, H-2 and H-3) turned out to offer the best trade-off between these two conflicting objectives:

(i) Solve the subproblems by completely disregarding the visitation counter resources for all customers in the dominance procedure, as in the first two stages of the three-stage exact approach.
(ii) Solve the subproblems by disregarding the visitation counter resources for all customers and also ignoring the trailer locations of the labels, i.e., ignoring the requirement that a label l_1 can only dominate a label l_2 if $\sigma_i^{tp}(l_1) = \sigma_i^{tp}(l_2)$.
(iii) Use the same dominance as in (ii), but solve the problem only at the root vertex of the branch-and-bound tree, i.e., perform heuristic column generation. After no more columns are found at the root vertex, solve a set covering problem with the generated columns. If, in the solution of the set covering problem, a customer is visited more than once, keep the customer only in the tour where the saving in tour length obtained by removing the customer from the tour is minimal over all tours where the customer is visited, and remove the customer from all other tours.

All three heuristic approaches also first solved the subproblems on thin graphs, as in the exact approach.

5.2 Test Instances

The application context that motivated this research is, as in several other papers described in the literature review, raw milk collection at farmyards (in this case in southern Bavaria, Germany). The aim was to be able to solve realistic instances. Therefore, the test instances were devised so as to resemble real-world situations with respect to number of customers and transshipment locations, customer supplies, and vehicle costs and capacities.

In practice, the following types of vehicle are used:

- Single lorries:
  - single lorry with two axles and a capacity of 10 tons
  - single lorry with three axles and a capacity of 15 tons
- Lorry-trailer combinations:
  - two-axle lorry with three-axle trailer, with a capacity of 10 and 15 tons, respectively ('2/3-combination')
  - three-axle lorry with two-axle trailer, with a capacity of 15 and 10 tons, respectively ('3/2-combination')

The cost data used are shown in Table 4. These data approximately reflect absolute values of the different cost types for each vehicle type as well as ratios of cost of one vehicle type to another. Two- and three-axle single lorries have the same cost data as two- and three-axle lorries of a lorry-trailer combination.

Cost type → Vehicle type ↓	Fixed	Distance-dependent
Two-axle lorry	18,000	65
Three-axle trailer	2,500	6
Three-axle lorry	20,000	70
Two-axle trailer	2,000	4

Table 4: Cost data for computational experiments

The transfer of milk from one tank is fast, comfortable and clean and requires no specific location for their operation, it is performed by the driver himself without imposing any additional cost. Therefore, load transfer performance only involves time costs.

The customer and transshipment locations were randomly selected on a 100×100 km grid with the depot located in the centre. The resulting Euclidean distances between each pair of

vertices were multiplied by a distance factor of 1.3. The customer supplies were chosen randomly from [1,000; 10,000]. As load transfer time, 2 minutes per 1,000 units of supply were assumed throughout.

The length of the planning horizon was assumed to be 12 hours or 1,320 minutes, respectively. All customers and pure transshipment locations have one time window: [0; 1,320]. This is supposed to model the fact that in the targeted application context, most customers and transshipment locations do not have time windows at all, or at least very long ones (hence, completely ignoring time windows would not be justified). For a code capable of solving problems with time windows, such instances represent the worst case and thus constitute a stress test. It is clear that with tighter time windows, larger instances can be solved.

For each $n \in \{6, \ldots, 10, 20, 25\}$, 30 so-called '$x\_y\_z$'-instances were created with the vehicle, cost and distance data specified above. x stands for the number of lorry customers, y stands for the number of trailer customers, and z stands for the number of pure transshipment locations, and $x = y = z = n$. This means that an x_y_z-instance has $1 + x + y + 3 \cdot (y + z) + 1$ vertices: One for the start depot, x for the lorry customers, y for the trailer customers, three for each transshipment location (of which there are $y+z$) to represent decoupling, transfer, and coupling, and one for the end depot. The following table shows the sizes of the instances and the resulting subnetworks. The column 'No. arcs in VRP' indicates the number of arcs in a 'standard' VRP with as many vertices as the LTC subnetwork in the same column. Considering the number of arcs, a 10_10_10-instance of the GTTRP hence corresponds to a VRP instance with 53 customers.

Instance Type	6_6_6	10_10_10	20_20_20	25_25_25
No. vertices in single lorry subnetworks	14	22	42	52
No. arcs in single lorry subnetworks	157	421	1,641	2,551
No. vertices in LTC subnetworks	50	82	162	202
No. arcs in LTC subnetworks	1,033	2,841	11,281	17,601
No. arcs in VRP	2,353	6,481	25,761	40,201
Arc ratio VRP/GTTRP	2.28	2.28	2.28	2.28

Table 5: Test instance sizes

5.3 Computational Results

The algorithms of Section 5.1 were implemented in C++ with the Visual C++ compiler using the Boost Graph library (BGL, `boost.org`), the ABACUS branch-and-cut-and-price-framework (`www.informatik.uni-koeln.de/abacus`) with CPLEX (`www.ilog.com/products/cplex`) as LP solver and the `r_c_shortest_paths` code from the BGL for the solution of the pricing problems. The tests were performed on a 2.16 GHz processor with 2 GB of main memory under

Windows XP Professional 32 SP 3. The results obtained are shown in the subsequent Tables 6–10. (The 20_20_20 and 25_25_25 instances could be solved only with the H-3 approach.)

The most important observations to be made in Tables 6–10 are:

- Solution quality:
  - In general, the solution quality of all heuristic algorithms is very good.
  - The H-1 approach was able to solve to optimality more than 95 % of the instances for which an optimal solution is known. The instance defining the worst relative solution quality (of 113.7 %) for the H-1 approach is a unique outlier. There were only two instances where the H-1 approach computed a solution that was more than 0.25 % worse than the best solution found with the exact algorithm.
  - The H-2 and H-3 approaches did not find many optimal solutions, but average deviations of 0.3 % and 1.3 % respectively from the solutions found with the exact algorithm are still highly satisfactory.
  - For a considerable percentage of instances, the heuristic approaches found better solutions than the exact one. This apparently paradox result is due to the fact that for such instances, the exact algorithm was not able to explore the branch-and-bound tree deep and wide enough within the available computation time.
  - It is also noteworthy that the average and maximal lengths (in number of arcs) of the tours in the solutions decreased steadily from the exact to the 'most heuristic' algorithm. This implies that the heuristic algorithms were not able to find the rather long tours needed to solve some instances to optimality.
- Computation times:
  - The computation time limits for each algorithm are indicated in the tables. The maximum times reported sometimes exceed these values, because the elapsed time was checked at the end of each pricing step, not within the (E)SPPRC solution algorithm and not with a second thread.
  - The computation times vary widely for all three branch-and-price approaches. For example, the shortest CPU time for the exact algorithm for a 6_6_6-instance was two seconds, the longest was 2,630 seconds (which is more than 1,000 times longer). For H-3, the column generation heuristic, the variation is much smaller.
  - The pricing step is by far the most time consuming part in the algorithms.
  - The number of subproblems to be solved varies widely for the branch-and-price algorithms. The number of (E)SPPRCs to be solved at each subproblem is much less variable. Consequently, the number of subproblems to be solved is the main determinant for the computation time of the branch-and-price algorithms for an instance.

Instance Type	6_6_6	7_7_7	8_8_8	9_9_9	10_10_10
No. instances solved to optimality	30	28	25	12	14
No. instances solved at root	12	12	8	5	7
COW time [s] (Max.: 11,100 s)	2 / 268 / 2,630	5 / 1,689 / 11,500	13 / 5,083 / 11,200	36 / 9,216 / 18,700	868 / 10,341 / 19,900
Pricing time [% COW time]	89 / 98 / 100	94 / 98 / 100	95 / 99 / 100	97 / 100 / 100	98 / 100 / 100
No. subproblems	1 / 97 / 1,535	1 / 136 / 1,565	1 / 825 / 9,021	1 / 202 / 3,943	1 / 293 / 3,503
No. (E)SPPRCs per subproblem	21 / 70 / 232	17 / 64 / 148	4 / 55 / 196	1 / 46 / 180	1 / 60 / 208
No. labels per (E)SPPRC	3,055 / 9,272 / 23,641	5,730 / 17,215 / 97,205	7,003 / 28,558 / 85,050	14,124 / 72,370 / 147,280	42,289 / 97,557 / 246,438
No. generated variables	237 / 1,893 / 17,505	265 / 2,438 / 21,847	306 / 5,285 / 53,453	426 / 1,500 / 13,893	624 / 1,077 / 2,326
Highest level in tree	1 / 8 / 35	1 / 9 / 33	1 / 22 / 161	1 / 17 / 143	1 / 9 / 43
No. tours	2 / 3.1 / 4	3 / 3.8 / 4	3 / 4.1 / 5	3 / 4.2 / 5	4 / 5.1 / 8
No. LTC tours	2 / 2.5 / 3	2 / 3.1 / 4	3 / 3.7 / 5	3 / 3.9 / 5	3 / 4.3 / 6
Longest tour [No. arcs]	6 / 9.0 / 12	7 / 9.0 / 11	7 / 9.8 / 13	8 / 10.7 / 14	8 / 10.3 / 18

Table 6: Computational results for exact algorithm (min. / avg. / max.)

Instance Type	6_6_6	7_7_7	8_8_8	9_9_9	10_10_10
No. instances solved to optimality	29	26	24	12	13
No. instances solved at root	13	12	8	5	6
COW time [s] (Max.: 3,900 s)	1 / 30 / 389	3 / 251 / 2,930	5 / 1,236 / 3,910	13 / 1,750 / 3,900	33 / 2,408 / 13,700
Pricing time [% COW time]	80 / 91 / 98	89 / 93 / 100	81 / 94 / 100	86 / 97 / 100	88 / 97 / 100
No. subproblems	1 / 95 / 2,015	1 / 123 / 1,463	1 / 1,980 / 11,855	1 / 929 / 6,637	1 / 931 / 4,233
No. (E)SPPRCs per subproblem	20 / 61 / 164	21 / 59 / 144	8 / 49 / 148	6 / 46 / 156	2 / 47 / 188
No. labels per (E)SPPRC	1,718 / 4,717 / 10,198	3,792 / 8,317 / 48,782	4,111 / 10,161 / 45,394	7,518 / 15,972 / 36,605	11,237 / 32,925 / 190,515
No. generated variables	235 / 1,693 / 24,983	265 / 2,754 / 36,791	306 / 16,921 / 75,016	426 / 13,689 / 54,831	625 / 8,582 / 34,054
Highest level in tree	1 / 7 / 34	1 / 8 / 33	1 / 20 / 87	1 / 38 / 233	1 / 23 / 203
No. tours	2 / 3.1 / 4	3 / 3.8 / 4	3 / 4.0 / 5	3 / 4.2 / 5	4 / 4.8 / 6
No. LTC tours	2 / 2.5 / 3	2 / 3.1 / 4	3 / 3.7 / 5	3 / 3.9 / 5	3 / 4.5 / 6
Longest tour [No. arcs]	6 / 8.9 / 12	7 / 8.9 / 11	7 / 9.9 / 13	8 / 10.4 / 14	8 / 10.1 / 14

Table 7: Computational results for H-1 algorithm (min. / avg. / max.)

Instance Type	6_6_6	7_7_7	8_8_8	9_9_9	10_10_10
No. instances solved to optimality	0	1	0	2	0
No. instances solved at root	13	11	7	4	4
COW time [s] (Max.: 3,900 s)	0 / 1 / 8	0 / 3 / 31	1 / 7 / 36	1 / 158 / 3,900	2 / 404 / 3,900
Pricing time [% COW time]	30 / 47 / 81	39 / 54 / 94	44 / 60 / 87	4 / 61 / 88	6 / 62 / 98
No. subproblems	1 / 10 / 81	1 / 13 / 79	1 / 35 / 165	1 / 145 / 1,773	1 / 304 / 2,861
No. (E)SPPRCs per subproblem	15 / 57 / 116	15 / 58 / 120	14 / 48 / 152	13 / 44 / 160	12 / 47 / 148
No. labels per (E)SPPRC	640 / 1,107 / 2,854	969 / 1,793 / 8,953	1,361 / 2,218 / 5,968	1,867 / 3,375 / 7,304	2,135 / 7,102 / 51,936
No. generated variables	154 / 395 / 1,825	233 / 476 / 2,577	268 / 813 / 3,525	371 / 2,405 / 17,520	413 / 5,076 / 47,280
Highest level in tree	1 / 4 / 17	1 / 5 / 29	1 / 7 / 18	1 / 17 / 119	1 / 16 / 105
No. tours	1 / 3.0 / 4	2 / 3.7 / 4	3 / 3.9 / 5	3 / 4.0 / 5	4 / 4.7 / 6
No. LTC tours	1 / 2.6 / 4	1 / 3.0 / 4	2 / 3.6 / 5	2 / 3.7 / 5	3 / 4.5 / 6
Longest tour [No. arcs]	4 / 8.6 / 11	7 / 8.4 / 10	7 / 8.9 / 12	8 / 9.3 / 11	8 / 9.3 / 11

Table 8: Computational results for H-2 algorithm (min. / avg. / max.)

Instance Type	6_6_6	7_7_7	8_8_8	9_9_9	10_10_10	20_20_20	25_25_25
No. instances solved to optimality	0	0	0	0	0	?	?
No. instances solved at root	30	30	30	30	30	30	30
COW time [s] (Max.: 3,900 s)	0 / 0 / 1	0 / 1 / 1	0 / 1 / 1	1 / 1 / 3	1 / 2 / 3	16 / 27 / 43	35 / 63 / 107
Pricing time [% COW time]	33 / 53 / 75	39 / 59 / 77	51 / 65 / 76	57 / 70 / 76	60 / 69 / 75	70 / 76 / 83	64 / 79 / 85
No. subproblems	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1
No. (E)SPPRCs per subproblem	68 / 90 / 116	76 / 99 / 124	68 / 108 / 156	84 / 121 / 160	88 / 122 / 184	148 / 182 / 232	164 / 193 / 236
No. labels per (E)SPPRC	714 / 1,055 / 1,908	969 / 1,418 / 2,153	1,173 / 2,113 / 3,615	2,106 / 3,208 / 5,251	2,658 / 3,944 / 6,538	19,549 / 28,090 / 39,953	32,867 / 47,204 / 77,468
No. generated variables	154 / 259 / 438	212 / 291 / 486	226 / 365 / 536	320 / 491 / 805	363 / 560 / 921	1,297 / 1,638 / 2,065	1,558 / 2,128 / 3,428
Highest level in tree	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1
No. tours	2 / 3.1 / 4	3 / 3.9 / 4	3 / 4.1 / 5	3 / 4.3 / 5	4 / 4.9 / 6	8 / 9.3 / 11	11 / 11.8 / 13
No. LTC tours	2 / 2.7 / 4	2 / 3.2 / 4	3 / 3.8 / 5	3 / 3.9 / 5	3 / 4.5 / 6	7 / 8.9 / 10	10 / 11.4 / 13
Longest tour [No. arcs]	6 / 8.8 / 11	6 / 8.3 / 10	7 / 8.9 / 11	8 / 9.4 / 11	8 / 9.3 / 11	9 / 10.3 / 12	9 / 10.0 / 12

Table 9: Computational results for H-3 algorithm (min. / avg. / max.)

	Exact	H-1	Ratio H-1 / Exact	H-2	Ratio H-2 / Exact	H-3	Ratio H-3 / Exact
No. instances solved to optimality	109	104	95.4 %	3	2.8 %	0	0.0 %
No. instances solved at root	44	44	100.0 %	39	88.6 %	n.a.	n.a.
Best relative solution quality			76.3 %		76.5 %		76.6 %
Avg. relative solution quality			99.0 %		100.3 %		101.3 %
Worst relative solution quality			113.7 %		107.5 %		108.3 %
No. times best solution was found	124	140	112.9 %	8	6.5 %	4	3.2 %
COW time [s]	2 / 5,253 / 19,900	1 / 1,135 / 13,700	20.6 %	0 / 115 / 3,900	2.0 %	0 / 1 / 3	0.0 %
Pricing time [% CPU time]	89 / 99 / 100	80 / 94 / 100	94.9 %	4 / 57 / 98	57.6 %	33 / 63 / 77	63.6 %
No. subproblems	1 / 307 / 9,021	1 / 811 / 11,855	264.1 %	1 / 101 / 2,861	32.9 %	1 / 1 / 1	n.a.
No. (E)SPPRCs per subproblem	1 / 59 / 232	2 / 52 / 188	88.1 %	12 / 51 / 160	86.4 %	68 / 108 / 184	183.1 %
No. labels per SPPRC	3,055 / 44,391 / 246,438	1,718 / 14,419 / 190,515	32.5 %	640 / 3,119 / 51,936	7.0 %	714 / 2,348 / 6,538	5.3 %
No. generated variables	237 / 2,438 / 53,453	235 / 8,728 / 75,016	358.4 %	154 / 1,833 / 47,280	75.2 %	154 / 393 / 921	16.1 %
Highest level in tree	1 / 13 / 161	1 / 19 / 233	149.0 %	1 / 10 / 119	74.4 %	1 / 1 / 1	n.a.
No. tours	2 / 4.0 / 8	2 / 4.0 / 6	100.0 %	1 / 3.9 / 6	97.5 %	4 / 4.1 / 6	102.5 %
No. LTC tours	2 / 3.5 / 6	2 / 3.5 / 6	100.0 %	1 / 3.5 / 6	100.0 %	2 / 3.6 / 6	102.9 %
Longest tour [No. arcs]	6 / 9.7 / 18	6 / 9.6 / 14	99.0 %	4 / 8.9 / 12	91.8 %	6 / 8.9 / 11	91.8 %

Table 10: Comparison of computational results over different approaches, instance classes 6_6_6–10_10_10, Exact avg. $\widehat{=}$ 100 %

Instance	No. lorry	No. trailer	Best solution				H-3 solution				Ratio	
number	customers	customers	Cost	Time [s]	No. single lorries	No. LTCs	Cost	Time [s]	No. single lorries	No. LTCs	Cost	Time
1	12	38	557.11	203	1	4	569.97	134	4	3	102.31 %	69 %
2	25	25	608.22	200	1	4	632.03	149	5	3	103.91 %	75 %
3	37	13	618.04	160	2	3	619.74	118	4	3	100.28 %	74 %
6	56	19	930.64	378	4	5	985.01	184	9	3	105.84 %	58 %

Table 11: Computational results for Chao TTRP instances

Overall, the computational results show that the H-3 heuristic approach offers the best trade-off between solution quality and computation time. Evidently, the visitation counters and, even more interestingly, the current trailer location act as fetters in the dominance procedures and slow down the solution progress without significantly improving solution quality.

The computational results obtained are satisfactory insofar as with the H-3 approach, instances with realistic network and cost structure and with realistic size (50 customers and 50 pure transshipment locations) could be solved in short time and with very high solution quality.

The test instances created for this paper use a vehicle cost function containing variable and fixed components. It is noteworthy that tests showed that when the fixed cost component was disregarded, the instances were considerably easier to solve with all four solution approaches: Not only did the overall computation times decrease, but also the difference in computation time between the instance that took longest and the one that took shortest to solve became smaller.

5.4 Computational Experiments with Existing TTRP Benchmark Instances

[8] introduced a set of 21 benchmark instances for the TTRP version he considered (no pure transshipment locations, homogeneous fixed fleet, no time windows, Euclidean distances). The instances range from 38 lorry and 12 trailer customers to 50 lorry and 149 trailer customers. Encouraged by the results obtained with the H-3 approach as well as by the paper by [9], who present a heuristic column generation approach to the heterogeneous fleet VRP and obtain very convincing results on benchmark instances for this problem, it was tried to solve the Chao benchmark instances with the H-3 approach (due to the size of the Chao instances, a solution with the other approaches was not possible).

The H-3 approach allows an unlimited number of vehicles, and it cannot easily be modified so as to use a fixed number. In addition, as explained in Section 3.2, the GTTRP allows that a trailer customer location be used as a transshipment location by more than one lorry-trailer-combination. Hence, the H-3 approach solves a relaxation of the problem studied by Chao.

Unfortunately, the results obtained were rather disappointing. Although all 21 instances could be solved within a few minutes, only the four smallest instances could be solved with the unmodified H-3 approach. To solve the larger instances, it was necessary to limit the number of labels to be created within one run of the SPPRC algorithm and to operate on thin graphs containing only the m_s shortest arcs in each forward star. The results for the four smallest instances were as given in Table 11. The best known solutions are all from [32]. These authors obtained their results on a 1.5 GHz processor. In the above table, the original times were divided by two, yielding a generous lower bound for the computation times had these experiments been performed on the same hardware as the ones described in this paper. In all four solutions, no transshipment location was used by more than one vehicle.

The solution quality for the other Chao instances was very low and decreased with increasing instance size; however, the scaled computation times were always shorter than the ones reported in [32].

The reason for these results is that the average ratio of customer supply to vehicle capacity is much smaller in the Chao instances. This makes the solution of the subproblems much harder, as many more paths are possible, and therefore many more labels have to be created. The approach of solving (E)SPPRCs with a labelling algorithm reaches its limits here.

As mentioned in the literature review, very recently [32] have studied the TTRP version introduced by Chao allowing an unlimited number of vehicles. They used a simulated annealing heuristic and were able to improve the best known solutions for 18 of the 21 Chao benchmark instances. They also created 36 new test instances, all of which had at least 75 customers and at least 18 trailer customers. Due to the unsatisfactory performance of the H-3 approach for the Chao instances, these instances were not tackled.

Considering that for the vehicle routing problem with time windows, '50 clients' means 50 vertices and 2,550 arcs (and a homogeneous fleet), the following quote from [17], p. 492, puts the above computational results in perspective: 'It should be noted that ... column generation has been the dominant approach for the Vehicle Routing Problem with Time Windows (VRPTW). Current branch-and-price algorithms can consistently solve tightly constrained instances (those with narrow time windows) with up to 100 clients. However, they often fail on less constrained instances with only 50 clients.' See also the results obtained by [38] for the capacitated vehicle routing problem (CVRP).

6 Summary and Research Outlook

This paper has studied the generalized truck-and-trailer routing problem (GTTRP) and has presented an arc-flow-based and a path-flow-based formulation for this problem. Moreover, the paper has described an exact solution procedure for the problem, a branch-and-price algorithm, and heuristic versions of this algorithm. A major advantage of the proposed solution approach over a possible branch-and-cut algorithm based on the arc flow formulation is that fleet planning (deciding on the number of vehicles to use of each possible type) and vehicle routing can be performed simultaneously without additional modelling or computational effort.

Extensive computational experiments have been performed. The experiments used randomly generated instances structured to resemble real-world situations and TTRP benchmark instances from the literature. The results showed that with a heuristic column generation approach, real-world GTTRP instances can be solved in short time with high solution quality. However, the results for the TTRP benchmark instances were not so successful due to the low ratio of customer

supplies to vehicle capacity in these instances, which makes the use of labelling algorithms for the solution of the pricing problems highly difficult.

The implementations can still be improved significantly. As the computational experiments showed, the number of subproblems to be solved is the main factor influencing the computation times. Therefore, the most important refinements (which may make the solution of the larger Chao instances and the new instances by [32] possible) are:

- The inclusion of stabilization in the column generation process ([30], pp. 1017 ff.).
- The development of better lower bounding procedures, i.e., the addition of cutting planes. It would be particularly interesting to investigate (the possibility of) generalizing the two-path cuts used for the VRPTW ([27]) to a heterogeneous fleet with trailers.
- The solution of the pricing problems by branch-and-cut ([26]).
- The solution of the pricing problems by (meta)heuristics, and the embedding of the branch-and-price framework into a metaheuristic framework. Recently, [35] have obtained very good results for the VRPTW with a large neighbourhood search algorithm that uses heuristic branch-and-price (heuristic solution of the pricing problem by tabu search, depth-first branching without backtracking) as the reconstruction procedure. This appears to be a very promising approach also for the GTTRP.

The first two points would reduce the number of subproblems to be solved, the last two points would (hopefully) reduce the average time needed for solving one subproblem.

A very interesting and challenging further area of research is the *vehicle routing problem with trailers and transshipments (VRPTT)* ([13]). The GTTRP as described above is a special case of the VRPTT. In the VRPTT, the assumption of a fixed lorry-trailer assignment is abandoned. A trailer may be pulled by any compatible lorry on a part or on the whole of its itinerary, and any lorry may perform a load transfer to any trailer. Moreover, there may also be so-called *support vehicles* (lorries as well as trailers), which are not technically equipped to visit customers. The vehicles equipped to visit customers may use these support vehicles as mobile depots for load transfers (note that all location-routing problems described in [34] are special cases of the VRPTT as shown in [13]). The difficulty with the VRPTT is that, by abandoning the fixed lorry-trailer assignment, the so-called interdependence problem arises, that is, the tours of different vehicles become interdependent: In the simplest case, if a lorry l_1 wants to transfer load into a trailer at a certain transshipment location, the trailer must be at this location before the transfer can start. If a different lorry l_2 pulls this trailer to the location, l_1's tour depends on l_2's tour, and l_1 may have to wait. Moreover, the amount of load that l_1 can transfer is affected by the amount of load other lorries have transferred into the trailer before. Such interdependencies do not usually occur in VRPs studied in the literature and require special solution approaches. It is beyond the scope of this paper to elaborate further on this issue. Suffice it to say that vehicle routing problems with interdependent tours such as the VRPTT are an emerging topic in VRP research and to refer the reader to the recent survey [15].

As can be seen, a lot of interesting research remains to be done, for the GTTRP itself and for its potential extensions and generalizations.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin, "Network Flows", *Prentice-Hall*, Upper Saddle River, 1993.

[2] A. Assad and B. Golden, "Arc Routing Methods and Applications", in: M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8, *Elsevier*, Amsterdam, 1995, pp. 375–483.

[3] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance, "Branch-and-Price: Column Generation for Solving Huge Integer Programs", *Operations Research*, vol. 46, 1998, pp. 316–329.

[4] L. Bodin and L. Levy, "Scheduling of Local Delivery Carrier Routes for the United States Postal Service", in: M. Dror (ed.), Arc Routing: Theory, Solutions, and Applications, *Kluwer*, Boston, 2000, pp. 419–442.

[5] N. Boland, J. Dethridge, and I. Dumitrescu, "Accelerated Label Setting Algorithms for the Elementary Resource Constrained Shortest Path Problem", *Operations Research Letters*, vol. 34, 2006, pp. 58–68.

[6] J. Brunswicker, "Optimale Standort- und Tourenplanung für die Rohmilcherfassung eines Molkereibetriebes", *Lit*, Münster, 1986.

[7] A. Chabrier, "Vehicle Routing Problem with Elementary Shortest Path Based Column Generation", *Computers & Operations Research*, vol. 33, 2006, pp. 2972–2990.

[8] I. Chao, "A Tabu Search Method for the Truck and Trailer Routing Problem", *Computers & Operations Research*, vol. 29, 2002, pp. 33–51.

[9] E. Choi and D. Tcha, "A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem", *Computers & Operations Research*, vol. 34, 2007, pp. 2080–2095.

[10] G. Dantzig and J. Ramser, "The Truck Dispatching Problem", *Management Science*, vol. 6, 1959, pp. 80–91.

[11] M. Daskin, "Network and Discrete Location", *Wiley*, New York, 1995.

[12] G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, F. Soumis, and D. Villeneuve, "A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems", in: T. Crainic and G. Laporte (eds.), Fleet Management and Logistics, *Kluwer*, Boston, 1998, pp. 57–93.

[13] M. Drexl, "On Some Generalized Routing Problems", *Ph. D. Thesis*, Faculty of Business and Economics, RWTH Aachen University, 2007.

[14] M. Drexl, "Branch-and-Price and Heuristic Column Generation for the Generalized Truck-and-Trailer Routing Problem", *Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz*, No. 1102, 2011.

[15] M. Drexl, "Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints", *Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz*, No. 1103, 2011.

[16] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, "An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems", *Networks*, vol. 44, 2004, pp. 216–229.

[17] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and R. Werneck, "Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem", *Mathematical Programming, Series A*, vol. 106, 2006, pp. 491–511.

[18] J. Gerdessen, "Vehicle Routing Problem with Trailers", *European Journal of Operational Research*, vol. 93, 1996, pp. 135–147.

[19] B. Golden, S. Raghavan, and E. Wasil (eds.), "The Vehicle Routing Problem: Latest Advances and New Challenges", Operations Research/Computer Science Interfaces Series, vol. 43, *Springer*, New York, 2008.

[20] A. Hoff, "Heuristics for Rich Vehicle Routing Problems", *Ph. D. Thesis*, Molde University College, 2006.

[21] A. Hoff and A. Løkketangen, "A Tabu Search Approach for Milk Collection in Western Norway Using Trucks and Trailers", *Molde University College*, 2006.

[22] A. Imai, E. Nishimura, and J. Current, "A Lagrangian Relaxation-Based Heuristic for the Vehicle Routing with Full Container Load", *European Journal of Operational Research*, vol. 176, 2007, pp. 87–105.

[23] I. Ioachim, S. Gélinas, F. Soumis, and J. Desrosiers, "A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs", *Networks*, vol. 31, 1998, pp. 193–204.

[24] S. Irnich and G. Desaulniers, "Shortest Path Problems with Resource Constraints", in: G. Desaulniers, J. Desrosiers, and M. Solomon (eds.), Column Generation, *Springer*, New York, 2005, pp. 33–65.

[25] S. Irnich and D. Villeneuve, "The Shortest Path Problem with Resource Constraints and k-Cycle Elimination for $k \geq 3$", *INFORMS Journal on Computing*, vol. 18, 2006, pp. 391–406.

[26] M. Jepsen, B. Petersen, and S. Spoorendonk, "A Branch-and-Cut Algorithm for the Elementary Shortest Path Problem with a Capacity Constraint", *Department of Computer Science, University of Copenhagen*, No. 08/01, 2008.

[27] N. Kohl, J. Desrosiers, O. Madsen, M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows", *Transportation Science*, vol. 33, 1999, pp. 101–116.

[28] G. Laporte, "Location-Routing Problems", in: A. Assad and B. Golden (eds.), Vehicle Routing: Methods and Studies, *Elsevier*, Amsterdam, 1988, pp. 163–197.

[29] G. Laporte, Y. Nobert, and S. Taillefer, "Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems", *Transportation Science*, vol. 22, 1988, pp. 161–172.

[30] M. Lübbecke and J. Desrosiers, "Selected Topics in Column Generation", *Operations Research*, vol. 53, 2005, pp. 1007–1023.

[31] S. Lin, V. Yu, and S. Chou, "Solving the Truck and Trailer Routing Problem Based on a Simulated Annealing Heuristic", *Computers & Operations Research*, vol. 36, 2009, pp. 1683–1692.

[32] S. Lin, V. Yu, and S. Chou, "A Note on the Truck and Trailer Routing Problem", *Expert Systems with Applications*, vol. 37, 2010, pp. 899–903.

[33] G. Nagy and S. Salhi, "Nested Heuristic Methods for the Location-Routeing Problem", *Journal of the Operational Research Society*, vol. 47, 1996, pp. 1166–1174.

[34] G. Nagy and S. Salhi, "Location-Routing: Issues, Models and Methods", *European Journal of Operational Research*, vol. 177, 2007, pp. 649–672.

[35] E. Prescott-Gagnon, G. Desaulniers, and L. Rousseau, "A Branch-and-Price-Based Large Neighborhood Search Algorithm for the Vehicle Routing Problem with Time Windows", *Networks*, vol. 54, 2009, pp. 190–204.

[36] G. Righini and M. Salani, "Symmetry Helps: Bounded Bi-Directional Dynamic Programming for the Elementary Shortest Path Problem with Resource Constraints", *Discrete Optimization*, vol. 3, 2006, pp. 255–273.

[37] S. Røpke, "Heuristic and Exact Algorithms for Vehicle Routing Problems", *Ph. D. Thesis*, Department of Computer Science, University of Copenhagen, 2005.

[38] M. Salani, "Branch-and-Price Algorithms for Vehicle Routing Problems", *Ph. D. Thesis*, Faculty of Mathematical, Physical and Natural Sciences, University of Milan, 2005.

[39] S. Scheuerer, "Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden", *Ph. D. Thesis*, Faculty of Business, Economics and Management Information Systems, University of Regensburg, 2004.

[40] S. Scheuerer, "A Tabu Search Heuristic for the Truck and Trailer Routing Problem", *Computers & Operations Research*, vol. 33, 2006, pp. 894–909.

[41] F. Semet, "A Two-Phase Algorithm for the Partial Accessibility Constrained Vehicle Routing Problem", *Annals of Operations Research*, vol. 61, 1995, pp. 45–65.

[42] F. Semet and E. Taillard, "Solving Real-Life Vehicle Routing Problems Efficiently Using Tabu Search", *Annals of Operations Research*, vol. 41, 1993, pp. 469–488.

[43] P. Toth and D. Vigo (eds.), "The Vehicle Routing Problem", *SIAM Monographs on Discrete Mathematics and Applications*, Philadelphia, 2002.

[44] R. Vahrenkamp, "Transportation Logistic in Rural Setting—The Case of Milk Collection", *Department of Business and Economics, University of Kassel*, No. 5/1989, 1989.

[45] F. Vanderbeck, "On Dantzig-Wolfe-Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm", *Operations Research*, vol. 48, 2000, pp. 111–128.

[46] H. Williams, "Model Building in Mathematical Programming", *Wiley*, Chichester, 1999.