Applications of the Vehicle Routing Problem with Trailers and Transshipments

Technical Report LM-2011-06

Michael Drexl
Chair of Logistics Management, Gutenberg School of Management and Economics,
Johannes Gutenberg University Mainz
and
Fraunhofer Centre for Applied Research on Supply Chain Services SCS

25th November 2011

Abstract

The vehicle routing problem with trailers and transshipments (VRPTT) is a recent and challenging extension of the well-known vehicle routing problem. The VRPTT constitutes an archetypal representative of the class of vehicle routing problems with multiple synchronization constraints (VRPMSs). In addition to the usual task covering constraints, VRPMSs require further synchronization between vehicles, concerning spatial, temporal, and load aspects. VRPMSs possess considerable practical relevance, but limited coverage in the scientific literature. The purpose of the present paper is to describe how several important types of VRPMSs, such as multi-echelon location-routing problems and simultaneous vehicle and crew routing problems, can be modelled as VRPTTs.
Keywords: Vehicle routing problem with trailers and transshipments; Vehicle routing problems with multiple synchronization constraints; Synchronization; Coordination; Models; Applications

1 Introduction

Vehicle routing problems (VRPs) are fundamental planning problems in logistics and transport, and they have been the subject of intensive study for more than half a century now (Toth/ Vigo [31], Golden et al. [12], Laporte [17]). A recent and challenging extension of the VRP is the vehicle routing problem with trailers and transshipments (VRPTT). The VRPTT constitutes an archetypal representative of the class of vehicle routing problems with multiple synchronization constraints (VRPMSs). VRPMSs are a broad class of VRPs which, despite their considerable practical relevance, have attracted comparatively little attention on the part of science so far. In classical VRPs, synchronization between vehicles is necessary only with respect to which vehicle visits which customer. VRPMSs are VRPs which exhibit additional synchronization requirements with regard to spacial, temporal, and load aspects. A recent survey of synchronization in vehicle routing (Drexl [10]) has shown that practical applications of VRPMSs abound and that the solution of several types of VRPMSs is still a research issue.

The fundamental difference between classical VRPs and VRPMSs is that the latter, contrary to the former, feature the so-called *interdependence problem*: In standard VRPs, vehicles are independent of one another in that a change in one route does not affect any other route. In VRPMSs, by contrast, a change in one route may have effects on other routes, due to the additional synchronization requirements. In the worst case, a change in one route may render all other routes infeasible. This has considerable implications on potential solution approaches. In fact, most exact and heuristic algorithms for classical VRPs rely on the fact that routes

are mutually independent. Consequently, these algorithms cannot directly be applied to solve VRPMSs.
A first step toward *solving* problems is properly *modelling* them. Therefore, the contribution of the present paper, similar to the works of Noon/Bean [21], Crainic et al. [6], and Baldacci et al. [1] for other routing problems, is to propose the VRPTT as a unified modelling tool for VRPMSs in general. To this end, it is described how several important types of VRPMSs can be represented as VRPTTs. This demonstrates the versatility of the VRPTT as a representational framework for many types of rich VRPs and points out that the VRPTT needs and deserves further study. The development of exact or heuristic solution algorithms for the VRPTT is beyond the scope of the paper.
The next section describes the VRPTT and develops a graph-theoretic modelling framework which can serve as a basis for algorithmic solution approaches. Subsequent sections describe transformations of classical VRPs and of several types of VRPMSs that were identified as particularly relevant and challenging in the above-mentioned survey. The paper ends with a short conclusion.

2 The vehicle routing problem with trailers and transshipments

2.1 Problem description

The VRPTT is a real-world problem arising in raw milk collection at farmyards (Drexl [9]). Basically, it can be described as follows. There is a set of customers with a given supply. To collect the supply, a set of heterogeneous vehicles stationed at one or several depots is available. In addition to potentially unequal costs, capacity, and temporal availability, the vehicles differ with respect to two orthogonal criteria: First, there are *autonomous vehicles* able to move in time and space on their own (lorries) and *non-autonomous vehicles*, which can move in time on their own, but must be pulled by a compatible autonomous vehicle to move in space (trailers). (Note that autonomy depends on the application context. The same real-world object may be considered autonomous in one application and non-autonomous in another.) Second, there are *task vehicles*, which are technically equipped to visit customers and collect supply, and there are *support vehicles*, which cannot visit customers, but can be used as mobile depots to which the task vehicles transfer load. The load transfers are carried out at *transshipment locations* (TLs) such as parking places or customer premises. Using such locations may incur one-time fixed costs as well as fixed costs per transshipment operation. The number of vehicles allowed to use a certain TL during the planning horizon and the number of transshipment operations at a certain TL during the planning horizon may be limited. There may also be limits on the number of vehicles allowed to be present and/or the number of transshipment operations allowed to be performed simultaneously at a TL. Moreover, single or multiple time windows are associated with the customers as well as with the TLs.
Some customers can only be visited by a lorry without a trailer and are hence called *lorry customers*. The other customers can be visited by a lorry with or without a trailer and are called *trailer customers*. Generally speaking, the vehicles are subject to *accessibility constraints*, that is, not every vehicle is necessarily able to visit every location, due to physical restrictions (e.g., limited manoeuvring space) or technical constraints (e.g., lack of pertinent equipment to perform a service). There is no fixed assignment of a trailer to a lorry. Any non-autonomous vehicle may be pulled, on the whole or on a part of its itinerary, by any compatible autonomous vehicle. What is more, any task or support vehicle is permitted to transfer its load partially or completely to any other compatible task or support vehicle at any TL arbitrarily often. Vehicles need not carry any load when returning to the depot. Lorries need not bring back a trailer, neither one they might have pulled when leaving the depot, nor any other. Not all vehicles have to be used.

The problem is to determine routes for lorries *and routes for trailers* so that total costs are minimized, the complete supply of all customers is collected and delivered to a depot, and loading capacities, accessibility constraints, and time windows are maintained. Moreover, it must be ensured that the routes are synchronized with respect to space, time and load, so that non-autonomous vehicles move in space only when accompanied by compatible other vehicles and that the vehicles involved in a load transfer operation visit the pertinent location at the right time and transfer and receive the right amount of load. The decisive point is that in a solution, a route is determined for each vehicle which is actually used, be it a lorry or a trailer, an autonomous or a non-autonomous one. An example route plan, which, for simplicity, does not contain support vehicles, is depicted in Figure 1.

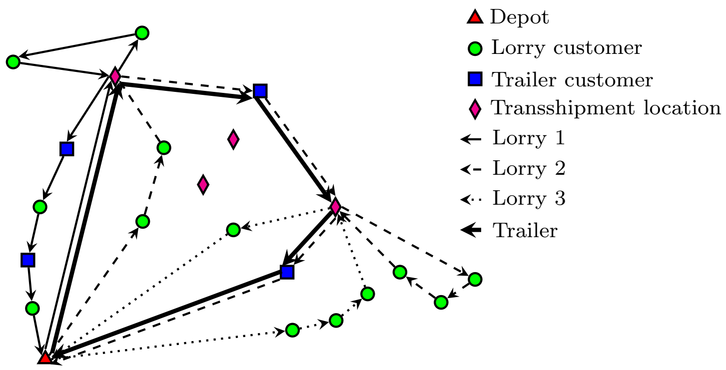

Figure 1: VRPTT example route plan

In the example, lorry 1 couples the trailer at the depot and goes to a TL, where it decouples the trailer. Lorry 1 then visits two lorry customers, returns to the trailer, transfers some load, leaves the trailer at the TL and returns to the depot via two lorry and two trailer customers. Lorry 2 starts at the depot and visits two lorry customers before coupling the trailer (after lorry 1 has performed its load transfer). Lorry 2 then visits a trailer customer, decouples the trailer at another TL, possibly performs a load transfer, visits some lorry customers, returns to the trailer, re-couples it and pulls it back to the depot via a trailer customer. Lorry 3 also starts at the depot, visits some lorry customers, and transfers some load to the trailer while lorry 2 is visiting the three rightmost lorry customers. After that, lorry 3 returns to the depot via another lorry customer. The two TLs in the centre of the figure are not used.
The central question in the VRPTT can be stated as:

Which vehicle transfers how much load when where into which other vehicle?

A transshipment or (de)coupling operation is defined by:

- The *location* where the operation takes place
- The *point in time* when the operation begins
- The *passive vehicle*, which provides capacity, that is, may receive load, and/or may be coupled to or decoupled from another vehicle
- The *active vehicle*, which requests capacity, that is, may transfer load, and/or may couple or decouple another vehicle
- The *amount of load* transferred into the passive vehicle, which is zero if only coupling or decoupling is performed, and which is negative if the passive vehicle supplies load for the active one

Hence, the decisive modelling issues resulting from the central question are the following:

- How to ensure that a trailer is accompanied by a compatible lorry on an arc, that is, how to synchronize the movements of vehicles?
- How to synchronize the visiting times of vehicles at transshipment locations?
- How to balance the load transfer amounts of vehicles exchanging load?

2.2 A network representation of the VRPTT

For a particular vehicle routing problem, there is a broad spectrum of options how to represent the information and data on the relevant objects and their relationships, such as the three synchronization issues just mentioned. These options range from 'model the underlying problem logic completely by means of decision variables and constraints' to 'create a highly involved network that by itself ensures feasibility and synchronization'. There is no silver bullet; it depends on the concrete problem type and the average instance data where a model for a certain VRP should be positioned on this continuum. In what follows, a descriptive, graph-based modelling framework is presented, which can serve to represent and model VRPTTs (and, as will be demonstrated, other important VRPMSs), and which can form the basis for a concrete solution approach, be it a branch-cut-price algorithm using a mixed integer programming formulation (Desaulniers et al. [7]), a constraint programming method (Rousseau et al. [27]), or a (meta-)heuristic using construction and improvement procedures with local and large neighbourhood search (Gendreau/Potvin [11]).

2.2.1 The basic modelling principles

In practice, that is, on real road networks, the distinction between autonomous and non-autonomous vehicles is sharp: Each vehicle is either of the one or the other type. The same holds for the distinction between task and support vehicles. As will be shown, however, for modelling purposes using graphs, it is sensible to take a more flexible approach. Essentially, *three fundamental modelling principles* will be employed:

- An adequate definition of *subnetworks for vehicles*, that is, the definition of which vertices and arcs may be visited and traversed by a certain vehicle or vehicle class with or without other vehicles (this means that (non-)autonomy is not necessarily a global, network-wide property of a vehicle, but a local, arc-specific one)
- The specification of *compatibilities between vehicles* with respect to the ability of common movement (formation of autonomous *composite* vehicles) and the ability to perform load transfers
- The setting of *limits on the amount of load transferred or received* by a particular vehicle at a particular location

Using these ideas, the VRPTT can be defined on a directed network $D = (V, A)$ with an associated set F of vehicles (the fleet).

2.2.2 The network

The vertex set V is partitioned into *four subsets*: V_{VD}, the *set of virtual depot vertices*, V_D, the *set of real depot vertices*, V_C, the *set of customer vertices*, and V_T, the *set of transshipment vertices.* $l(i)$ denotes the real-world location corresponding to i for each $i \in V \setminus V_{VD}$. $V_{VD} = \{s, e\}$, where s (e) is the virtual start (end) depot vertex. Both have no corresponding real-world location. $V_D = V_{SD} \cup V_{ED}$, where V_{SD} (V_{ED}) is the set of start (end) depot vertices.

Each vertex $i \in V$ has a time window $[a_i, b_i]$ indicating the earliest and latest point in time for the beginning of an operation to be performed at i. This time window may have zero length ($a_i = b_i$), that is, represent a concrete point in time. The time window of each $i \in V \setminus V_{VD}$ is completely contained in one of $l(i)$'s time windows. Each $i \in V_C$ has an associated supply s_i. $F_i \subseteq F$ is the set of vehicles allowed to visit $i \in V$.

2.2.3 The fleet

The fleet F is partitioned into K *classes or vehicle types.* The vehicles in a class are identical with respect to all relevant attributes, such as those mentioned at the beginning of Section 2.1. In particular, they have the same potential start and end depots. All vehicles are initially stationed at the virtual start depot vertex s and must end their routes at the virtual end depot vertex e.

Each vehicle which is used moves from s to some vertex in the set of its potential start depot vertices, V^k_{SD}, at the beginning of its route, and from some vertex in the set of its potential end depot vertices, V^k_{ED}, to e at the end of its route. Unused vehicles move directly from s to e. If the start and/or end depot of a vehicle k is/are known in advance, V^k_{SD} and/or V^k_{ED} contain only one element. If the single-depot case is considered, that is, if each vehicle is assigned to one start and one end depot ex ante, the virtual depot vertices s and e are not necessary. q^k indicates the loading capacity of vehicle or vehicle class k. $F^k_{trans,comp}$ is the set of vehicles with which vehicle k can exchange load.

2.2.4 The subnetworks

$V^k \subseteq V$ ($A^k \subseteq A$) is the subset of vertices (arcs) that vehicles of class k are allowed to visit (traverse). Consequently, for each vehicle class k, there is a subnetwork $D^k = (V^k, A^k)$ describing k's possible movements. For all $k \in F$, A^k_{single} is the set of arcs k is able to traverse without being accompanied by another vehicle. $F^{k,(i,j)}_{move,comp}$ is the set of vehicles with which vehicle $k \in F$ can move along arc $(i,j) \in A^k \setminus A^k_{single}$, that is, for a lorry k, the set of trailers k can pull along (i,j), and for a trailer k', the set of lorries which can pull k' along (i,j). $V^k_{C,0}$ is the set of customer vertices a task vehicle is allowed to visit without being allowed to collect any load there.
Moreover, with each arc $(i,j) \in A^k$ are associated vehicle-dependent costs c^k_{ij} and a vehicle-dependent travel time t^k_{ij}. The usual cost types are fixed, distance-, time-, and stop-dependent costs. Fixed vehicle costs, that is, one-time costs incurred for making a vehicle available, are considered on the arcs leading from e to real start depot vertices. Distance- and stop-dependent costs are considered on the respective arcs. Since waiting times are possible in the presence of time windows, time-dependent costs must be considered globally in an objective function by measuring overall route duration, which may be more than the sum of travel and service times.
Each subnetwork for a vehicle $k \in F$ contains only vertices i with $k \in F_i$, that is, $V^k = \{i \in V : k \in F_i\}$. Lorries cannot enter the start and end depot vertices of other lorries; trailers cannot enter any start or end depot vertices other than their own. Unless otherwise specified for certain modelling aspects, each subnetwork contains an arc between each pair of vertices with the following exceptions:

- No arcs enter (leave) the virtual start (end) depot vertex.
- k's real start (end) depot vertex/vertices can be reached only from s (left only to e).
- There are no arcs from trailer start depot vertices to lorry customer vertices and from lorry customer vertices to trailer end depot vertices.
- Of course, there are also no arcs (i,j) where $a_i + t^k_{ij} > b_j$.

2.2.5 Modelling transshipments

In real-world VRPTTs, the following restrictions on transshipments are possible at any potential transshipment location l:

- A limit on
  - the number of times a vehicle is allowed to visit l
  - the number of different vehicles allowed to visit l
  - the number of different vehicles allowed to transfer or receive load at l, that is, the number of potential active and passive vehicles
  - the number of transshipment operations performed simultaneously at l
  - the overall number of transshipment operations performed at l
  - the total amount of load transferred at l
- Intervals within which
  - the amount of load that a vehicle transfers or receives at l during one transshipment operation and overall must lie
  - the overall amount of load transferred at l must lie

This information must be captured by parameters and considered in a concrete model and solution approach, for example by pertinent constraints in a mixed integer program. Details on the practical relevance of these parameters in real-world VRPTTs are provided by Drexl [9].

The definitions given in Sections 2.2.2–2.2.4 deliberately leave some freedom regarding the precise design of the network for a particular problem. To illustrate this scope for modelling transshipments, consider the following two extreme situations: It can be decided to create one vertex for each time window of each physical location and to allow that all vehicles visit all transshipment vertices more than once and transfer or receive arbitrary amounts of load. Then, in addition to the routing decisions for each vehicle, the following decisions must be modelled 'outside' the network, that is, for example, by decision variables in a mixed integer program: The maximal number of visits of each vehicle at each vertex, the point in time when each visit of each vehicle takes place, and the amount of load transferred or received during each visit. This corresponds to the approach 'model the problem logic by means of decision variables and constraints' mentioned at the beginning of Section 2.2.

Alternatively, it can be decided to define a fixed-schedule space-time-operation-vehicle network in which each transshipment vertex corresponds to a concrete location, a concrete point in time when the transshipment starts, a concrete passive vehicle, and a concrete load transfer amount. Then, there are no decisions to take in addition to the routing decisions for each vehicle. This corresponds to the approach 'create a network that by itself ensures feasibility and synchronization'. In this way, a specific configuration of the network lies the foundation for an optimization model.

The decision how to configure the network has to take the concrete aspects of a specific application into account, and different set-ups are possible. The necessity to find a problem-adequate modelling configuration is a constitutive property of VRPTTs and VRPMSs in general. However, the basic ideas of subnetworks and compatibility specifications allow the consideration of a very broad range of VRPTT variants. In the following sections, it will be shown how other important basic classes of VRPMSs can be represented with the described VRPTT modelling framework.

3 Extensions of the VRP

It is evident that 'standard' vehicle routing problems without multiple synchronization constraints, such as the capacitated VRP and the VRP with time windows, are special cases of the VRPTT. By its definition, the VRPTT already encompasses the heterogeneous fleet versions of these problems. Real-world constraints such as multiple capacity constraints, vehicle-specific time windows etc. are also easily accommodated.

Moreover, the concepts of subnetworks and compatibilities between vehicles with respect to movement and load transfer allow modelling multiple-depot VRPs, VRPs with multiple use of vehicles, VRPs with multiple planning periods, the generalized VRP (GVRP), the open VRP, capacitated arc routing problems (CARPs), and the truck-and-trailer routing problem (TTRP). This is explained in the following.

- The *multiple-depot VRP* (Cordeau et al. [5]), where, for each vehicle, one out of several potential real depots must be selected as start and end depot, can be modelled with the VRPTT framework as follows: For each 'real' vehicle (lorry) k, a virtual vehicle (trailer) k^v is introduced. k^v is non-autonomous and able to leave the virtual start depot vertex s only together with k, even along the arc (s, e) for unused vehicles. k^v can only visit s, e, and k's real start and end depot vertices, that is, the vertices in V^k_{SD} and V^k_{ED}. From a real start depot vertex, a virtual vehicle must move directly to the corresponding real end depot vertex, and it is able to do so without being accompanied by its associated real vehicle. To reach e, k^v must again be pulled by k. In this way, it is ensured that the real vehicle, at the end of its route, will visit the correct real end depot vertex.

- *VRPs with multiple use of vehicles* (Taillard et al. [30]), where one vehicle may perform several routes starting and ending at a certain depot, can also be modelled. To this end, transshipment vertices for each real depot are introduced. Virtual vehicles as defined in the previous paragraph are allowed to move on their own from the start depot vertices to these special transshipment vertices and between the latter, but must be accompanied by their associated real vehicles when moving from these transshipment vertices to the corresponding real end depot vertex. The virtual vehicles are uncapacitated. Real vehicles can only transfer load into their corresponding virtual vehicle, and virtual vehicles can only receive load from their corresponding real vehicle. Thus, the real vehicles will unload completely at one of the transshipment vertices and/or the selected real end depot vertex.
- Of course, these special transshipment vertices may have associated time windows representing multiple periods (for example, for each depot, there may be one such vertex per weekday). In this way, *multiple-period VRPs* (Zäpfel/Bögl [32]) can be modelled.
- In the *generalized VRP* (Baldacci et al. [1]), the set of customers is partitioned into a set of clusters, and instead of requiring that each customer be visited exactly once, it is stipulated that exactly one customer from each cluster be visited exactly once. Since the VRPTT as modelled above already covers the case where there are multiple vertices for one customer and exactly one of these vertices must be visited, the GVRP can be modelled as a VRPTT, too.
- Also the (single- as well as multiple-depot) *open VRP* (Brandão [2]), where the routes of the vehicles need not end where they started, can easily be handled with the above model: It is sufficient to remove all real end depot vertices and introduce an arc with zero costs and travel time from each customer or transshipment vertex directly to the virtual end depot vertex e.
- In addition, it is well known that *capacitated arc routing problems* (Golden/Wong [13]) can be transformed into vehicle routing problems (Pearn et al. [24], Longo et al. [19]); consequently, it is possible to model CARPs as VRPTTs.
- Finally, the *truck-and-trailer routing problem* (Semet/Taillard [29], Chao [4], Scheuerer [28]) is a rather well-studied vehicle routing problem which, as its name implies, also considers trailers, and which can be modelled as a VRPTT. In fact, the TTRP is a special case of the VRPTT where there is a fixed lorry-trailer assignment. This means that each trailer can be pulled by a unique associated lorry, and only this lorry is permitted to transfer load into the trailer.

4 N-echelon vehicle and location-routing problems

Gonzalez Feliu et al. [15] and Perboli et al. [25] formally introduce the class of *multi-echelon* (or *N-echelon*) *vehicle routing problems* and are the first to use these terms. The basic idea behind this problem class is that customers are not delivered directly from a central depot, but via N legs in an N-stage distribution network. An N-stage distribution network contains $N+1$ levels of location. Echelon $n \in \{1, \dots, N\}$ considers transports from location level $n-1$ to n, see Figure 2. For each echelon n, there are dedicated vehicles which can only visit the locations or *facilities* defining echelon n. This means that only the vehicles of echelon N are task vehicles, that is, are allowed to visit customers; all other vehicles are support vehicles. Load transfers are only possible between vehicles of different echelons. The difference to distribution network design problems lies in the fact that for each vehicle in the problem, a complete route is computed. Gonzalez-Feliu [14] studies the general N-echelon *location*-routing problem (LRP). The difference between the N-echelon vehicle routing problem and the N-echelon location-routing problem is that the latter considers fixed costs for opening a facility, contrary to the former.

There are many different variants of N-echelon routing problems. Subsequently, it is first described how a basic problem can be modelled as a VRPTT; extensions are considered afterwards. For simplicity, the description is based on the 2-echelon LRP, but the elaborations generalize directly to arbitrary $N \in \mathbb{N}$.

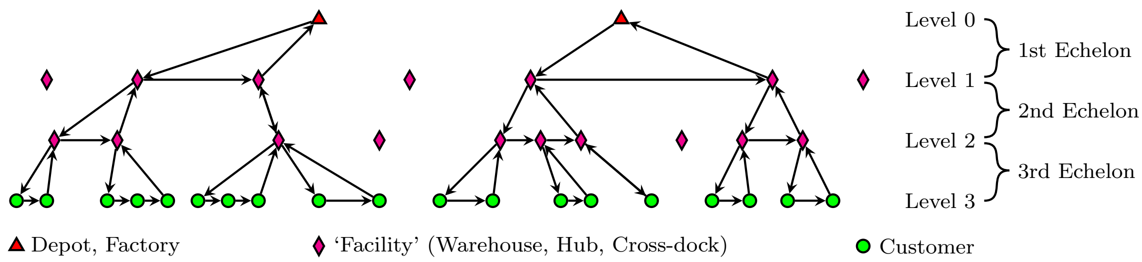

Figure 2: Example of a 3-echelon routing problem

The basic ideas are that (i) virtual depots, real depots, and customers retain their meaning, whereas transshipment locations are used to model facilities, and that (ii) echelons are represented by an adequate definition of subnetworks for support and task vehicles.
More precisely, if there are n_0 and n_1 potential facilities of level 0 and 1 respectively, each facility is represented by one transshipment vertex $t^0_1, t^0_2, \dots, t^0_{n^0}$ and $t^1_1, t^1_2, \dots, t^1_{n^1}$. If no temporal aspects are considered, it is sufficient to have only one vertex per facility/transshipment location. To define the subnetworks, for each support vehicle class, there is one support vehicle for the first echelon, and for each task vehicle class, there is one task vehicle for the second echelon. Support vehicles may only move between depots and facilities of the first echelon. Task vehicles may only move between depots and vertices of the second echelon.
For the facilities of level 0, it is sufficient to define one virtual support vehicle, say, k^0. The route of k^0 only defines which level-0 facilities to open. Therefore, to avoid symmetries, it is sufficient to define the subnetwork for k^0 with arcs from s to each level-0 facility, from each level-0 facility to e, and arcs (t^0_1, t^0_2), $(t^0_1, t^0_3) \dots (t^0_1, t^0_{n^0})$, (t^0_2, t^0_3), $(t^0_2, t^0_4) \dots (t^0_2, t^0_{n^0}) \dots$ between level-0 facilities. k^0 is uncapacitated and is hence able to deliver the necessary amount of load from the virtual depot to each of the selected level-0 facilities.
A support vehicle serving the first echelon, that is, moving between the facilities of levels 0 and 1, starts its route at the virtual start depot, visits its selected real depot, then necessarily one of the open facilities of level 0 (since the vehicle is initially empty and must load before visiting a level-1 facility), then one or more level-1 facilities, then again a level-0 facility etc. Finally, it visits its assigned end depot and returns to the virtual end depot. A customer vehicle operates analogously between level-1 facilities and customers.
The following extensions occur in practice:

- Facilities with limited capacity
- The restriction that, on echelon n, only one out of several potential vehicles is allowed to actually visit a facility of level n
- A fixed assignment of vehicles to facilities, that is, each vehicle on echelon n is permitted to visit only one particular facility of level $n-1$ or, for echelons $n = 1, \dots, N-1$, each vehicle on echelon n is permitted to visit only one particular facility of level n
- A fixed assignment of facilities of level $n = 1, \dots, N$ to facilities of level $n-1$, that is, the requirement that a certain facility be served by a vehicle assigned to one particular other facility
- Time windows at exactly one echelon (without the requirement of temporal synchronization of vehicles of different echelons)
- Necessity of temporal synchronization of vehicles of different echelons (this includes time windows at facilities of all echelons)

Ways to handle these extensions are described in what follows.
Capacitated facilities l can be modelled by setting the allowable interval for the overall amount of load transferred to $[0, q_l]$ for all facility vertices $i \in V_T$, where q_l is the capacity of facility l and $l(i) = l$.

The requirement that only one vehicle be allowed to visit a level-n facility i can be modelled by setting the corresponding parameter for the number of different vehicles which may deliver load at i to one (see Section 2.2.5).
If sufficiently many vehicles of each class are available, a fixed assignment of vehicles of echelon n to facilities of level $n-1$ can be modelled by introducing, for each potential facility l of level $n-1$, one vehicle of each class. This vehicle is capable of visiting, on echelon n, only the vertex corresponding to l. If the number of vehicles is limited, such a fixed assignment can be modelled as follows: All vehicles (lorries) of echelon n are non-autonomous and are considered to have a load transfer amount of zero at all vertices (by setting the intervals for the amount of load each vehicle transfers or receives at any facility of level $n-1$ to $[0, 0]$). For each combination of potential facility l of level $n-1$ and lorry k of echelon n, there is a virtual non-autonomous trailer $k'_{l,k}$ which is permitted to visit, on level $n-1$, only the vertex corresponding to l and which can only be pulled by k. $k'_{l,k}$ might thus be called 'facility assignment trailer'. $k'_{l,k}$ has the same capacity as k and is allowed to leave the virtual start depot vertex s only together with k (except for the arc (s, e) for unused vehicles). Now, if k is to use only facility l on level $n-1$, k couples the corresponding facility assignment trailer at the virtual start depot vertex s, and keeps it coupled for the complete route. In this manner, the two vehicles can only visit the vertex corresponding to l on level $n-1$. Since k cannot receive or transfer any load, it will never decouple $k'_{l,k}$ en route. The case where each vehicle on echelon n may visit only one facility of level n is analogous.
An ex-ante defined fixed assignment of a facility of level $n = 1, \ldots, N$ (regarding customers as 'level-N facilities') to a facility of level $n-1$ can also be modelled by facility assignment trailers. For example, if facility l_n must be served by a vehicle stationed at facility l_{n-1}, only facility assignment trailers assigned to l_{n-1} are allowed to transfer a positive amount of load at l_n.
A combination of such assignment requirements may make it necessary that a lorry pulls more than one trailer at a time. This is possible and is described in more detail in Section 7.
Time windows on one echelon can be considered if one vehicle is introduced for each real vehicle of the respective echelon.
By introducing one vehicle for each real vehicle of each echelon, temporal synchronization of vehicles of the first and second echelon (including the consideration of time windows at facilities of both echelons) can be modelled. Load transfers are then handled as described in the section on the VRPTT. The issues at transshipment locations with respect to how to deal with the fact that a vehicle may want to transfer load to another vehicle at the same location more than once etc. are exactly the same as for VRPTTs.
An advantage of modelling N-echelon routing problems as VRPTTs lies in the fact that, contrary to existing models, non-autonomous objects such as trailers and swap-body platforms can be considered. To this author's experience, these are used in many real-world applications of N-echelon LRPs, but no pertinent scientific publications are known.
Classical location-routing problems as described in Nagy/Salhi [20] are special cases of N-echelon LRPs with $N = 1$. In addition, Nagy/Salhi [20] describe several applications of multi-echelon LRPs where, on one or more echelons, no route planning is required, but only a selection of the facilities to be used and an assignment of these facilities to those of the next highest echelon is sought. All such problems are also special cases of the N-echelon LRP, since they can be modelled using facility assignment trailers. Besides, the classical location-routing problem with uncapacitated facilities, a fixed assignment of vehicles to facilities, and no time windows can be modelled as a VRPTT with one lorry and one trailer, as described in Drexl [9].

5 Simultaneous vehicle and crew routing and scheduling problems

For the most part, the VRP literature does not distinguish between a vehicle and its driver. In their monograph on VRPs, for example, Toth/Vigo [31] state that throughout, 'the constraints imposed on drivers are imbedded in those associated with the corresponding vehicles'. However, it is a fact that drivers regularly need breaks and rests and must obey the existing pertinent social legislation and trade union rules regarding driving, break, and rest times. On the other hand, vehicles can essentially be used twenty-four hours a day. Consequently, considering a vehicle and a driver a fixed unit inevitably leads to a suboptimal temporal utilization of vehicles. Simultaneous vehicle and crew routing and scheduling problems (SVCRSPs) (Hollis et al. [16]) are concerned with the situation where the required tasks have no given timetable/no fixed schedule, and where a driver-vehicle combination is not considered an inseparable unit anymore, so that routes have to be planned for both vehicles and drivers. For such problems to make sense, there must be a set of locations where drivers can change vehicles and vice versa (*relay stations*).

Essentially, SVCRSPs are VRPTTs with the sole specific characteristic that all vehicles are non-autonomous. More precisely, such problems can be modelled, or rather, interpreted, as VRPTTs in the following way: For each driver and each vehicle, there is one non-autonomous vehicle. The vehicles corresponding to drivers have zero capacity. The relay stations correspond to the transshipment locations. At the transshipment locations/relay stations, a driver (vehicle) can 'uncouple' a vehicle (driver) and couple a different one. Transshipments of load between vehicles may be allowed or not. In the latter case, the interval for the overall amount of load transferred is simply set to $[0, 0]$ at all transshipment vertices $i \in V_T$. If transshipments are allowed, there may also be support vehicles, which are non-autonomous as well.

6 Personnel dispatching problems with spatio-temporal synchronization constraints

There is quite a number of applications where persons must be synchronized with respect to space and time to perform some kind of service, but where no load transfers are performed and no vehicle capacities are relevant. Examples include the dispatching of service technicians with different qualifications who have to meet to repair machines, or homecare staff routing, where two nurses must visit a disabled person at the same time for lifting purposes or with a specified delay to apply medicine after a meal etc. (see, for example, Li et al. [18], Bredström/Rönnqvist [3], Dohn et al. [8]).

Such problems can be represented as VRPTTs in the following way. Each vehicle corresponds to a person, and all vehicles are uncapacitated task vehicles. There are no transshipment locations, or, put differently, the sets of customer and transshipment vertices happen to coincide. Qualifications of different staff members for certain types of service are represented by appropriate accessibility constraints at vertices. If two persons with two different qualifications u_1, u_2 must be present at a location at the same time to perform a task, this is represented by two customer vertices i and j, where i (j) may be called task entry (task exit) vertex. All vehicles corresponding to persons with qualification u_1 or u_2 can reach i. The only arc leaving i is the arc (i, j). All vehicles are non-autonomous along (i, j). To move along (i, j), a vehicle corresponding to a person with qualification u_1 must be accompanied by a vehicle corresponding to a person with qualification u_2 and vice versa. The arc (i, j) is also the only arc entering j. From j, all vehicles can move on their own to other task entry or end depot vertices. The parameter for the number of allowed visits at i by all vehicles altogether is set to two. In this way, it is ensured that exactly two compatible vehicles (qualified persons) meet at the task location at exactly the same time.

If one visit must precede the other, it is sufficient to create two customer vertices i and j for the task, to set the two parameters for the number of allowed visits at i and j by all vehicles altogether to one, and to allow that i (j) can only be reached by vehicles corresponding to persons with qualification u_1 (u_2). No restrictions with respect to entering and leaving arcs apply in this case. Temporal relationships between the visiting times at i and j can then be established as described for transshipments in Section 2.2, that is, by appropriate constraints on visiting times or by creating fixed-schedule space-time-operation-vehicle networks.

7 Problems with more than two types of vehicle

After the preceding explanations, it is easy to imagine the situation where more than two types of vehicle may be allowed or required to join to move in space. To be precise, two aspects must be distinguished: It is possible that (i) an autonomous elementary or composite object can 'pull' more than one elementary non-autonomous object, and that (ii) two or more non-autonomous objects must join to form an autonomous object. These extensions are of practical as well as theoretical relevance.

A practical example for a VRPTT featuring the first case is road transport of agricultural products with special vehicles, where it is allowed that one tractor pulls two trailers at the same time. A practical example for the second case is when drivers and lorries are separate planning entities as described in Section 5. Then, a trailer must join with a driver and a lorry to move in space.

Furthermore, in the simultaneous vehicle and crew routing and scheduling problems described in Section 5, it is possible to consider driver shuttle transports, where small shuttle vans are used to transport lorry drivers between relay stations. Doing so increases the flexibility for matching lorries and drivers, since a lorry driver is enabled to leave a lorry at a relay station and subsequently drive a lorry starting from a different relay station. Each shuttle van is essentially an autonomous support vehicle with capacity zero which can 'couple' zero or more drivers and is not compatible with task vehicles.

Such situations are represented by the model developed in Section 2.2 as follows. Remember that $F^{k,(i,j)}_{move,comp}$ was defined as the set of vehicles with which vehicle k can move along arc (i, j): For a lorry k, it is the set of trailers k can pull, and for a trailer k', it is the set of lorries which can pull k'. The definition of this set does not restrict the number of trailers (non-autonomous vehicles) a lorry (autonomous vehicle) may pull at the same time, so that the above case (i) is covered. Moreover, the 'set of vehicles' capable of moving k along an arc may of course contain elementary as well as composite vehicles, so that case (ii) is also covered.

A very important point is that the number of synchronization requirements grows only linearly when more than two types of vehicle are considered: If, for example, an autonomous lorry k is supposed to pull the two trailers k' and k'' along an arc (i, j), it is sufficient to synchronize k with k' and with k''. This will ensure that k' and k'' are also synchronized with each other.

An important application which can be modelled by using more than two types of vehicle is described in the next section.

8 Pickup-and-delivery problems with (trailers and) transshipments

A popular extension of the classical VRP is the class of vehicle routing problems with pickups and deliveries, or *pickup-and-delivery problems (PDPs).* In a classical VRP, either each customer has a certain supply to be collected and brought to a depot, or each customer has a certain demand to be fulfilled from a depot. In PDPs, there is a set of transport requests that must be fulfilled. A request consists in the transport of a certain amount of load from a request-specific pickup location to a request-specific delivery location. (There are other types of PDPs. The two-paper survey by Parragh et al. [22], [23] on PDPs lists several problem variants. All of these are essentially special cases of the general PDP as just described.) An important observation is that, when no transshipments are allowed in VRPs and PDPs, a given solution in form of a set of vehicle routes completely determines the path each request takes, be the latter a simple supply or demand request or a pickup-and-delivery request. This is because a request is transported by exactly one vehicle, and only this vehicle visits the corresponding request location(s). When transshipments are possible, this is no longer the case, because the vehicle picking up a request need not necessarily transport it to the depot/delivery location.

PDPs with and without transshipments can be represented as follows within the modelling framework introduced above. For each request r, two customer vertices v_r^+, v_r^- and one virtual vehicle, a dedicated *request vehicle* k_r, are introduced. Request vehicles are non-autonomous and have capacity zero. For each customer vertex i, s_i was defined as the supply of i. s_i may take positive as well as negative values and thus represent a supply as well as a demand. Hence, $s_{v_r^+} > 0$ ($s_{v_r^-} < 0$) specifies the amount of load to be picked up at the pickup location (delivered to the delivery location) of request r, and $s_{v_r^+} = -s_{v_r^-}$.

Each request vehicle k_r moves from s, the virtual start depot vertex, directly to v_r^+, and from v_r^- directly to e, the virtual end depot vertex. This is modelled by adding (s, v_r^+) and (v_r^-, e) to A^{k_r} as the only arcs emanating from s and entering e. Request vehicles cannot use the arc (s, e), since this would imply that the corresponding request is not fulfilled. All compatible real vehicles, that is, vehicles which are technically equipped and allowed to transport r, can reach v_r^+, but are non-autonomous on all arcs leaving v_r^+ and can leave v_r^+ only together with k_r. Similarly, all compatible real vehicles can leave v_r^-, but are non-autonomous on all arcs entering v_r^- and can reach v_r^- only together with k_r.

To model transshipments of pickup-and-delivery requests, for each request r, two vertices v_r^{t-} and v_r^{t+} are introduced for each relevant time window of each desired potential transshipment location. The two vertices are linked by one arc (v_r^{t-}, v_r^{t+}). The first vertex, v_r^{t-}, is only reachable by a composite vehicle of which k_r is part of. At this vertex, k_r is 'decoupled' from the vehicle(s) which has/have accompanied k_r to v_r^{t-}. This is modelled by requiring that k_r can leave v_r^{t-} only via the arc (v_r^{t-}, v_r^{t+}), whereas no other vehicle is allowed to use this arc. Vehicles other than k_r move from v_r^{t-} to any other vertex. At vertex v_r^{t+}, k_r is 'coupled' by a suitable elementary or composite vehicle. This is modelled by requiring that k_r can enter v_r^{t+} only via the arc (v_r^{t-}, v_r^{t+}), whereas all other vehicles may enter v_r^{t+} by any other arc, and that v_r^{t+} can only be left by a composite vehicle of which k_r is part of. To ensure this, the allowable interval for the overall amount of load transferred at both v_r^{t-} and v_r^{t+} is set to $[s_{v_r^+}, s_{v_r^+}]$, and the parameters for the number of different vehicles allowed to transfer load at v_r^{t-} and to receive load at v_r^{t+} as well as those for the overall number of transshipment operations performed at v_r^{t-} and v_r^{t+} are set to one. Now, in order to represent the load transfer, the request vehicle k_r is assumed to have a loading capacity of $s_{v_r^+}$. The intervals for the amount of load received by k_r at all vertices v_r^{t-} and transferred from k_r at all vertices v_r^{t+} are set to $[s_{v_r^+}, s_{v_r^+}]$. The intervals for the amount of load received by or transferred from k_r at all other transshipment vertices are set to $[0, 0]$, and $V_{C,0}^{k_r} = V_C$, that is, k_r is not allowed to collect any load at customer vertices. Thus, the real vehicle carrying a request r to v_r^{t-} transfers it to k_r there, and k_r transfers r to the real vehicle carrying away r from v_r^{t+}. In this way, the request vehicle ensures that the 'right' load, the one picked up at v_r^+, is delivered to v_r^-.

Note that it is possible that a problem instance comprises 'normal' customer vertices as in the VRPTT version described in Section 2 and pickup-and-delivery requests at the same time. Note further that the above representation of pickup-and-delivery requests implies that more than two types of vehicle may join to move in space, as described in the previous section.

9 Conclusion

Vehicle routing problems with multiple synchronization constraints are challenging optimization problems. In contrast to many other types of VRPs and despite their practical relevance, VRPMSs have only recently entered the focus of the scientific community. This may partly be owed to the fact that they are difficult to solve, and that solution approaches for classical VRPs cannot directly be applied to VRPMSs. This author is unaware of any solution procedure for N-echelon vehicle or location-routing problems with temporal synchronization of vehicles of different echelons. Literature on pickup-and-delivery problems with transshipments and simultaneous load transfers, on simultaneous vehicle and crew routing and scheduling problems, and on problems with more than two types of vehicle is still very scarce.

The vehicle routing problem with trailers and transshipments is an archetypal example of a VRPMS. The present paper has demonstrated the usefulness of the VRPTT as a general modelling tool by representing several classes of VRPMSs as VRPTTs. No claim is made that, even if powerful algorithms for the VRPTT were available (which is not yet the case), solving the described applications by transforming them into VRPTTs would necessarily be the method of choice. However, as Pisinger/Ropke [26] have shown, general heuristics capable of solving a broad range of vehicle routing problems can indeed produce high-quality solutions. Moreover, it is to be expected that future algorithmic advances for the VRPTT could yield valuable insights for the solution of other types of VRPMSs. Therefore, the development of exact and heuristic solution procedures for the VRPTT with continuous load variables and volume-dependent load transfer times constitutes a challenging as well as promising area of future research.

Acknowledgement This research was funded by the Deutsche Forschungsgemeinschaft (DFG) under grant no. IR 122/5-1.

References

[1] Baldacci R, Bartolini E, Laporte G (2010):
Some Applications of the Generalized Vehicle Routing Problem
Journal of the Operational Research Society 61: 1072–1077

[2] Brandão J (2004):
A Tabu Search Algorithm for the Open Vehicle Routing Problem
European Journal of Operational Research 157: 552–564

[3] Bredström D, Rönnqvist M (2008):
Combined Vehicle Routing and Scheduling with Temporal Precedence and Synchronization Constraints
European Journal of Operational Research 191: 19–29

[4] Chao I (2002):
A Tabu Search Method for the Truck and Trailer Routing Problem
Computers & Operations Research 29: 33–51

[5] Cordeau J, Gendreau M, Laporte G (1997):
A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems
Networks 30: 105–119

[6] Crainic T, Ricciardi N, Storchi G (2009):
Models for Evaluating and Planning City Logistics Systems
Transportation Science 43: 432–454

[7] Desaulniers G, Desrosiers J, Solomon M (eds) (2005):
Column Generation
Springer, New York

[8] Dohn A, Kolind E, Clausen J (2009):
The Manpower Allocation Problem with Time Windows and Job-Teaming Constraints: A Branch-and-Price Approach
Computers & Operations Research 36: 1145–1157

[9] Drexl M (2007):
On Some Generalized Routing Problems
PhD Thesis, Faculty of Business and Economics, RWTH Aachen University

[10] Drexl M (2011):
Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints
Technical Report 1103, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz

[11] Gendreau M, Potvin JY (eds) (2010):
Handbook of Metaheuristics, Second Edition
Springer, New York

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

[13] Golden B, Wong R (1981):
Capacitated Arc Routing Problems
Networks 11: 305–315

[14] Gonzalez-Feliu J (2009):
The Multi-Echelon Location-Routing Problem: Concepts and Methods for Tactical and Operational Planning
Technical Report, Transport Economics Laboratory, Lyon, France

[15] Gonzalez Feliu J, Perboli G, Tadei R, Vigo D (2008):
The Two-Echelon Capacitated Vehicle Routing Problem
Technical Report, Control and Computer Engineering Department, Politecnico di Torino, Italy

[16] Hollis B, Forbes M, Douglas B (2006):
Vehicle Routing and Crew Scheduling for Metropolitan Mail Distribution at Australia Post
European Journal of Operational Research 173: 133–150

[17] Laporte G (2009):
Fifty Years of Vehicle Routing
Transportation Science 43: 408–416

[18] Li Y, Lim A, Rodrigues B (2005):
Manpower Allocation with Time Windows and Job-Teaming Constraints
Naval Research Logistics 52: 302–311

[19] Longo H, Poggi de Aragão M, Uchoa E (2006):
Solving Capacitated Arc Routing Problems Using a Transformation to the CVRP
Computers & Operations Research 33: 1823–1837

[20] Nagy G, Salhi S (2007):
Location-Routing: Issues, Models and Methods
European Journal of Operational Research 177: 649–672

[21] Noon C, Bean J (1993):
An Efficient Transformation of the Generalized Traveling Salesman Problem
INFOR 31: 39–44

[22] Parragh S, Doerner K, Hartl R (2008):
A Survey on Pickup and Delivery Models Part I: Transportation between Customers and Depot
Journal für Betriebswirtschaft 58: 21–51

[23] Parragh S, Doerner K, Hartl R (2008):
A Survey on Pickup and Delivery Models Part II: Transportation between Pickup and Delivery Locations
Journal für Betriebswirtschaft 58: 81–117

[24] Pearn W, Assad A, Golden B (1987):
Transforming Arc Routing into Node Routing Problems
Computers & Operations Research 14: 285–288

[25] Perboli G, Tadei R, Vigo D (2008):
The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics
Technical Report CIRRELT-2008-55, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

[26] Pisinger D, Ropke S (2007):
A General Heuristic for Vehicle Routing Problems
Computers & Operations Research 34: 2403–2435

[27] Rousseau LM, Gendreau M, Pesant G, Focacci F (2004):
Solving VRPTWs with Constraint Programming Based Column Generation
Annals of Operations Research 131: 159–185

[28] Scheuerer S (2006):
A Tabu Search Heuristic for the Truck and Trailer Routing Problem
Computers & Operations Research 33: 894–909

[29] Semet F, Taillard E (1993):
Solving Real-Life Vehicle Routing Problems Efficiently Using Tabu Search
Annals of Operations Research 41: 469–488

[30] Taillard E, Laporte G, Gendreau M (1996):
Vehicle Routeing with Multiple Use of Vehicles
Journal of the Operational Research Society 47: 1065–1070

[31] Toth P, Vigo D (eds) (2002):
The Vehicle Routing Problem
SIAM Monographs on Discrete Mathematics and Applications, Philadelphia

[32] Zäpfel G, Bögl M (2008):
Multi-Period Vehicle Routing and Crew Scheduling with Outsourcing Options
International Journal of Production Economics 113: 980–996