A variable neighborhood search heuristic for the vehicle routing problem with trailers and transshipments

Kees ter Brugge

April 30, 2017

Abstract

The vehicle routing problem with trailers and transshipments (VRPTT) generalizes the truck and trailer routing problem (TTRP) such that a single trailer can be shared by multiple lorries. A simpler representation of load transfers has been formulated for the VRPTT. A variable neighborhood heuristic (VNS) has been developed that finds results close to the upper bounds of all instances with up to eight customers within minutes. The results of the VNS for the VRPTT are the first that include the time related costs of a solution, which allows for a comparison with results for the TTRP on these instances. The comparison suggest that these instances are not a good choice for showing the benefits of trailers sharing, since trailer sharing was not used in any of the best solutions found by the VNS. An extension was made to the model which allows lorries to take multiple trips during one planning horizon. This extension resulted in significant cost reductions on many instances.

Contents

1 Introduction

The profitability of a transport company, or any company for that matter, is closely linked to the efficiency with which it uses its resources. In a transport company this roughly translates to the use of its trucks and man hours, whilst fulfilling all service requirements. A more efficient use of resources by transport companies will reduce the environmental impact of that industry, which for the U.S. has been estimated as 27% of the total greenhouse gas emissions in 2013 [1]. The research that has been done on the vehicle routing problem (VRP) can help manage these challenges.

Laporte defines the VRP concisely as 'the problem of designing least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints' [2]. It was first introduced by Dantzig and Ramser in 1959 [3] and has driven the development of exact algorithms and heuristics ever since. In practically no time after its introduction hundreds of models and optimization methods were considered for various versions of the VRP. Today, many software packages exist on the market, all attempting to solve some of the real-world versions of the problem [4].

The economic incentives behind being able to solve real-world VRPs have produced many versions of the problem over the years, all catering to different operational realities. These extensions of the VRP are often called rich vehicle routing problems (RVRPs). There is a need for a unifying solution framework for these RVRPs, since it will increase our understanding of the impact specific problem attributes have on our ability to find solutions. Furthermore, it will reduce the development time of solution methods for problems with new types problem attributes [5].

Progress has been made in creating heuristics that perform well on a broad class of problems [5, 6]. Still, in Drexl's comparison of commercial software and scientific research for modeling and solving VRPs he concludes: 'models and algorithms for integrated and synchronized vehicle routing are still scarce: in almost all vehicle routing models and algorithms, the routes of the different vehicles are assumed to be independent of one another, so that modifying one route does not have any effects on other routes'. In [7] the case is made that in a high number of cases this assumption does not hold.

Problems with this interdependence attribute belong to the class of problems called vehicle routing problems with multiple synchronization constraints (VRPMS). These problems are of great practical relevance and quite hard to solve. In [8] Drexl proposes and demonstrates the value of the vehicle routing problem with trailers and transshipment (VRPTT) as a general modeling tool for several classes of VRPMSs. Drexl solved small instances of the VRPTT using a branch-and-cut method [9], but concludes that 'a heuristic procedure for the VRPTT is needed, but still missing'.

VRP heuristics have gotten much attention. Exact algorithms have trouble solving even moderately sized VRPs and have a slow convergence rate [10]. Heuristics can be used to find high-quality solutions quickly for realisticallysized problems. They do not guarantee optimality though. The adaptive large neighborhood search (ALNS) heuristic is a promising optimization method [6]. Good results have been achieved by applying it to a broad class of VRPs. Masson et al. have had good results applying it to the dial-a-ride problem with transfers in [11], which is a problem that belongs to the class of VRPMSs .

An outline of a heuristic approach to solving VRPMSs is given in [12]. Drexl suggests a two-stage approach that is also used in [13] for simultaneous vehicle and crew routing and scheduling.

1.1 About TransMission

The inducement to start working on the VRPTT has been provided by a company called TransMission. TransMission is the largest cooperation of independent transport and distribution companies in the Benelux. The transport and distribution companies within the TransMission group work under one name. These organizations are also independent operations, and many of them are family companies. TransMission has 18 depots whereof thirteen are located in The Netherlands, four in Belgium and one in Luxembourg. Each depot services a region, such that they cover the Benelux together. In Figure 1 the separate regions can be seen on the map.

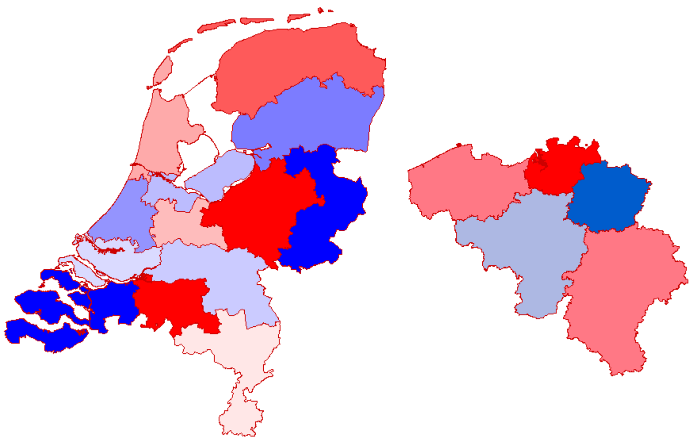

Figure 1: The regions serviced by the transport and distribution companies within the TransMission group . The depot located in Luxembourg services the South-East of Belgium and the whole of Luxembourg.

TransMission uses different types of trucks and trailers and many of their depots can be used as cross-docks. Line hauls are used to move cargo between depots during nights. Depots pick up and deliver the cargo in their service regions during days. There is a long-term centrally planned set of line hauls. There is also a set of line hauls that is planned daily by the depots themselves. TransMission would like to to produce its transport plans every day using all information available to the depots. To achieve this, producing the transport plan needs to become automated.

1.2 Scope

TransMission's problem is an extremely rich one. The first step to solve a problem is to model it properly. It is a pick up and delivery problem without fixed assignments of trailers to lorries where goods can be transfered between vehicles before they reach their destination.

In [14] results are reported on the pickup and delivery problem with transshipments. In this paper the available fleet is homogeneous and does not contain trailers. The VRPTT considered in [15] only deals with collecting one homogeneous good, instead of picking up and delivering various goods to various locations. Both works do not cover all features of TransMission's problem. Of these two models, the VRPTT has been estimated to be the best basis for a model for TransMission's problem. Its potential to model a wide class of problems has already been demonstrated in [8].

Powerful algorithms for solving the VRPTT do not exist yet [9]. Stochastic optimization methods and/or heuristics may offer solutions in cases where solving problems exactly is too slow. Developing such an algorithm is therefore the most logical first step towards solving TransMission's problem. This task is the scope of this thesis. The optimization method will be a simplified version of the ALNS, namely a variable neighborhood search (VNS). The ALNS has shown promising results on large instances in [14].

How the results of this thesis will be used to automate the production of transport plans for TransMission is outside the scope of this thesis.

1.3 Methodology

A week was spent within TransMission to work in different departments in order to gain domain knowledge. The problem was defined and the scope was determined in collaboration with TransMission.

The python programming language [16] was used to make the algorithm since it allows for rapid prototyping and easy visualizations. The Numpy package [17] was used for fast array computations. The Numba library [18] was used to speed up performance-critical parts of the algorithm. Instances provided in [15, 19] were used to analyze the performance of the algorithm.

1.4 Outline

This section provides the problem under consideration with context. Section 2 explains the problem and formulates the model. In Section 3 all parts of the optimization method are explained. The test methods and the results that were produced by the optimization method are described and interpreted in Section 4. In Section 5 conclusions are drawn and Section 6 outlines promising directions of this area of research.

2 The Problem

2.1 Problem Description

The VRPTT can be described as follows: There is a set of customers with a known, deterministic supply of a single good. There is a set of transshipment locations that may be used for parking and load transfers. All locations have opening and closing times. To collect the customers' supply, a fleet of heterogeneous, limited-capacity vehicles stationed at the vehicle depot is available. Some customers can not be visited with trailers because of accessibility constraints. While goods are collected, costs should be minimized. A lorry ends its path if it unloads at the depot or after it decouples a trailer at the depot.

The fleet of vehicles consists of lorries and trailers. Trailers can only move if pulled by a lorry. Not all trailers and lorries are compatible with each other. All vehicles may have different capacities, fixed costs and variable costs. Lorries may differ in the amount of time they need to collect a certain amount of customer supply. There is no fixed assignment between trailers and lorries. Trailers can be shared by multiple lorries. Any trailer can be pulled by any compatible lorry.

Transshipment locations can be used by lorries to transfer load to a trailer. There are no costs associated with coupling and uncoupling a trailer except for the cost incurred by the time it takes to finish the operation. A vehicle will start and end its route at the depot location.
Customers that can not be visited with trailers are called lorry customers. Customers that can be visited with trailers are called trailer customers. Split collections are not allowed, hence each customer's supply is collected by one lorry. There is no customer with a supply greater than the capacity of the lorry with the least capacity. Trailer customer locations can also be used as transshipment locations.

The VRP is a special case of the VRPTT, namely an instance without transshipment locations and without trailers. Since the VRP is NP-hard [2], the VRPTT is also NP-hard.

2.2 Problem Model

A solution to an instance of the VRPTT should answer the following questions:

- How should the customers be divided over the lorries and in which order should the lorries visit them to collect the customers' supply?
- How should the lorries use the trailers?
- How should the load be divided between the vehicles?
- When do all the events in the plan take place?

2.2.1 Simple Example

Let us explore a simple example of the VRPTT. In Figure 2 we can see an illustration of a VRPTT. For the moment we ignore some aspects of the VRPTT for the sake of clarity. The VRPTT does not allow a vehicle to leave the depot after it visits the depot.

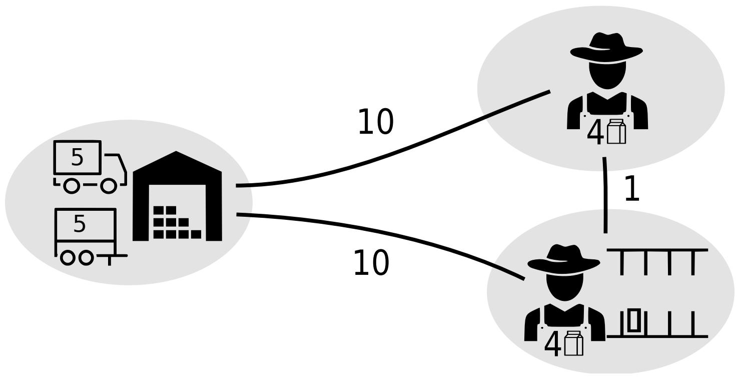

Figure 2: Illustration of an VRPTT with a trailer customer, a lorry customer and a fleet consisting of a trailer and a lorry.

The example problem contains three locations. One of the locations is the depot. At the depot there is a fleet consisting of a lorry and a trailer. The other two locations are customers. Both customers have a supply of four that needs to be collected by the vehicles and needs to be unloaded at the depot. Both vehicles in the fleet have a capacity of five. The customers are at a distance of one of each other and at a distance of ten to the depot of the transporter.

The customer on the bottom has a parking lot, hence it has enough space to be visited with a trailer. This customer is therefore called a trailer customer. A location that can be visited with a trailer can be used to park a trailer and/or to transfer load from a lorry to a trailer. Such a location is called a transshipment location. The other customer does not have a parking lot and can therefore not be visited with a trailer, which makes it a lorry customer.

2.2.2 Events

Let us attempt to construct a plan that answers at least the first two questions that a solution to the VRPTT needs to answer. Since vehicles are not allowed to leave the depot after visiting it, the lorry can not unload at the depot between visiting the customers. The lorry and the trailer can not separately take care of a customer, since the trailer can not move on its own. Since the lorry alone does not have enough capacity to collect the supply of both customers, it needs to use the capacity of the trailer.

The lorry can not visit the lorry customer whilst coupled to the trailer. Let us say that they go to the trailer customer first. So far the plan is as follows:

1. The lorry starts it path at the depot.

2. The lorry couples the trailer.

3. The lorry and trailer drive to the trailer customer.

Now there are two good courses of action. Either the lorry collects the supply of the trailer customer first or it decouples and parks the trailer first such that it can collect the supply of the lorry customer. Let us say that it collects the supply of the trailer customers first, then the plan continues as follows:

4. The lorry collects the supply of the trailer customer.
5. The lorry transfers at least three units of the supply to the trailer such that it has at least four capacity available for the supply of the lorry customer.
6. The lorry parks and decouples the trailer.
7. The lorry drives to the lorry customer.
8. The lorry collects the supply of the lorry customer.
9. The lorry drives back to the trailer customer.
10. The lorry couples the trailer.
11. The lorry drives back to the depot.
12. The lorry decouples the trailer and unloads.
13. The lorry ends its path and unloads.

This is quite a verbose description of a transport plan. Notice that every step is either about the lorry travelling between locations or about the lorry performing an activity at a location. This list of steps can be compressed into a list where every step consist of an activity and a location, see Table 1. From now on this combination is called an event. The sequence of events that a lorry is involved in is its path.

Event #	Location	Activity	Event ID
1	depot	start path	A
2	depot	couple the trailer	C
3	trailer customer	collect supply	F
4	trailer customer	decouple the trailer	G
5	lorry customer	collect supply	E
6	trailer customer	couple the trailer	H
7	depot	decouple trailer	D
8	depot	end path	B

Table 1: The events in the plan.

Notice that the load transfer from the lorry to the vehicle is not treated as a separate event. This is because we allow the load to be transfered from the lorry to the trailer at any event as long as they are coupled. In this case the transfer happens in either event three or four and the amount that is transfered lies between three and four such that the lorry has enough capacity to collect the supply of the lorry customer.

Notice also that by choosing to park a trailer, two events are created. A trailer needs to start and end its path at the depot, hence if it is decoupled at some transshipment location it needs to be coupled again there as well.

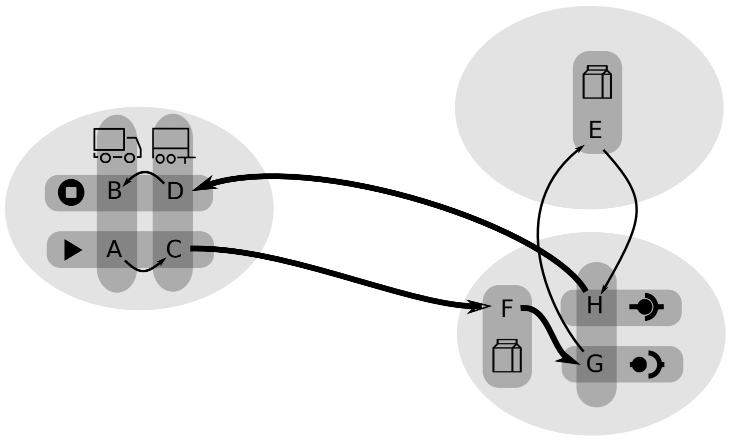

Figure 3: An illustration of the lorry's path. Every event is indicated with the event ID supplied in Table 1. The activity of the event is illustrated with a symbol. The position indicates the location of the event. The bold lines indicate where the lorry is coupled to the trailer. The thin lines indicate where the lorry is not coupled to the trailer.

If we look at Figure 3 we can see all the events in our plan and the path of the lorry. From the path of the lorry we can infer the path of the trailer, namely it starts at C, then we follow the bold line to F and G. Transshipment events come in pairs, a couple event and a decouple event. A trailer decoupled at G can be coupled again at H. From H we follow the bold line back to D.

Notice that each event in the plan is unique, in that it occurs only once in the path of the lorry. Furthermore, notice how we can answer the first two questions that a solution to the VRPTT needs to answer by looking at the path of the lorry.

2.2.3 Load Division

The third question that needs to be answered is about how, given the events of a plan, the customer supply ends up at the depot. The vehicles have a limited capacity, so they need to coordinate how load is divided between them. This is done by determining at which event of the plan a load transfer from a lorry to a trailer needs to take place and the amount that needs to be transfered.

The customers' supply flow from customers to the depot as presented in Figure 4. The only sources of supply are the customers, both with a supply of four. The only sink for the load is the depot. The supply can reach the depot by flowing from a customer to the lorry and then to the depot. Alternatively, supply can flow from a customer to the lorry, then to the trailer, then to the

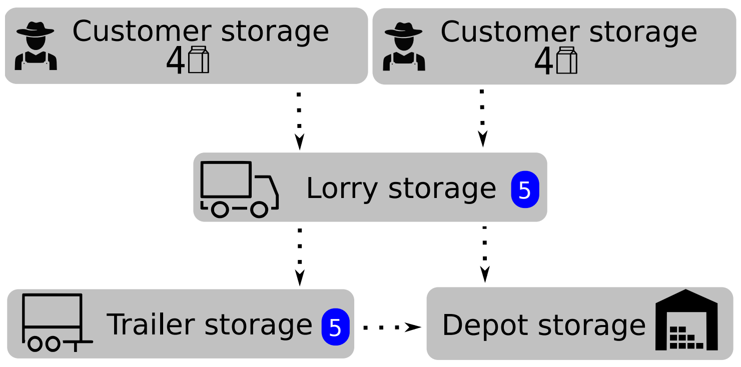

Figure 4: An illustration representing how load can flow from the customers to the depot. The amount of supply of the customers is indicated with the milk icon and the capacity of the vehicles is indicated in blue.

depot. Before the vehicles can be finished with collecting all eight units of customer supply, at least three units of supply must have flown to the trailer, since the lorry can only hold five units of supply at any one time. No more than five units of supply can have flown to the trailer in total, since it can only hold five units. Notice how the lorry can decrease its amount of load before unloading at the depot, but that the trailer can not.

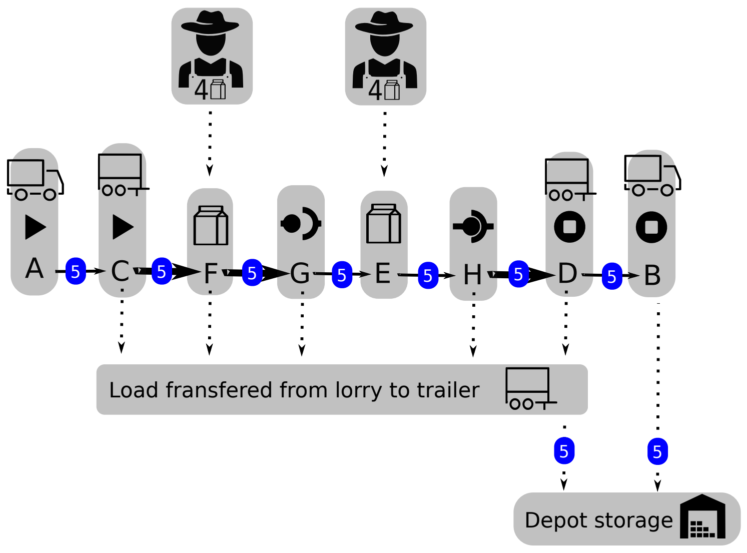

Figure 5: The load division problem. Dashed arrows represent the ways that load can flow. If flow restrictions exist, they are indicated in blue. Arrows indicate how load can flow from event to event as cargo inside the lorry. Bold arrows indicate when the lorry is coupled to the trailer.

To not only determine the amount of supply that needs to be transfered, but also the event at which the transfer needs to take place, we need to consider the flow of supply per event. The ways supply can flow at each event is illustrated in Figure 5. Every time the lorry is at an event, it can choose to transfer load to the trailer, if they are coupled. At the supply collection events, F and E, load can flow from a customer to the lorry. When the lorry ends its path at the depot, event B, load can flow from the lorry to the depot. The only time load can flow out of the trailer is when it decouples at the depot, event D. The maximum amount of load that can flow from a vehicle to the depot is equal to its capacity, which is five for both vehicles. We can ensure that the trailer is never overloaded during its path if we constrain the amount of flow from the trailer to the depot to the trailer's capacity, since that is the only time that load can flow out of the trailer. Load can flow into and out of the lorry at multiple events, therefore we have to restrict the flow between events to the lorry's capacity such that during its path it never carries more load than its capacity.

This problem of dividing load between vehicles can be abstracted to a maximum flow problem, see Figure 6. The maximum flow problem asks how much load can transfer from a source to a sink. The vertices of the maximum flow problem consist of the events of the plan, a special vertex P for the trailer to show the flow of load from the lorry to the trailer, one sink *sink* and one source *source*. We already have one sink, which is the depot. However, we have two sources of supply, which we replace with one source. The source has an edge to a customer's supply collection event with a capacity equal to the customers amount of supply, such that the maximum amount of supply that can be collected at the customer's is maintained. We have already determined where and how much load can flow, which is reflected in the rest of edges and the edge capacities.

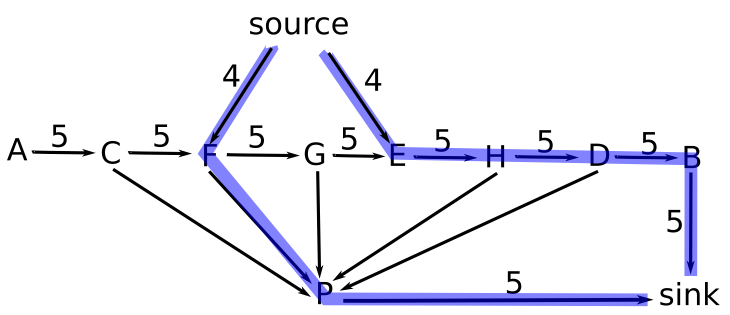

Figure 6: A directed graph representing the load division problem. If an edge has a limited capacity, the capacity is shown.

A possible solution to this max flow problem is a flow of size four on path $\{S, F, P, T\}$ and a flow of size four on path $\{S, E, H, D, B, T\}$. This is the solution illustrated in Figure 6. The third question a solution to the VRPTT needs to answer can thus be solved by solving a maximum flow problem.

If the maximum flow is smaller than the sum of the customers' supply, we know that given the current event plan no way of dividing the load between the vehicles exists, that would allow us to collect all the customers' supply. Not creating a separate event for load transfers has enabled us to infer this.

2.2.4 Timing

The fourth question that needs to be answered is about at what time the events of a plan take place. It would be preferable if the lorry would take the least amount of time possible to complete its path and start performing each activity as early as possible. However, there should be enough time for the lorry to travel between locations, to perform the activities of the events and not violate the time windows of the locations.

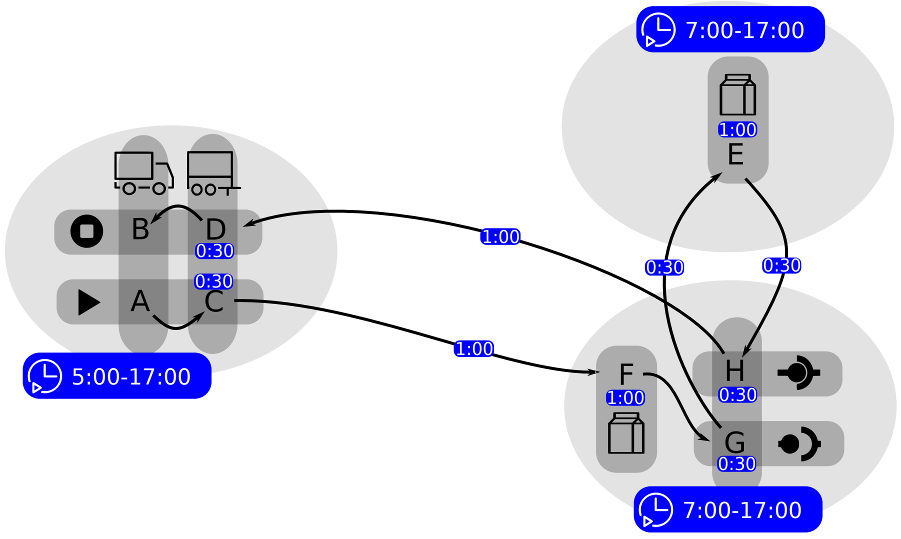

Figure 7: An illustration of the lorry's path with time information illustrated in blue in hours. Each location has a time window that holds for all events at that location. If no time information is provided the travel duration or event duration is zero. As Figure 7 illustrates, some events don't take up any time and going from event to event on the same location does not take up any time.

To illustrate the problem we extend the example of Section 2.2.1 with time windows, travel times, event durations and a planning horizon of 0:00 till 22:00 hours. The rest of the time related information is provided in Figure 7.

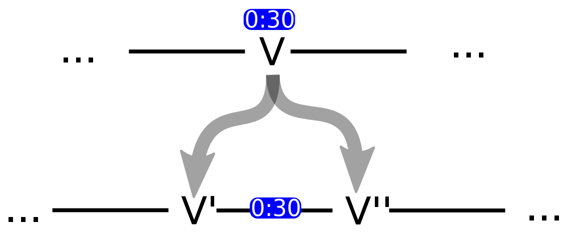

Figure 8: An illustration of how the duration of an event V is modeled by replacing the event with two other events V' and V'' with an edge weight.

Similar as the division of load, the problem of timing the events can also be modeled as a graph problem. This is done by creating a precedence graph, where edge weights represent the time between events. A source and a sink are added to model the beginning and the end of the planning horizon. Since some events have a duration themselves each event is first divided into two separate events such that the duration of an event can be represented as the edge weight between the two new events as shown in Figure 8. Furthermore, a directed edge is added between the decouple and the couple event at the transshipment location to model that a trailer must be decoupled before it can be coupled again.

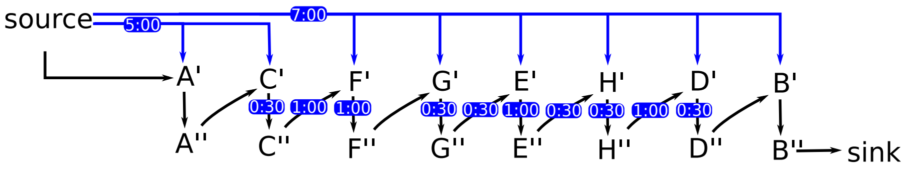

Figure 9: An illustration of the precedence graph. The edges that model the opening times of the locations at which the events take place are indicated with the blue edges. The black edges indicate the event durations and the travel times between the events. If an edge's weight is not shown it has the value zero.

Let us first try to answer at what time the lorry can finish its path. We answer this by performing a longest path search on the precedence graph shown in Figure 9, which also has edges that model the opening times of the locations at which the events take place. We see that the longest path $\{source, F', F'', G', G'', E', E'', H', H'', D', D'', B', B''\}$ from the source to the end of the event that stops the lorry's path, B'', is 12:30 hours. Therefore, the earliest that the lorry can end its path is at 12:30 o'clock. The path has taken the lorry 7:30 hours.

If the lorry start its path as early as possible, 5:00 o'clock, the lorry would arrive at the trailer customer at 6:30 and would have to wait have an hour before it could start collecting the supply. We can perform a similar longest path search backwards to see at what time the lorry can start its path whilst finishing its path at 12:30 o'clock. The longest path search is performed on the antecedence illustrated in Figure 10. The time between the time at which the lorry can end its path at the earliest and the end of the planning horizon is 22:00 - 12:30 = 9:30 hours.

The longest path in the antecedence graph from the sink to the start of the lorry's path A' is 16:30 hours, hence the lorry can start its path at 5:30 and still end its path 12:30 o'clock.
We now perform the longest path search on the precedence graph illustrate in Figure 9 again, only this time the edge weight of the edge between the source and A' is 5:30. This longest search provides the timing of the events that minimizes the lorry's path duration, schedules activities as early as possible whilst taking into account the travel times, the activity durations and the location's time windows. The resulting time table is given in Table 2

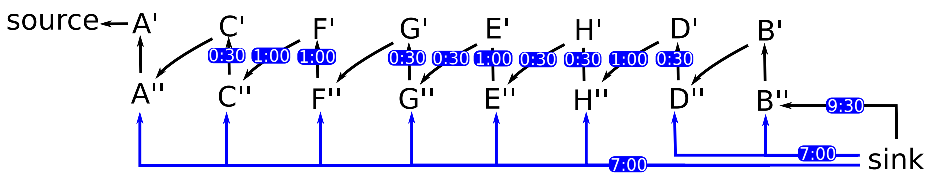

Figure 10: An illustration of the precedence graph. The edges that model the opening times of the locations at which the events take place are indicated with the blue edges. The black edges indicate the event durations and the travel times between the events. If an edge's weight is not shown it has the value zero.

Event ID	Start time	End time
A	5:30	5:30
C	5:30	6:00
F	7:00	8:00
G	8:00	8:30
E	9:00	10:00
H	10:30	11:00
D	12:00	12:30

Table 2: The timing of the events in the lorry's path.

2.2.5 Multiple Trips

In the VRPTT a trailer can only be coupled by a lorry at the depot if this is the first thing a lorry does after starting its path. Furthermore, after the lorry decouples a trailer at the depot is has to end its path. The VRPTT can be extended in a useful way by removing these limitations.

If lorries are not limited anymore to only coupling (decoupling) trailers at the depot at the beginning (ending) of their path, the following situation may occur: a lorry is full after collecting customer supply; the lorry couples a trailer at the depot and transfers load to it; the lorry decouples the trailer again, without the trailer even having left the depot; the lorry goes to the next customer to collect its supply. The trailer functions in such a case as an unloading dock. This de facto removes the restriction that a lorry can only unload once at the depot. Therefore this model will be named the vehicle routing problem with trailers, transshipments and multiple unloads (VRPTTMU).

2.3 Problem Formulation

2.3.1 Instance Sets and Parameters

In this section we define the sets and parameters provided with the instances used in this thesis.

Let $\mathcal{L}$ of be a set of locations and let $\mathcal{D} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}^0_+$ be a mapping from a pair of locations to the distance between them. Let $\mathcal{L}^{\text{types}} = \{\text{depot}, \text{lorry customer}, \text{trailer customer}, \text{pure transshipment location}\}$ be a set of location types. Each location $i \in \mathcal{L}$ has a location type $\mathit{locationType}_i \in \mathcal{L}^{\text{types}}$, a start of its time window $\alpha_i \in \mathbb{R}^0_+$ and an end of its time window $\beta_i \in \mathbb{R}_+$.

Each customer location $i \in \mathcal{L}^{\text{customer}} := \{j \in \mathcal{L} : \mathit{locationType}_j \in \{\text{lorry customer}, \text{trailer customer}\}\}$ has a supply $\mathit{locationSupply}_i \in \mathbb{R}_+$ that needs to be collected.

Let $\mathcal{F}$ be a set of vehicles. Let $\mathcal{F}^{\text{types}} = \{\text{lorry}, \text{trailer}\}$ be a set of vehicle types. Let $\mathcal{F}^{\text{axle}-\text{types}} = \{2 - \text{axles}, 3 - \text{axles}\}$ be a set of axles types of vehicles. Each vehicle $i \in \mathcal{F}$ has a vehicle type $f_i^{\text{type}} \in \mathcal{F}^{\text{types}}$, a capacity $f_i^{\text{capacity}} \in \mathbb{R}_+$, a fixed cost $f_i^{\text{fixed}} \in \mathbb{R}^0_+$, a variable distance cost $f_i^{\text{distance}} \in \mathbb{R}^0_+$, a variable time cost $f_i^{\text{time}} \in \mathbb{R}^0_+$, an amount of axles $f_i^{\text{axle}-\text{type}} \in \mathcal{F}^{\text{axle}-\text{types}}$, a maximum speed $f_i^{\text{speed}} \in \mathbb{R}_+$, that is used on distances longer than ϕ distance units, a slower speed $f_i^{\text{shortspeed}} \in \mathbb{R}_+$ that is used on distances shorter than or equal to ϕ distance units.

Each lorry $i \in \mathcal{F}^{\text{lorry}} := \{j \in \mathcal{F} : f_j^{\text{type}} = \text{lorry}\}$ has a load transfer speed $f_i^{\text{loadspeed}} \in \mathbb{R}_+$ which is expressed in terms of time per unit.

The time that it takes to decouple or couple a trailer at the depot is τ^D time units. The time that it takes to decouple or couple a trailer at a transshipment location is τ^R time units. Let $\tau^{\min}$ and $\tau^{\max}$ be the start and end respectively of the instance's time horizon.

2.3.2 Model Sets and Parameters

In this section we define extra sets and parameters to be able to construct the model.

Transshipments consist of two activities, namely uncoupling and coupling trailers. Let $\mathcal{R}^-$ and $\mathcal{R}^+$ be the sets of transshipment decouple and couple vertices respectively. Each transshipment location $j \in \mathcal{L}^{\text{transshipment}} := \{i \in \mathcal{L} : \mathit{locationType}_i \in \{\text{pure transshipment location, trailer customer}\}\}$ has decouple vertices $r_m^{j,-} \in \mathcal{R}^-, m \in \{1, \ldots, \eta\}$ and couple vertices $r_m^{j,+} \in \mathcal{R}^+, m \in \{1, \ldots, \eta\}$ where η is the maximum number of transshipments that can take place at a transshipment location. Let $\mathcal{E}^{\text{pairs}} = \{(r_m^{j,-}, r_m^{j,+}) : r_m^{j,-} \in \mathcal{R}^-, r_m^{j,+} \in \mathcal{R}^+, m \in \{1, \ldots, \eta\}, j \in \mathcal{L}^{\text{transshipment}}\}$ be the set of transshipment vertex pairs, alternatively interpreted as the set of edges from each transshipment decouple vertex to its couple vertex.

Let $\mathcal{S}$ be the set of supply collection vertices. Each customer location $i \in \mathcal{L}^{\text{customer}}$ has a supply collection vertex $s_i \in \mathcal{S}$ that is used to collect the supply of the customer. let $\mathcal{S}^{\text{trailer}} := \{s_i \in \mathcal{S} : \mathit{locationType}_i = \text{trailer customer}\}$ be the set of all supply collection vertices located at trailer customers and $\mathcal{S}^{\text{lorry}} := \{s_i \in \mathcal{S} : \mathit{locationType}_i = \text{lorry customer}\}$ be the set of all supply collection vertices located at lorry customers.

Let $\mathcal{M}^+$ and $\mathcal{M}^-$ be the set of all lorry start and end vertices respectively. At the depot each lorry $k \in \mathcal{F}^{\text{lorry}}$ has a start vertex $m_k^+ \in \mathcal{M}^+$ that starts its path and an end vertex $m_k^- \in \mathcal{M}^-$ that ends its path and unloads the lorry.

Let $\mathcal{N}^+$ and $\mathcal{N}^-$ be the set of all trailer start and end vertices respectively. At the depot each trailer $l \in \mathcal{F}^{\text{trailer}} := \{k \in \mathcal{F} : f_k^{\text{type}} = \text{trailer}\}$ has a start vertex $n_l^+ \in \mathcal{N}^+$ that is used to couple the trailer to a lorry, and an end vertex $n_l^- \in \mathcal{N}^-$ that decouples the trailer from the lorry. When the trailer is decoupled it is unloaded.

For each vertex $v \in \mathcal{V} := \mathcal{M}^+ \cup \mathcal{M}^- \cup \mathcal{N}^+ \cup \mathcal{N}^- \cup \mathcal{R}^- \cup \mathcal{R}^+ \cup \mathcal{S}$, let v^{loc} be the location that the vertex belongs to, $\tau_v^{\text{start}} = \alpha_{v^{\text{loc}}}$ be the start time of the vertex's time window, $\tau_v^{\text{end}} = \beta_{v^{\text{loc}}}$ be the closing time of the vertex's time window.

Let the duration of a vertex visit be denoted as

$$\tau_{v,k}^{\text{visit}} = \begin{cases} 0, & \text{if } v \in \mathcal{M}^- \cup \mathcal{M}^+, \\ \tau^D, & \text{if } v \in \mathcal{N}^- \cup \mathcal{N}^+, \\ \tau^R, & \text{if } v \in \mathcal{R}^- \cup \mathcal{R}^+, \\ f_k^{\text{loadspeed}} \cdot \mathit{locationSupply}_{v^{\text{loc}}}, & \text{if } v \in \mathcal{S}, \end{cases} \quad v \in \mathcal{V},\ k \in \mathcal{F}^{\text{lorry}}. \tag{1}$$

Let $\delta_{u,v} = \mathcal{D}(u^{\text{loc}}, v^{\text{loc}}),\ u, v \in \mathcal{V}$ be the distance between two vertices u and v.

Let the travel time between vertices $u, v \in \mathcal{V}$ by vehicle $k \in \mathcal{F}$ be denoted as

$$\tau_{u,v,k}^{\text{travel}} = \begin{cases} \delta_{u,v}/f_l^{\text{shortspeed}}, & \text{if } \delta_{u,v} \le \phi, \\ \delta_{u,v}/f_l^{\text{speed}}, & \text{otherwise}, \end{cases} \quad u, v \in \mathcal{V},\ k \in \mathcal{F}. \tag{2}$$

Let $\tau_{u,v,k,l}^{\text{extra}} = \max(0, \tau_{u,v,l}^{\text{travel}} - \tau_{u,v,k}^{\text{travel}}),\ k \in \mathcal{F}^{\text{lorry}},\ (u,v) \in \mathcal{E},\ l \in \mathcal{F}^{\text{trailer}}$ be the amount of extra time it takes for lorry k to traverse edge (u, v) if the lorry is coupled to trailer l.

2.3.3 Decision Variables

Events are the terms in which the transport plan is formulated. All possible events can be represented using the just defined vertices.

A transport plan consist of paths, the assignment of vehicles to those paths, the time table and the load table of those paths, all of which will be defined. Solving the VRPTT yields such a transport plan.

The VRPTT is described with a mixed-integer linear problem over the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of all edges between the vertices $\mathcal{E} = \{(u, v) : u, v \in \mathcal{V}, u \neq v\}$.

From now on the path of lorry $k \in \mathcal{F}^{\text{lorry}}$ means a simple path in $\mathcal{G}$ from the lorry's start vertex m_k^+ to its end m_k^-.

We model a lorry's path using the binary variable

$$x_{u,v}^k = \begin{cases} 1, & \text{if lorry } k \in \mathcal{F}^{\text{lorry}} \text{ traverses edge } (u,v) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

We model the movements of trailers by using the binary variable $y_{u,v}^{k,l}$, where

$$y_{u,v}^{k,l} = \begin{cases} 1, & \text{if lorry } k \in \mathcal{F}^{\text{lorry}} \text{ is coupled to trailer } l \in \mathcal{F}^{\text{trailer}} \text{ whilst traversing edge } (u,v) \in \mathcal{E} \text{ ,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The time table contains information about the point in time at which a vertex is reached by a lorry. We model the time table with the variable $t_{u,v}^k$, where

$$t_{u,v}^k = \begin{cases} a, & \text{if a lorry } k \in \mathcal{F}^{\text{lorry}} \text{ traverses edge } (u,v) \in \mathcal{E} \text{ and arrives at } j \text{ at time } a \in \mathbb{R}_+^0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The load table contains information about the load with which a vehicle traverses an edge. We model the load table with the variable $z_{u,v}^k$, where

$$z_{u,v}^k = \begin{cases} a, & \text{if a vehicle } k \in \mathcal{F} \text{ traverses edge } (u,v) \in \mathcal{E} \text{ with load } a \in \mathbb{R}_+^0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Let $X = (x_{u,v}^k)$ for $k \in \mathcal{F}^{\text{lorry}}, u, v \in \mathcal{V}$ be the lorries' paths , let $Y = (y_{u,v}^{k,l})$ for $k \in \mathcal{F}^{\text{lorry}}, l \in \mathcal{F}^{\text{trailer}}, u, v \in \mathcal{V}$ be the trailers' paths, let $T = (t_{u,v}^k)$ for $k \in \mathcal{F}^{\text{lorry}}, u, v \in \mathcal{V}$ be the time table and let $L = (z_{u,v}^k)$ for $k \in \mathcal{F}, u, v \in \mathcal{V}$ be the load table. A solution to the VRPTT can then be expressed with the tuple (X, Y, L, T). Let from now on a solution's path assignment refer to (X, Y).

2.3.4 Lorry Constraints

A vertex can be on one path at the most,

$$\sum_{v \in \mathcal{V}} \sum_{k \in \mathcal{F}^{\text{lorry}}} x_{u,v}^k \leq 1, \quad u \in \mathcal{V}. \tag{7}$$

A vertex can either be the start of a lorry's path and have no predecessor, be the end of a lorry's path and have no successor, have both a predecessor and a successor or have neither. Hence,

$$\sum_{u \in V} \left(x^k_{u,v} - x^k_{v,u} \right) = \begin{cases} -1, & \text{if } v = m^+_k, \\ 1, & \text{if } v = m^-_k, \\ 0, & \text{otherwise,} \end{cases} \qquad v \in \mathcal{V},\ k \in \mathcal{F}^{\text{lorry}}. \tag{8}$$

Every supply collection vertex is visited by a lorry,

$$\sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{S}} \sum_{k \in \mathcal{F}^{\text{lorry}}} x^k_{u,v} \geq |\mathcal{S}|. \tag{9}$$

2.3.5 Trailer Constraints

A trailer can only traverse an edge that is traversed by a lorry. A lorry can pull at most one trailer at a time,

$$\sum_{l \in \mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} - x^k_{u,v} \leq 0, \quad u, v \in \mathcal{V}, k \in \mathcal{F}^{\text{lorry}}. \tag{10}$$

No lorry customers can be visited with a trailer,

$$\sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{S}^{\text{lorry}}} \sum_{k \in \mathcal{F}^{\text{lorry}}} \sum_{l \in \mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} \leq 0. \tag{11}$$

A lorry can not arrive at a couple vertex whilst coupled to a trailer,

$$\sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{N}^+ \cup \mathcal{R}^+} \sum_{k \in \mathcal{F}^{\text{lorry}}} \sum_{l \in \mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} \leq 0. \tag{12}$$

A lorry can not leave a decouple vertex without being coupled to a trailer,

$$\sum_{u \in \mathcal{N}^- \cup \mathcal{R}^-} \sum_{v \in \mathcal{V}} \sum_{k \in \mathcal{F}^{\text{lorry}}} \sum_{l \in \mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} \leq 0. \tag{13}$$

The vertices a lorry can only depart from whilst coupled to a trailer are the trailer coupling vertices. The vertices a lorry can only arrive at whilst coupled to a trailer are the trailer decoupling vertices,

$$\begin{aligned} &\sum_{l \in \mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} - x^k_{u,v} = 0, \\ &\qquad (u,v) \in \{(u',v') : u' \in \{\mathcal{N}^+ \cup \mathcal{R}^+\}, v' \in \mathcal{V}\} \cup \\ &\qquad\qquad \{(u',v') : u' \in \mathcal{V}, v' \in \{\mathcal{N}^- \cup \mathcal{R}^-\}\}, \\ &\qquad k \in \mathcal{F}^{\text{lorry}}. \end{aligned} \tag{14}$$

A lorry departs from a trailer customer's supply collection vertex with a trailer if and only if it also arrived with that trailer,

$$\sum_{u \in \mathcal{V}} \left(y^{k,l}_{u,v} - y^{k,l}_{v,u} \right) = 0, \quad v \in \mathcal{S}^{\text{trailer}},\ k \in \mathcal{F}^{\text{lorry}},\ l \in \mathcal{F}^{\text{trailer}}. \tag{15}$$

Each trailer can only be coupled and decoupled at the depot at the start and end vertex designated to the trailer,

$$y^{k,l}_{u,w} = y^{k,l}_{w,v} = 0, \quad u \in \mathcal{N}^+ \setminus \{n^+_l\},\ v \in \mathcal{N}^- \setminus \{n^-_l\}, \tag{16}$$
$$w \in \mathcal{V}, k \in \mathcal{F}^{\text{lorry}}, l \in \mathcal{F}^{\text{trailer}}.$$

Each transshipment couple vertex is paired with a decouple vertex. A trailer departs from a transshipment location's couple vertex if and only if it also arrives at the decouple vertex that is paired with the couple vertex,

$$\sum_{w\in\mathcal{V}} \sum_{k\in\mathcal{F}^{\text{lorry}}} \left(y^{k,l}_{w,u} - y^{k,l}_{v,w}\right) = 0, \quad l \in\ \mathcal{F}^{\text{trailer}}, (u,v) \in\ \mathcal{E}^{\text{pairs}} \tag{17}$$

A lorry with three axles can not be coupled to a trailer with three axles,

$$y^{k,l}_{u,v} = 0, \quad u, v \in \mathcal{V}, \tag{18}$$
$$k \in \{k' \in \mathcal{F}^{\text{lorry}} : f^{\text{axle-type}}_{k'} = 3 - \text{axles}\},$$
$$l \in \{l' \in \mathcal{F}^{\text{trailer}} : f^{\text{axle-type}}_{l'} = 3 - \text{axles}\}.$$

2.3.6 Load Constraints

The load of a lorry on an edge can not exceed the lorry's capacity. If a lorry does not traverse an edge, the load of the lorry on that edge is zero,

$$z^k_{u,v} - x^k_{u,v} f^{\text{capacity}}_k \leq 0, \quad u, v \in \mathcal{V},\ k \in \mathcal{F}^{\text{lorry}}. \tag{19}$$

The load of a trailer on an edge can not exceed the trailer's capacity. If a trailer does not traverse an edge, the load of that trailer on that edge is zero.

$$z^l_{u,v} - f^{\text{capacity}}_l \sum_{k\in\mathcal{F}^{\text{lorry}}} y^{k,l}_{u,v} \leq 0, \quad u, v \in \mathcal{V},\ l \in \mathcal{F}^{\text{trailer}}. \tag{20}$$

No supply is left at the customer, hence whilst visiting a customer's supply collection vertex, the sum of the load of the collecting vehicles is increased with at least the amount of supply that the customer has,

$$\sum_{v\in\mathcal{V}} \sum_{k\in\mathcal{F}} \left(z^k_{s_j,v} - z^k_{v,s_j}\right) \geq \mathit{locationSupply}_j, \quad j \in \mathcal{L}^{\text{customer}}. \tag{21}$$

On vertices other than the supply collection vertices, the load of a lorry can stay the same, or decrease in case some of the load is transfered to a trailer,

$$\sum_{u\in\mathcal{V}} \left(z^k_{u,v} - z^k_{v,u}\right) \leq 0, \quad v \in \mathcal{V} \setminus \mathcal{S},\ k \in \mathcal{F}^{\text{lorry}}. \tag{22}$$

On a supply collection vertex, the load of a lorry can not increase more than the supply of the customer,

$$\sum_{v\in\mathcal{V}} \left(z^k_{v,s_j} - z^k_{s_j,v}\right) \leq \mathit{locationSupply}_j, \quad j \in \mathcal{L}^{\text{customer}},\ k \in \mathcal{F}^{\text{lorry}}. \tag{23}$$

Each decouple and couple transshipment vertex pair has, if used, two lorry loads and a trailer load incoming and outgoing. The sum of the incoming loads of such a vertex pair is equal to the sum of the outgoing loads,

$$\sum_{w\in\mathcal{V}}\sum_{k\in\mathcal{F}}\left(z^k_{w,u}-z^k_{u,w}+z^k_{w,v}-z^k_{v,w}\right)=0,\quad (u,v)\in\mathcal{E}^{\text{pairs}} \tag{24}$$

The load of a lorry upon arrival at a trailer start vertex is equal to the load of the trailer and the lorry upon departure,

$$\sum_{u\in\mathcal{V}}\sum_{k\in\mathcal{F}}\left(z^k_{u,v}-z^k_{v,u}\right)=0,\quad v\in\mathcal{N}^+. \tag{25}$$

A trailer can not be decoupled at the trailer's end vertex whilst being loaded more than its capacity,

$$\sum_{u\in\mathcal{V}}\sum_{k\in\mathcal{F}}\left(z^k_{u,v}-z^k_{v,u}\right)\le f^{\text{capacity}}_l,\quad v:=n^-_l,\ l\in\mathcal{F}^{\text{trailer}}. \tag{26}$$

2.3.7 Time Constraints

A lorry can arrive at, but not service, a vertex before the start of the vertex's time window. The maximum speed of a lorry and a trailer together is equal to the minimum of their maximum speeds.

The following function calculates the amount of travel time that it took the vehicles that have traversed edge $(u,v)\in\mathcal{E}$ in path assignment (X,Y),

$$\mathit{travelTime}(u,v,X,Y)=\sum_{k\in\mathcal{F}^{\text{lorry}}}\left(x^k_{u,v}\tau^{\text{travel}}_{u,v,k}+\sum_{l\in\mathcal{F}^{\text{trailer}}}y^{k,l}_{u,v}\tau^{\text{extra}}_{u,v,k,l}\right).$$

A lorry starts a vertex visit as soon as it arrives or as soon as the time window opens, whichever is latest. After a lorry finishes a visit it departs immediately towards the vertex' successor. Hence, given the path assignment (X,Y) and time table T the start time of the visit of vertex $u\in\mathcal{V}$ can be calculated with the following function,

$$\mathit{visitationStart}(u,X,Y,T)=\sum_{v\in\mathcal{V}}\left(\sum_{k\in\mathcal{F}^{\text{lorry}}}\left(t^k_{u,v}-x^k_{u,v}\tau^{\text{visit}}_{u,k}\right)-\mathit{travelTime}(u,v,X,Y)\right).$$

A vertex must be visited early enough such that the visit, which may include loading supply, can be finished before the end of the vertex's time window. If an edge is not traversed by a lorry, the arrival time of that edge for that lorry is zero,

$$\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}}\left(t^k_{u,v}-x^k_{u,v}\left(\tau^{\text{end}}_v-\tau^{\text{visit}}_{v,k}\right)\right)\le 0,\quad u\in\mathcal{V}. \tag{27}$$

The time that a lorry needs between two consecutive vertex arrivals is possibly some waiting time before the visit starts, the duration of the first vertex visit plus the lorry's travel time between the two vertices,

$$\sum_{v\in\mathcal{V}}x^k_{u,v}(\tau^{\text{visit}}_{u,k}+\tau^{\text{travel}}_{u,v,k})\le\sum_{v\in\mathcal{V}}(t^k_{u,v}-t^k_{v,u}),\qquad u\in\mathcal{V}\setminus\mathcal{M}^-,k\in\mathcal{F}^{\text{lorry}}. \tag{28}$$

If a lorry is pulling a trailer whilst traversing an edge its maximum speed could be lowered, hence the following constraints also need to be satisfied,

$$\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} y^{k,l}_{u,v}(\tau^{\text{visit}}_{u,k}+\tau^{\text{travel}}_{u,v,l}) \leq \sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} (t^k_{u,v}-t^k_{v,u}), \qquad u\in\mathcal{V}\setminus\mathcal{M}^-, l\in\mathcal{F}^{\text{trailer}}. \tag{29}$$

Only after a lorry has started and finished decoupling a trailer at a transshipment vertex can a start be made with coupling the trailer again,

$$\mathit{visitationStart}(u,X,Y,T)+\tau^{\text{R}}\sum_{w\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} x^k_{w,u} \leq \mathit{visitationStart}(v,X,Y,T), \quad (u,v)\in\mathcal{E}^{\text{pairs}}. \tag{30}$$

The time a lorry needs between the start of the time window of the vertex it is currently at and arriving at the next vertex is at least the time needed for the current vertex's visit plus the lorry's travel time between the two vertices,

$$\sum_{v\in\mathcal{V}} x^k_{u,v}(\tau^{\text{visit}}_{u,k}+\tau^{\text{travel}}_{u,v,k}) \leq \sum_{v\in\mathcal{V}}(t^k_{i,v}-x^k_{v,u}\tau^{\text{start}}_u), \qquad u\in\mathcal{V}\setminus\mathcal{M}^-, k\in\mathcal{F}^{\text{lorry}}. \tag{31}$$

If a lorry is pulling a trailer, its maximum speed could be lowered, hence the following constraints also need to be satisfied,

$$\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} y^{k,l}_{u,v}(\tau^{\text{visit}}_{u,k}+\tau^{\text{travel}}_{u,v,l}) \leq \sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} (t^k_{u,v}-x^k_{v,u}\tau^{\text{start}}_u), \qquad u\in\mathcal{V}\setminus\mathcal{M}^-, l\in\mathcal{F}^{\text{trailer}}. \tag{32}$$

2.3.8 Symmetry Breaking Constraint

Each transshipment location has η duplicate transshipment vertex pairs. They can be ordered with the following constraint to reduce the solution space,

$$\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} x^k_{v,r^{j,-}_{m+1}} \leq \sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{F}^{\text{lorry}}} x^k_{v,r^{j,-}_m}, \quad m\in\{1,\dots,\eta-1\}, j\in\mathcal{L}^{\text{transshipment}}. \tag{33}$$

There are still many sources of symmetry. Not every vehicle has unique features, hence many vehicles are interchangeable. Furthermore, the ordering of transshipment pair still permits some symmetry. Further removing these sources of symmetry is outside the scope of this thesis and may be interesting for future work.

2.3.9 Unload Constraint

The VRPTTMU is turned into the VRPTT by adding the following constraint,

$$\begin{aligned} x^k_{u,v} &= 0, \\ (u,v) &\in \{(u',v') : u'\in\mathcal{V}\setminus\mathcal{M}^+,\ v'\in\mathcal{N}^+\}\cup \\ &\quad \{(u',v') : u'\in\mathcal{N}^-,\ v'\in\mathcal{V}\setminus\mathcal{M}^-\}, \\ k &\in \mathcal{F}^{\text{lorry}}. \end{aligned} \tag{34}$$

The constraint ensures that a lorry can couple (decouple) a trailer at the depot if and only if it is the first (last) thing it does on its path, which makes multiple unloads impossible.

2.3.10 Decision Variable Constraints

The binary decision variables are constrained as follows,

$$x_{u,v}^{k}, y_{u,v}^{k,l} \in \{0,1\}, \quad u,v \in \mathcal{V},\ k \in \mathcal{F}^{\text{lorry}},\ l \in \mathcal{F}^{\text{trailer}}. \tag{35}$$

The real-valued nonnegative decision variables are constrained as follows,

$$t_{u,v}^{k}, z_{u,v}^{l} \in \mathbb{R}_{+}^{0}, \quad u,v \in \mathcal{V},\ k \in \mathcal{F}^{\text{lorry}},\ l \in \mathcal{F}. \tag{36}$$

2.3.11 Objective Function

The objective function that is minimized to solve the VRPTT and the VRPTTMU is the cost function C, which is the sum of the fixed costs, the distance variable costs and the time variable costs of the vehicles,

$$C(X,Y,T) = C^{\text{fixed}}(X) + C^{\text{distance}}(X,Y) + C^{\text{time}}(X,Y,T), \tag{37}$$

all of which will be defined. The objective function is subject to constraints (7) - (36).

The distance costs function C^{distance} is the sum of the distance costs of all traversed edges by all vehicles,

$$C^{\text{distance}}(X,Y) = \sum_{u,v\in\mathcal{V}} \sum_{k\in F^{\text{lorry}}} \delta_{u,v} \left(x_{u,v}^{k} f_{k}^{\text{distance}} + \sum_{l\in F^{\text{trailer}}} y_{u,v}^{k,l} f_{l}^{\text{distance}} \right). \tag{38}$$

The time costs function C^{time} is the the sum of the time cost of all vehicles,

$$\begin{aligned} C^{\text{time}}(X,Y,T) = &\sum_{k\in\mathcal{F}^{\text{lorry}}} f_{k}^{\text{time}} \left(\mathit{visitationStart}(m_k^-, X, Y, T) + \tau_{m_k^-}^{\text{visit}} - \mathit{visitationStart}(m_k^+, X, Y, T) \right) \\ &\sum_{l\in\mathcal{F}^{\text{trailer}}} f_{l}^{\text{time}} \left(\mathit{visitationStart}(n_l^-, X, Y, T) + \tau_{n_l^-}^{\text{visit}} - \mathit{visitationStart}(n_l^+, X, Y, T) \right). \end{aligned} \tag{39}$$

The fixed costs function C^{fixed} is the the sum of the fixed costs of all used vehicles,

$$C^{\text{fixed}}(X) = \sum_{k\in\mathcal{F}^{\text{lorry}}} \left((1 - x_{m_k^+, m_k^-}^{k}) \cdot f_{k}^{\text{fixed}} + \sum_{v\in\mathcal{V}} \sum_{l\in\mathcal{F}^{\text{trailer}}} x_{v,n_l^+}^{k} \cdot f_{l}^{\text{fixed}} \right). \tag{40}$$

A lorry counts as being used when it has other vertices on its path than its start and end vertex. A trailer is considered as being used when it has been coupled.

3 Optimization Method

The optimization method used in this thesis is a variation of the variable neighborhood search (VNS). Multiple neighborhood structures are used in the search to avoid getting stuck in local optima. This section describes how this heuristic is used to obtain solutions that satisfy the model constraints.

The optimization method includes finding solutions satisfying only a subset of the model constraints. Coverage of which constraints may be violated in this process will be given in Section 3.1. In Section 3.2 neighborhood structures are described which are used to find new path assignments. In Section 3.3 it is shown how to cast the calculation of a load table related to a path assignment as a maximum flow problem. In Section 3.4 the method used to calculate a load table related to a path assignment by solving a longest path problem is described. In Section 3.5 the overall structure of the algorithm is described.

3.1 Constraint Violations

In the process of finding solutions that satisfy all constraints it is helpful to also consider solutions that only satisfy a subset of the constraints. The constraints that may be broken by solutions are called the soft constraints. The penalties that are incurred by breaking the soft constraints are covered below. All constraints not mentioned below are hard constraints.

If in a path assignment not all customers are served, constraint (9) is broken. Let $\omega^{\text{unserved}-\text{customer}}$ be the penalty incurred per unvisited customer vertex and let

$$U^{\text{unserved}-\text{customer}}(X,Y) = \omega^{\text{unserved}-\text{customer}} \left(|\mathcal{S}| - \sum_{u\in\mathcal{V}} \sum_{v\in\mathcal{S}} \sum_{k\in\mathcal{F}^{\text{lorry}}} x^k_{u,v} \right) \tag{41}$$

be the total of the penalties incurred by not visiting customers' supply collection vertices.

If in a path assignment a lorry customer is visited with a trailer, constraint (11) is broken. Let $\omega^{\text{lorry}-\text{customer}}$ be the penalty incurred per lorry customer visited with a trailer and let

$$U^{\text{lorry}-\text{customer}}(X,Y) = \omega^{\text{lorry}-\text{customer}} \sum_{u\in\mathcal{V}} \sum_{v\in\mathcal{S}^{\text{lorry}}} \sum_{k\in\mathcal{F}^{\text{lorry}}} \sum_{l\in\mathcal{F}^{\text{trailer}}} y^{k,l}_{u,v} \tag{42}$$

be the total of the penalties incurred by visiting lorry customers with trailers.

If in a time table the end of a vertex' time window is broken, then one constraint of the set of constraints (27) is broken. Let $\omega^{\text{time}-\text{window}}$ be the penalty incurred per unit time and let

$$U^{\text{time}-\text{window}}(X,Y,T) = \omega^{\text{time}-\text{window}} \sum_{u,v\in\mathcal{V}} \sum_{k\in\mathcal{F}^{\text{lorry}}} \max\left(0, t^k_{u,v} - x^k_{u,v}\left(\tau^{\text{end}}_v - \tau^{\text{visit}}_{v,k}\right)\right) \tag{43}$$

be the total of the penalties incurred by violating the ends of the vertices' time windows.

If in a load table the vehicles do not have enough capacity to load the all of the supply of the visited customers, a constraint of the set of constraints (21) is broken. Let $\omega^{\text{capacity}-\text{shortage}}$ be the penalty incurred per unit supply that is not picked up of the visited customers and let

$$U^{\text{capacity}-\text{shortage}}(X,Y,L) = \omega^{\text{capacity}-\text{shortage}} \sum_{j\in\mathcal{L}^{\text{customer}}} \sum_{u\in\mathcal{V}} \left(\sum_{k\in\mathcal{F}^{\text{lorry}}} x^k_{u,s_j} loc^{\text{supply}}_j - \sum_{k\in\mathcal{F}} \left(z^k_{s_j,u} - z^k_{u,s_j} \right) \right) \tag{44}$$

be the sum of the penalties incurred by leaving behind supply at the visited supply collection vertices.

The penalty U of a solution (X,Y,L,T) is the sum of all incurred penalties,

$$\begin{aligned} U(X,Y,L,T) = U^{\text{unserved}-\text{customer}}(X,Y,L,T) + U^{\text{lorry}-\text{customer}}(X,Y) + \\ U^{\text{time}-\text{window}}(X,Y,T) + U^{\text{capacity}-\text{shortage}}(X,Y,L) \end{aligned} \tag{45}$$

A solution to the VRPTT should satisfy all constraints, i.e.,constraints (7) - (36). A solution to the VRPTTMU should satisfy the same constraints except for constraint (34). The hard constraints consist of all model constraints other than soft constraints (9), (11), (21), and (27).

3.2 Neighborhoods

In this section multiple neighborhood operators will be introduced. A neighborhood operator takes as input a path assignment and generates new path assignments by changing some part of the input path assignment. These new path assignments are filtered out if they break any hard constraints. Some operators are designed to yield improved path assignments. Some are designed to enable other operators to yield improved path assignments. The following operators can be applied to a path assignment to find new path assignments:

- *Move customer:* For each customer's supply collection vertex and for each possible position on each lorry's path the operator yields the path assignment with the supply collection vertex inserted in that position and removed from its previous position.
- *Remove customer:* For each visited customer's supply collection vertex the operator yields a path assignment in which the vertex is removed from the lorry's paths.
- *Insert trailer:* For each lorry that is not coupled to a trailer during its path and for each unused trailer the operator yields a path assignment in which the trailer is inserted into the lorry's path. A trailer is inserted in a lorry's path by inserting the trailer's start and end vertex on the first position after the lorry's start vertex and the last position before the lorry's end vertex respectively.

- *Remove trailer:* For each used trailer the operator yields a path assignment in which the trailer is removed. A trailer is removed by removing all couple and decouple vertices visited by that trailer from the lorries' paths, this includes the trailer's start and end vertex.

- *Insert consecutive pair:* For each position in the path of any lorry at which a trailer is pulled and for each transshipment location the operator yields a path assignment in which the vertices of a transshipment pair of that transshipment location are inserted consecutively at that position. This operator functions as an enabler of operators *move customer* and *share trailer*.

- *Remove consecutive pair:* For each transshipment decouple vertex that is immediately followed by its paired couple vertex in any lorry's path, yield a path assignment in which those two vertices are removed from the lorry's path. This operator functions as a clean-up.

- *Remove lorry:* For each used lorry the operator yields a path assignment in which that lorry is removed. A lorry is removed by removing all vertices from the lorry's path except for its own start and end vertex and by removing any trailer that was pulled by the lorry.

- *Swap trailer:* For each used trailer and for each unused trailer the operator yields a path assignment in which the used trailer is swapped for the unused trailer. A trailer and an unused trailer are swapped by swapping the start and end vertices of the used trailer with the start and end vertices of the unused trailer.

- *Swap lorry:* For each used lorry and for each unused lorry the operator yields a path assignment in which the used lorry is swapped with the unused lorry. A used and an unused lorry are swapped by moving all vertices of the used lorry's path except for its own start and vertex, to the unused lorry's path.

- *Share trailer:* Let it be given that a lorry k_1 visits both the decouple $i \in \mathcal{R}^-$ and couple vertex $j \in \mathcal{R}^+$ of a transshipment pair. In that case there must have been a couple vertex $u \in \mathcal{R}^+ \cup \mathcal{N}^+$ that was a predecessor of i in the path of lorry k_1 and a decouple vertex $v \in \mathcal{R}^- \cup \mathcal{N}^-$ that is a successor of j in the path of lorry k_1. The trailer can be shared with another lorry k_2 by moving vertices u and i to the path of lorry k_2 such that u precedes i and no other trailer is already pulled on that part of k_2's path. Alternatively, the trailer could have been shared by moving vertices j and v to the path of lorry k_2 such that j precedes v and no other trailer is already pulled on that part of k_2's path. For each transhipment pair of which both vertices are visited by the same lorry k and for each other lorry k_2 the operator yields a path assignments for both ways described above that k and k_2 can share a trailer.

- *Merge lorries:* For each pair of lorries the operator yields $nSamples \in \mathbb{N}$ path assignment in which the pair of lorries is merged. A pair of lorries is merged by first removing all trailers that were pulled by them. Then all of the supply collection vertices in their paths are moved in a random order to one of the the two lorries. A random trailer is inserted into that lorry's path and lastly transshipment vertices are inserted such that the trailer is decoupled and coupled before and after every lorry customer at the nearest transshipment vertex pair .

3.3 Load Table

A load table L indicates how much load a vehicle is carrying whilst traversing an edge. Given a path assignment (X, Y) that satisfies the hard constraints a related load table L can be constructed that also satisfies the hard constraints. The load table does not influence the costs of a solution. It is possible that for a path assignments that satisfies the soft constraints no related load table exists in which the whole supply of each customer is collected. In that case the load table violates a soft constraint and incurs the penalty described in equation (44).

We find a load table L by modeling the problem as a maximum flow problem in a capacity graph. The capacity graph needs to model the capacities of the vehicles, the amount of supply of the customers, the flow of the customer supply from the customers to the lorries, whether and where load flows from lorries to trailers and the flow of load from the vehicles to the depot.

Let $\mathcal{G}^{\mathrm{c}} = (\mathcal{V}^{\mathrm{c}}, \mathcal{E}^{\mathrm{c}}, \mathit{capacity})$ be the capacity graph, where $\mathcal{V}^{\mathrm{c}}$ is a set of vertices, $\mathcal{E}^{\mathrm{c}}$ is a set of edges and $\mathit{capacity} : \mathcal{E}^{\mathrm{c}} \to \mathbb{R}^{+}$ is a mapping from the edges to their capacities.

The capacity graph $\mathcal{G}^{\mathrm{c}}$ contains all vertices of graph $\mathcal{G}$, i.e., $\mathcal{V} \subset \mathcal{V}^{\mathrm{c}}$. Let $\mathcal{P}$ be a set that contains one vertex for each trailer and such that $\mathcal{P} \cap \mathcal{V} = \emptyset$. The capacity graph contains for each trailer $l \in \mathcal{F}^{\text{trailer}}$ a vertex $p_l \in \mathcal{P}$ to model the flow of load from a lorry to the trailer and from the trailer to the depot, i.e., $\mathcal{P} \subset \mathcal{V}^{\mathrm{c}}$. See Figure 6 for an illustration of a capacity graph where set $\mathcal{P}$ contains only one element, the vertex named P. Let $\mathit{source} \in \mathcal{V}^{\mathrm{c}}$ be a source vertex that functions as a supersource for the supply of the customers. It is illustrated in Figure 6 with the vertex "source". Let $\mathit{sink} \in \mathcal{V}^{\mathrm{c}}$ be a sink vertex that functions as a supersink for the load of the lorries and trailers. It is illustrated in Figure 6 with the vertex "sink".

Each edge $(u, v) \in \mathcal{E}$ that is traversed by a lorry $k \in \mathcal{F}^{\text{lorry}}$ is also an edge in the capacity graph, i.e., $\{(u, v) \in \mathcal{E} : \sum_{k \in \mathcal{F}^{\text{lorry}}} x^{k}_{u,v} = 1\} \subset \mathcal{V}^{\mathrm{c}}$, with a capacity $\mathit{capacity}((u, v)) = f^{\text{capacity}}_{k}$ equal to lorry k's capacity .

For each vertex $u \in \mathcal{V}$ visited by a lorry and a trailer $l \in \mathcal{F}^{\text{trailer}}$ the capacity graph contains an edge (u, p_l), $p_l \in \mathcal{P}$ with a capacity equal to the trailer's capacity $\mathit{capacity}((u, p_l)) := f^{\text{capacity}}_{l}$ to model whether load was transfered to the trailer there, i.e., $\{(u, p_l) : \sum_{v \in \mathcal{V}} \sum_{k \in \mathcal{F}^{\text{lorry}}} \left(y^{k,l}_{u,v} + y^{k,l}_{v,u}\right) > 0, u \in \mathcal{V}, p_l \in \mathcal{P}, l \in \mathcal{F}^{\text{trailer}}\} \subset \mathcal{E}^{\mathrm{c}}$.

The capacity graph contains an edge from the supersource source to each customer's supply collection vertex $s_i \in \mathcal{S}, i \in \mathcal{L}^{\text{customer}}$, with a capacity $\mathit{capacity}((\mathit{source}, s_i)) := \mathit{locationSupply}_i$ equal to the customer's supply, i.e.,

$\{(source, s_i) : s_i \in \mathcal{S}\} \subset \mathcal{E}^{\mathrm{c}}$.

The capacity graph contains an edge from the end of path vertex $m_k^- \in \mathcal{M}^-$ of each lorry $k \in \mathcal{F}^{\text{lorry}}$ to the supersink *sink* with a capacity $\mathit{capacity}((m_k^-, sink)) := f_k^{\text{capacity}}$ equal to the lorry's capacity, i.e., $\{(m_k^-, sink) : k \in \mathcal{F}^{\text{lorry}}\} \subset \mathcal{E}^{\mathrm{c}}$.

The capacity graph contains an edge from the vertex $p_l \in \mathcal{P}$ of each trailer $l \in \mathcal{F}^{\text{trailer}}$ to the supersink *sink* with a capacity $\mathit{capacity}((p_l, sink)) := f_l^{\text{capacity}}$ equal to the trailer's capacity, i.e., $\{(p_l, sink) : l \in \mathcal{F}^{\text{trailer}}\} \subset \mathcal{E}^{\mathrm{c}}$.

A feasible solution to the maximum flow problem from source *source* to sink *sink* on capacity graph $\mathcal{G}^{\mathrm{c}}$ is a mapping $f^{\mathrm{c}} : \mathcal{E}^{\mathrm{c}} \to \mathbb{R}_0^+$ representing the flow on each edge $(u, v) \in \mathcal{E}^{\mathrm{c}}$. Let $|f^{\mathrm{c}}| = \sum_{v \in \mathcal{V}^{\mathrm{c}}} f^{\mathrm{c}}(source, v)$ be the total amount of flow from the source vertex, which is equal to the amount of supply that was collected from the customers and unloaded at the depot.

Load table L can be constructed from f^{c}. We need the load of each vehicle on each traversed edge. The values concerning the load of the lorries are simply,

$$z_{u,v}^k = x_{u,v}^k \cdot f^{\mathrm{c}}((u, v)), \quad u, v \in \mathcal{V}, k \in \mathcal{F}^{\text{lorry}}.$$

The loads of the trailers need to be identified using another method. The amount of load that is transfered from a lorry to a trailer $l \in \mathcal{F}^{\text{trailer}}$ that visit vertex $v \in \mathcal{V}^{\mathrm{c}}$ together, is equal to $f^{\mathrm{c}}((v, p_l))$. To calculate the load of a trailer l when departing from a vertex v, we need to know the flow of load $f^{\mathrm{c}}(v, p_l)$ to trailer l at vertex v and the flow of load to trailer l at all vertices it visited previous to v.

The sequence in which trailer l visited vertices needs to be determined. Let $\mathcal{E}^{\text{used}-\text{trailer}} = \{(u, v) \in \mathcal{E} : \sum_{k \in \mathcal{F}^{\text{lorry}}} \sum_{l \in \mathcal{F}^{\text{trailer}}} y_{u,v}^{k,l} = 1\}$ be the set of edges traversed by a trailer and a lorry. Let $\mathcal{G}^{\text{trailer}} = (\mathcal{V}, \mathcal{E}^{\text{pairs}} \cup \mathcal{E}^{\text{used}-\text{trailer}})$ be a graph that can be used to determine the order in which trailers visit vertices .

Let the trailer path $\mathit{trailerPath}_l$ of trailer $l \in \mathcal{F}^{\text{trailer}}$ be the simple path in $\mathcal{G}^{\text{trailer}}$ from n_l^+ to n_l^-. The order of the vertices on the path can be used to determine the total amount of load that has flown from lorries to trailer l at and before visiting vertex u, namely $\sum_{v \in \mathit{trailerPath}_l}^{u} f^{\mathrm{c}}((v, p_l))$.

The values of load table L can thus be determined as,

$$z_{u,v}^k = \begin{cases} \sum_{h \in \mathcal{F}^{\text{lorry}}} y_{u,v}^{h,k} \cdot \sum_{v \in \mathit{trailerPath}_k}^{u} f^{\mathrm{c}}((v, p_h)), & \text{if } k \in \mathcal{F}^{\text{trailer}}, \\ x_{u,v}^k \cdot f^{\mathrm{c}}((u, v)), & \text{if } k \in \mathcal{F}^{\text{lorry}}, \end{cases} \quad u, v \in \mathcal{V}, k \in \mathcal{F} \tag{46}$$

If path assignment (X, Y) satisfies the hard constraints this procedure yields a related load table L that also satisfies the hard constraints.

3.4 Time Table

A time table T indicates at what time a lorry finishes traversing an edge. From a path assignment (X, Y) that satisfies the hard constraints we can construct a related time table T that also satisfies the hard constraints. We want to construct a related time table T with four goals in mind. First, the time table should not break the time related hard constraints (28) - (32). Second, the time table must satisfy the time related soft constraint (27) if possible, i.e., each vertex visit should end before the ending of the vertex's time window. If this soft constraint is not satisfied by the time table the penalty described in (43) will be incurred. Third, if violating a time window is unavoidable, the violation should be as small as possible. Fourth, the time table should be such that the time related costs are as small as possible.

We find a time table T related to path assignment (X, Y) by first calculating the earliest time at which the lorries can arrive at the vertices on their paths without breaking any vertex' opening time. We mark the vertices at which the lorries end their paths and the vertices that already have their closing time violated. Then we calculate the latest time at which the lorries can start their paths and arrive at the vertices in their paths whilst arriving at the earliest arrival time at the marked vertices and without violating any unmarked vertex' closing time. These latest path start times are used as the start times of the lorries in the time table. The time table is then built up according to two rules. First, lorries start the visit of a vertex as soon as they arrive or as soon as the vertex opens, whichever happens the latest. Second, if a vertex visit is finished the lorry departs immediately towards the next vertex.

3.4.1 Earliest Arrival

We calculate the earliest time at which lorries can arrive at the vertices in their paths by casting the problem as a longest path problem in a precedence graph $\mathcal{G}^{\text{earliest}} = (\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{earliest}}, \mathit{duration}^{\text{earliest}})$, where $\mathcal{V}^{\text{precedence}}$ is a set of vertices, $\mathcal{E}^{\text{earliest}}$ is a set of edges, and $\mathit{duration}^{\text{earliest}} : \mathcal{E}^{\text{earliest}} \to \mathbb{R}_0^+$ is a mapping from the edges $\mathcal{E}^{\text{earliest}}$ to their weights, which represent durations.

The precedence graph $\mathcal{G}^{\text{earliest}}$ needs to model the following constraints:

1. The opening times of vertices.
2. The durations of vertex visits.
3. The order in which a lorry visits vertices according to its path with the appropriate travel times.
4. The precedence a decouple vertex of a transshipment vertex pair has over the couple vertex it is paired with.

Since we need to model the duration of a vertex visit, but vertices in a precedence graph do not have weights, we need to employ a trick. Let $\mathcal{V}^{\text{start}}$ be a set of visitation start vertices and let $\mathcal{V}^{\text{end}}$ be a set of visitation end vertices. For each vertex $v \in \mathcal{V}$ let $v^{\text{start}} \in \mathcal{V}^{\text{start}}$ be a vertex that represents the start of v's visitation and let $v^{\text{end}} \in \mathcal{V}^{\text{end}}$ be a vertex that represents the end of v's visitation. The duration of v's visit can then be modeled with the weight of edge $(v^{\text{start}}, v^{\text{end}})$. Let $\mathit{source}, \mathit{sink}$ be the supersource and supersink vertex respectively. Then let $\mathcal{V}^{\text{precedence}} := \mathcal{V}^{\text{start}} \cup \mathcal{V}^{\text{end}} \cup \{\mathit{source}, \mathit{sink}\}$.

For each vertex v visited by lorry k, i.e., for each $(v, k) \in \{(v', k') : \sum_{w \in \mathcal{V}} \left(x^{k'}_{v',w} + x^{k'}_{w,v'} \right) > 0,\ v' \in \mathcal{V},\ k' \in \mathcal{F}^{\text{lorry}}\}$, the following holds:

- $(\mathit{source}, v^{\text{start}}) \in \mathcal{E}^{\text{earliest}}$ with edge weight $\mathit{duration}^{\text{earliest}}((\mathit{source}, v^{\text{start}})) := \tau^{\text{start}}_v - \tau^{\min}$. This covers bullet point 1.
- $(v^{\text{start}}, v^{\text{end}}) \in \mathcal{E}^{\text{earliest}}$ with edge weight $\mathit{duration}^{\text{earliest}}((v^{\text{start}}, v^{\text{end}})) := \tau^{\text{duration}}_{v,k}$. This covers bullet point 2.

For each edge (u, v) traversed by lorry k, i.e.,for each $(u, v, k) \in \{(u', v', k') : x^{k'}_{u',v'} = 1,\ u', v' \in \mathcal{V},\ k' \in \mathcal{F}^{\text{lorry}}\}$, the following holds:

- $(u, v) \in \mathcal{E}^{\text{earliest}}$ with edge weight $\mathit{duration}^{\text{earliest}}((u, v)) := \mathit{travelTime}(u, v, X, Y)$. This covers bullet point 3.

For each transshipment vertex pair $(u, v) \in \mathcal{E}^{\text{pairs}}$, the following holds:

- $(u^{\text{end}}, v^{\text{start}}) \in \mathcal{E}^{\text{earliest}}$ with edge weight $\mathit{duration}^{\text{earliest}}((u^{\text{end}}, v^{\text{start}})) := 0$. This covers bullet point 4.

Now we can solve a longest path problem on precedence graph G^{earliest} which yields a mapping $f^{\text{earliest}} : \mathcal{V}^{\text{precedence}} \times \mathcal{V}^{\text{precedence}} \to \mathbb{R}^+_0$ that returns the length of the longest path in G^{earliest} from u to v for $u, v \in \mathcal{V}^{\text{precedence}}$ or zero if there is no such path .

3.4.2 Latest Departure

To calculate the latest time at which lorries can start their paths we cast the problem as a longest path problem in a precedence graph $\mathcal{G}^{\text{latest}} = (\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{latest}}, \mathit{duration}^{\text{latest}})$, where $\mathcal{E}^{\text{latest}}$ is a set of edges and $\mathit{duration}^{\text{latest}} : \mathcal{E}^{\text{latest}} \to \mathbb{R}^+_0$ is a mapping from the edges $\mathcal{E}^{\text{latest}}$ to their weights, which represent durations. The latest time at which a lorry $k \in \mathcal{F}^{\text{lorry}}$ can start its path can be discovered by finding the longest path from the supersink sink to the vertex $m^{+\text{start}}_k$ that represents the beginning of the visit of the vertex where the lorry start its path.

The precedence graph $\mathcal{G}^{\text{latest}}$ needs to model the following constraints:

1. The closing times of vertices.
2. The durations of vertex visits.
3. The order in which a lorry visits vertices according to its path with the appropriate travel times.
4. The precedence a decouple vertex of a transshipment vertex pair has over the couple vertex it is paired with.
5. The time at which lorries must end their paths, which is at the earliest possible time.

6. The time at which lorries must visit vertices whose closing time is inevitably violated given path assignment (X, Y), which is at the earliest possible time.

For each vertex v visited by lorry k, i.e., for each $(v, k) \in \{(v', k') : \sum_{w \in \mathcal{V}} \left(x^{k'}_{v',w} + x^{k'}_{w,v'} \right) > 0,\ v' \in \mathcal{V},\ k' \in \mathcal{F}^{\text{lorry}}\}$, the following holds:

- $(sink, v^{\text{end}}) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((sink, v^{\text{end}})) := \tau^{\max} - \tau^{\text{end}}_v$. This covers bullet point 1.
- $(v^{\text{end}}, v^{\text{start}}) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((v^{\text{end}}, v^{\text{start}})) := \tau^{\text{duration}}_{v,k}$. This covers bullet point 2.

For each edge (u, v) traversed by lorry k, i.e., for each $(u, v, k) \in \{(u', v', k') : x^{k'}_{u',v'} = 1,\ u', v' \in \mathcal{V},\ k' \in \mathcal{F}^{\text{lorry}}\}$, the following holds:

- $(v, u) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((v, u)) := travelTime(u, v, X, Y)$. This covers bullet point three.

For each transshipment vertex pair $(u, v) \in \mathcal{E}^{\text{pairs}}$, the following holds:

- $(v^{\text{start}}, u^{\text{end}}) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((v^{\text{start}}, u^{\text{end}})) := 0$. This covers bullet point four.

For each lorry $k \in \mathcal{F}^{\text{lorry}}$ the following holds:

- $(sink, m^-_k) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((sink, m^-_k)) := \tau^{\max} - f^{\text{earliest}}(source, m^-_k)$. This covers bullet point 5.

For each vertex whose closing time is already violated if it is visited at the earliest possible time $v^{\text{end}} \in \{u^{\text{end}} : f^{\text{earliest}}(source, u^{\text{end}}) > \tau^{\text{end}}_u,\ u^{\text{end}} \in \mathcal{V}^{\text{end}}\}$ the following holds:

- $(sink, v^{\text{end}}) \in \mathcal{E}^{\text{latest}}$ with edge weight $duration^{\text{latest}}((sink, v^{\text{end}})) := \tau^{\max} - f^{\text{earliest}}(source, v^{\text{end}})$. This covers bullet point 6.

Now we can solve a longest path problem on precedence graph G^{latest} which yields a mapping $f^{\text{latest}} : \mathcal{V}^{\text{precedence}} \times \mathcal{V}^{\text{precedence}} \to \mathbb{R}^+_0$ that returns the length of the longest path in G^{latest} from u to v for $u, v \in \mathcal{V}^{\text{precedence}}$ or zero if there is no such path. The latest time at which a lorry k can start its path is then $\tau^{\max} - f^{\text{latest}}(sink, m^{+\text{start}}_k)$.

3.4.3 Generating the Time Table

The values of the time table can now be generated by calculating the earliest arrival times again but this time the lorries can start their paths no earlier than the just calculated latest start times.

Let precedence graph $\mathcal{G}^{\text{result}} = (\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{earliest}}, duration^{\text{result}})$, where $duration^{\text{result}} : \mathcal{E}^{\text{earliest}} \to \mathbb{R}^+_0$ is a mapping from the edges $\mathcal{E}^{\text{earliest}}$ to their weights, which represent durations.

Let $\mathcal{E}^{\text{latest}-\text{start}} := \{(source, v^{\text{start}}) : v \in \mathcal{M}^+\} \subset \mathcal{E}^{\text{earliest}}$ be the set of edges whose weigths we change to constrain the earliest departure times of the lorries whilst calculating the earliest arrival times,

$$duration^{\text{result}}(u,v) := \begin{cases} \tau^{\max} - f^{\text{latest}}(sink, v), & \text{if } (u,v) \in \mathcal{E}^{\text{latest}-\text{start}}, \\ duration^{\text{earliest}}(u,v), & \text{if } (u,v) \in \mathcal{E}^{\text{earliest}} \setminus \mathcal{E}^{\text{latest}-\text{start}}. \end{cases}$$

Now we can solve a longest path problem on precedence graph G^{result} which yields a mapping $f^{\text{result}} : \mathcal{V}^{\text{precedence}} \times \mathcal{V}^{\text{precedence}} \to \mathbb{R}_0^+$ that returns the length of the longest path in G^{latest} from u to v for $u, v \in \mathcal{V}^{\text{precedence}}$ or zero if there is no such path.

Time table T related to path assignment (X,Y) can be constructed from the mapping f^{result} as follows:

$$t_{u,v}^k = x_{u,v}^k \left(f^{\text{result}}(source, u^{\text{end}}) + travelTime(u,v,X,Y)\right), \quad u,v \in \mathcal{V}, k \in \mathcal{F}^{\text{lorry}}.$$

For the VRPTT this time table fulfills all four goals we set out to achieve with the time table. For the VRPTTMU it may be possible to further reduce the time related costs of the trailers whilst satisfying the hard and the soft constraints. Developing such a procedure is outside the scope of this thesis. When considering the VRPTT, the trailers' time related costs are minimized by minimizing the durations of the lorries' paths.

3.5 Algorithm

In this section the VNS heuristic that used to generate solutions to the VRPTT and the VRPTTMU is described.

A few concepts need to be explained first. A priority queue PQ is a collection for items that efficiently keeps the items sorted with respect to a certain priority function of the items' attributes, whilst adding or removing items from it. The VNS uses the priority queue to store solutions whose neighborhoods have not been searched yet and uses their scores to sort them. Solutions with lower scores will have their neighborhoods searched first for new solutions. The score $score(X,Y,L,T) = C(X,Y,T) + U(X,Y,L,T)$ of a solution (X,Y,L,T) is the sum of its costs and its penalties. The tabu set $TABU$ is a collection for items that can efficiently store items and check whether an item is a member of it. The VNS uses a tabu set to store path assignments and to check whether a path assignment has been seen found before.

The VNS starts with an initial solution in the priority queue PQ and with the solution's path assignment in the tabu set $TABU$. The initial solution can just be the solution where every lorry goes from its start vertex straight to its end vertex. With probability $p \in (0,1)$ a random path assignment (X,Y) is removed from the priority queue, otherwise the path assignment with the lowest score is removed from the priority queue. A set of new path assignments is generated by applying all neighborhood operators to the path assignment (X,Y). The VNS then filters out any path assignment that breaks a hard constraint. The

VNS filters out any path assignments that is already a member of the tabu set to reduce the amount of redundant computation. The solution (X, Y, L, T) and the score $score(X, Y, L, T)$ will be calculated for each path assignment that is not filtered out. These solutions will be compared with the best solution found thus far and added to the priority queue. This process repeats itself until a stopping condition is satisfied or until the priority queue is empty. The solution with the lowest cost and without penalty is then returned.

4 Tests and Results

This section first provides details about the implementation and the used instances. Thereafter the results are described and interpreted.

4.1 Instances

This subsection describes the set of VRPTT test instances from [9] structured to resemble the situation in raw milk collection.

The customer and transshipment locations are randomly selected on a 100 by 100 kilometer grid with the depot located in the center. The distances between the vertices is equal to the euclidean distances increased by 30 % and rounded up the nearest integer. The amount of customer supplies are chosen randomly from the interval $[1000, 10000]$ such that even the smallest vehicle can hold the whole supply of any customer. The instances have a planning horizon of 1320 minutes. The time window of each location is equal to the whole planning horizon, hence the interval [0,1320]. These nonrestrictive time windows correspond to the situation in raw milk collection where few customers or transshipment locations have time windows.

With these parameters, 30 instances were created per instance size, where the instance size varied between two and 25 customers. The amount of transshipment locations is always equal to the amount of customers. Half of the customers consist of trailer customers, the other half of lorry customers. Half of the transshipment locations consist of trailer customers, the other half of pure transshipment locations.

The cut-off distance ϕ is five kilometers. The duration of visiting a transshipment vertex τ^{R} is five minutes. The values used for time and distance are nonnegative integers. In a note accompanying the instances Drexl describes that the travel time in minutes between a pair of vertices is computed by dividing the distance by the driving speed and multiplying it by 60. Only then should the value be rounded off to the nearest integer.

The instances use two lorry and two trailer classes as specified in Table 3. Enough vehicles of each class are included such that all of the customer's supply can be collected with either only class 1 lorries and class 2 trailers, or class 2 lorries and class 1 trailers.

For this thesis the available fleet of each instance has been modified by adding a synthetic trailer class to facilitate multiple unloads. For each customer in an instance a synthetic trailer is added.

Trailers will still incur their fixed costs if they are used as unloading docks, even though de facto no trailer is used. Furthermore, using a trailer as unloading dock reduces the amount of available trailers that can be used as actual trailers. To resolve these two issues, synthetic trailers without fixed costs, with infinite distance costs per distance unit and an infinite capacity are added to the fleet of the instances. These trailers will never be used as actual trailer due to their distance costs. In the VRPTT these synthetic trailers don't have an impact on the best found solutions. If a lorry would couple such a synthetic trailer, it will have to be the first event on its path after starting its path. After coupling the trailer the lorry will either move away from the depot and incur infinite distance costs or go to the synthetic trailer's decouple vertex after which the lorry will have to end its path without having contributed at all to the collection of customer supply.

Table 3: Vehicle Parameters

Vehicle class	Number of axles	Fixed cost	Cost per km	Cost per hour	Capacity	Driving speed short distance	Driving speed long distance	Load transfer time in minutes per 1000 units
Lorry 1	2	180	0.65	36	10000	25	65	2
Lorry 2	3	200	0.70	36	15000	25	65	2
Trailer 1	2	20	0.04	0	10000	25	65	
Trailer 2	3	25	0.06	0	15000	25	65	
Synthetic Trailer	2	0	∞	0	∞	25	65	

4.2 Bounds

Upper bounds (VRPTT_UB) and lower bounds (VRPTT_LB) on the minimum costs for the VRPTT are provided in [9] for instances with up to eight customers. For all instances with two and four customers, the optimal value was found. The cost function that is used in [9] excludes time related costs, i.e., the bounds are on the sum of the fixed and the distance costs. Optimal values for the truck and trailer routing problem (TTRP) for the instances used in [9] are provided in [19]. These values are referred to as TTRP_OPT. The costs function that is used in [19] includes time related costs. The TTRP is a special version of the VRPTT, where trailers are not allowed to be shared, i.e., a trailer can not be decoupled by one lorry and coupled again by another lorry. The values of the bounds and the values found by the VNS are all included in Appendix B.1.

4.3 Implementation Details

The results reported in this section were obtained by running the optimization method three times. The algorithm was stopped after three minutes or 3000 iterations, whichever came first. In Table 4 the used parameters are given. These parameters were found to be effective without performing a hyperparameter optimization, so better parameters can probably be found. The maximum flow algorithm that is used to calculate the load table is the push-relabel algorithm [20]. The longest path algorithm that is used to calculate the time table is the Floyd-Warshall algorithm [21].

Table 4: Method parameters

Parameter	Value
τ^{D}	0
p	0.2
$nSamples$	10
η	3
$\omega^{\mathrm{unserved-customer}}$	40000
$\omega^{\mathrm{lorry-customer}}$	20000
$\omega^{\mathrm{time-window}}$	2
$\omega^{\mathrm{capacity-shortage}}$	4

4.4 Results

In this subsection we interpret the results found by the VNS. The time related cost are included when the solutions found by the VNS for the VRPTT and VRPTTMU are compared with the TTRP_OPT and excluded when compared with the VRPTT_LB and VRPTT_UB. For each instance the available bounds and the values found by the VNS are plotted in Figure 11. Statistic of the gaps between the values found by the VNS and the available bounds are given in Table 5.

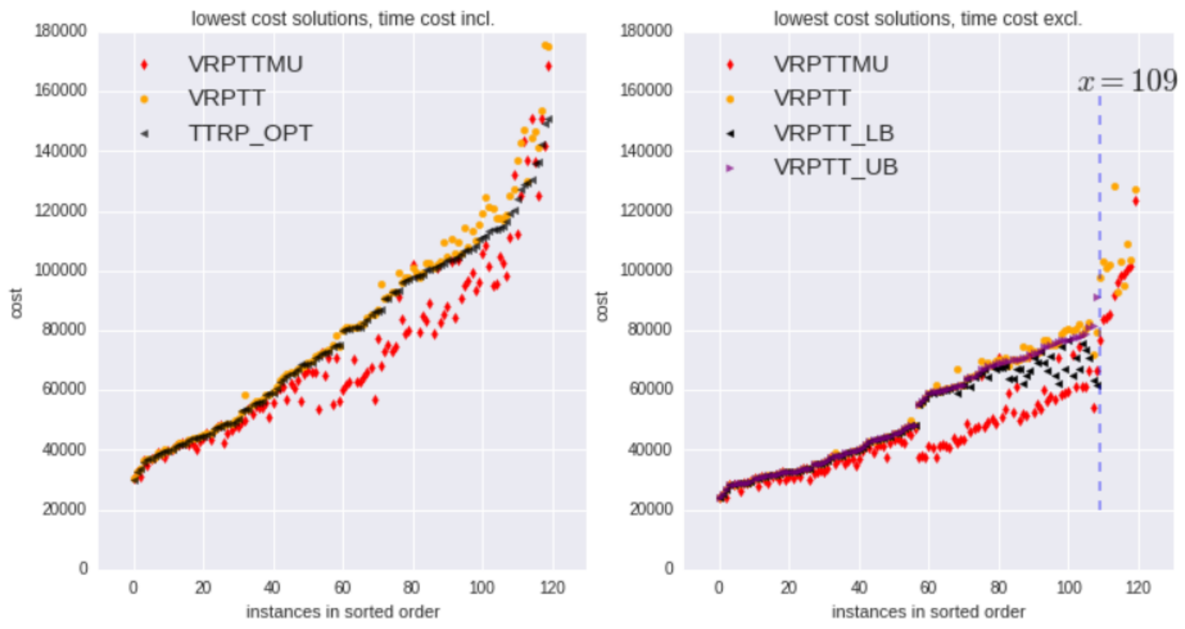

Figure 11: The lowest cost found by the VNS for each instance with the available bounds. The left plot excludes time cost. Its instances are sorted such that the values of TTRP_OPT increase monotonically. The plot on the left includes time cost. Its instances are sorted such that all 109 known values of VRPTT_UB increase monotonically.

	TTRP_OPT, time cost incl.		VRPTT_LB, time cost excl.		VRPTT_UB, time cost excl.	
	VRPTT	VRPTTMU	VRPTT	VRPTTMU	VRPTT	VRPTTMU
2 customers						
count	30.0	30.0	30.0	30.0	30.0	30.0
mean	0.0	-3.5	-0.0	-4.5	-0.0	-4.5
std	0.6	4.9	0.1	4.8	0.1	4.8
min	-1.3	-17.6	-0.3	-16.0	-0.3	-16.0
max	0.6	0.6	0.0	0.0	0.0	0.0
4 customers						
count	30.0	30.0	30.0	30.0	30.0	30.0
mean	0.7	-10.5	0.1	-13.8	0.1	-13.8
std	1.6	10.1	0.8	12.8	0.8	12.8
min	-0.8	-34.2	-0.4	-37.0	-0.4	-37.0
max	9.0	0.6	3.6	0.0	3.6	0.0
6 customers						
count	30.0	30.0	30.0	30.0	30.0	30.0
mean	1.5	-12.9	3.3	-18.3	1.0	-20.0
std	2.3	7.5	4.3	10.6	2.0	10.7
min	-0.5	-24.5	-0.2	-32.9	-0.4	-32.9
max	9.5	4.7	15.8	5.0	9.6	2.5
8 customers						
count	30.0	30.0	19.0	19.0	19.0	19.0
mean	6.2	-5.2	13.5	-9.2	0.6	-19.7
std	4.8	9.9	7.4	12.9	5.0	9.2
min	0.5	-19.0	2.9	-25.4	-12.6	-33.3
max	17.9	15.8	28.8	13.3	5.5	-0.8

Table 5: Statistic per instance size of the gaps between the available bounds and the lowest cost solutions found by the VNS for the VRPTT and VRPTTMU.

4.4.1 VRPTT

In Table 5 we see that for one of the instances with two customers the VNS found a solution for the VRPTT that has a gap of -0.3 % with the lower bound of the VRPTT, which should not be possible. For instances with only two customers, where the probability of the VNS finding the optimal solution is the greatest, the mean and the standard deviation of the gap with the optimal value of the VRPTT is 0.0 % and 0.1 % respectively and with the optimal value of the TTRP 0.0 % and 0.6 % respectively. The reason for this error is unknown, but at least there is evidence that the mean of the error is zero.

For none of the instances did the best solution found by the VNS use trailer sharing. This is the reason that in Figure 11 none of the values found by the VNS are significantly lower than the optimal values of the TTRP. All negative gaps between the values found by the VNS for the VRPTT and the optimal value of the TTRP must thus be attributed to the aforementioned error and not to the benefit of being able to share trailers.

Since the results for the VRPTT described in [9] exclude time related costs and the results for the TTRP described in [19] include time related costs, no conclusions can be drawn about whether sharing trailers leads to solutions with lower costs on these instances based on these results alone. Only an indirect comparison of these values can be made by using the results found by the VNS. If sharing trailers leads to lower cost solutions on these instances, there may be many instances for which the gap with the upper bound of the VRPTT is signif-icanlty greater than the gap with the optimal value of the TTRP. In Figure 12 the gap with respect to the optimal value of the TTRP is plotted against the gaps with respect to the bounds of the VRPTT for each instance. The figure does not support the conclusion that trailer sharing leads to significantly lower costs on these instances. In Table 5 we see that the mean gaps for instances with two, four, six and eight customers of the VRPTT with respect to the lower bound of the VRPTT are 0.0, 0.1, 3.3 and 13.5 % respectively and with respect to the upper bound 0.0, 0.1, 1.0, 0.6 % respectively. Therefore, the benefit of sharing trailers can not be ruled out, but can not be shown to be definitely present either.

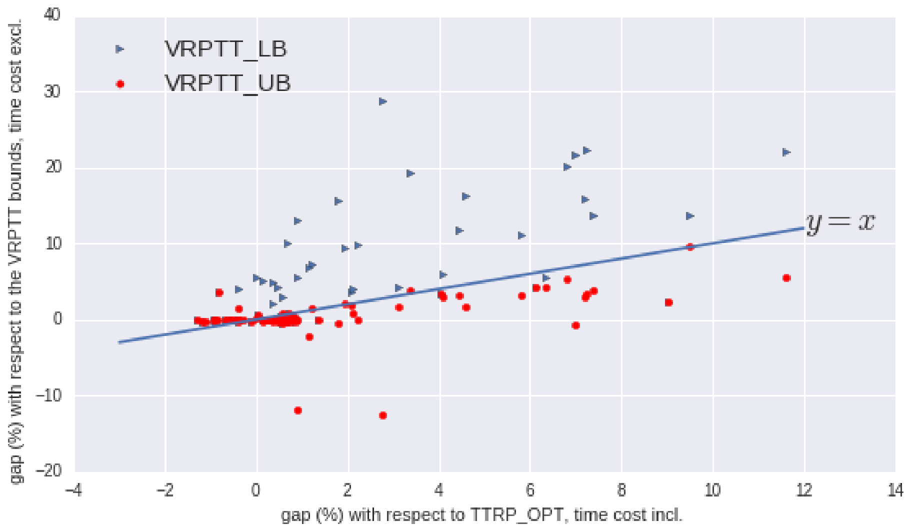

Figure 12: For each instance the gap with respect to the TTRP optimum against the gap with respect to the bounds of the VRPTT.

The VNS was able to find solutions for all instances including the eleven in-stances for which no VRPTT bounds were available.

4.4.2 VRPTTMU

The difference between the VRPTT and the VRPTTMU is that the latter allows lorries to do multiple trips by allowing it to visit the depot multiple times to couple a trailer, decouple a trailer or unload whereas the former does not. In Figure 11 we can see that for many instances the VNS finds solutions with significantly lower costs than than the TTRP and the VRPTT. This is not surprising since in some instances less vehicles can be used, which results in less fixed costs. The instances have such nonrestrictive time windows that one lorry can visit many customers interleaved with depot visits.

In Table 5 we see that for instances with two, four, six and eight customers the mean gaps with respect to the optima of the TTRP are -3.5, -10.5, -12.9 and -5.2 respectively and with respect to the lower bounds of the VRPTT -4.5, -13.8, -18.3 and -9.2 % respectively. This indicates that significant cost reductions can be achieved by allowing lorries to do multiple trips. For each instance the VNS found a solution to the VRPTTMU.

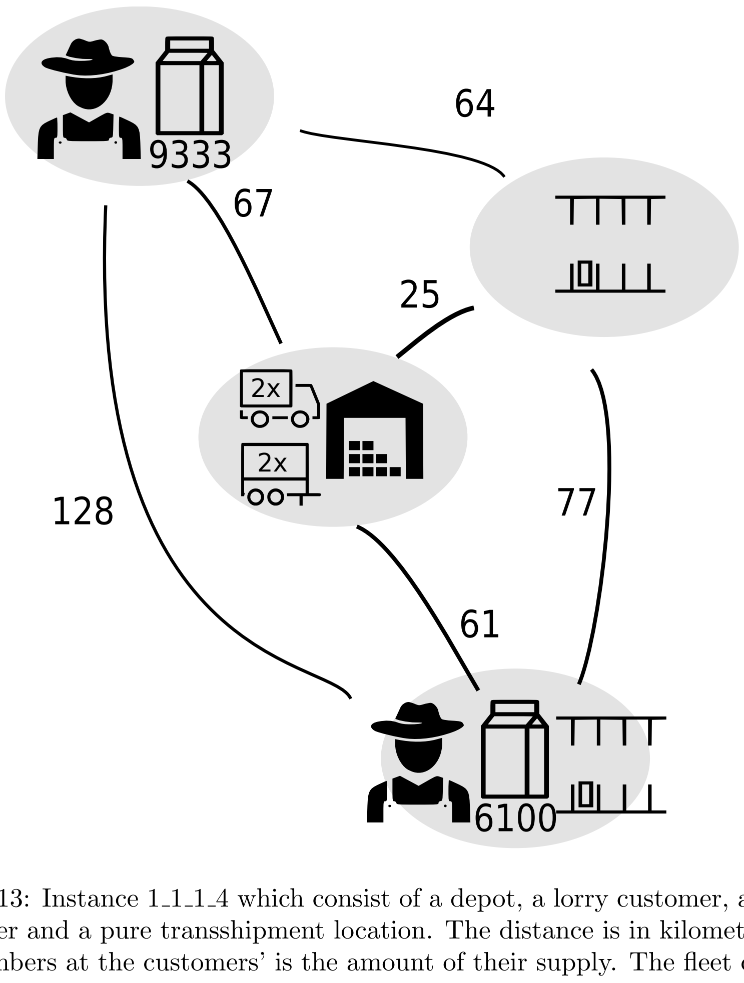

Figure 13: Instance 1_1_1_4 which consist of a depot, a lorry customer, a trailer customer and a pure transshipment location. The distance is in kilometers and the numbers at the customers' is the amount of their supply. The fleet consists of one of each of the vehicles described in Table 3.

An example of the benefits of allowing lorries to do multiple trips is given in Figure 13. It illustrates one of the instances with two customers. The fleet consist of one of each of the vehicles described in Table 3. There is a lorry customer with a supply of 9333 units, a trailer customer with a supply of 6100 units and a pure transshipment location. The time window of each location is the interval [0, 1320] minutes hence there is enough time for one lorry to visit both customers and collect their supplies. The sum of the customer supplies is equal to 15433 which is greater than the capacity of either lorry. If the lorry is allowed to do multiple trips, it can unload at the depot between visiting the customers. If this is not allowed, the cheapest solution is to use a trailer. The fixed cost of this trailer will need to be paid together with its distance related costs. In Appendix B.2 we can see that allowing lorries to do multiple trips leads to a gap of -13.8 % with respect to the optimal value of the TTRP including time costs and -12.5 % with respect to the optimal value of the VRPTT excluding time costs for this instance.

5 Conclusion

A few contributions have been made in this thesis. Trailers have their own maximum speed such that it is possible that a lorry's maximum speed is higher without a trailer than with a trailer. The manner in which the model represents load transfers has been simplified by removing the need for vertices only dedicated to transfering load. The VRPTT has been remodeled such that it can be extended to the VRPTTMU by only removing one set of constraints and by adding a certain type of trailer to the instances that does not have influence on the best solutions of the VRPTT. The VRPTTMU is able to represent the situation where lorries visit the depot multiple times without ending their path to either couple or decouple a trailer or to unload their load. This allows lorries to take multiple trips which resulted in significantly decreased costs on many test instances. A variable neighborhood search heuristic has been presented and implemented which found solutions for the VRPTT and the VRPTTMU for all instances with up to eight customers in minutes. The objective function of the VRPTT has been extended such that besides the fixed costs and the distance related costs, the objective function is also a function of the time related costs. This allowed a comparison with results to the TTRP for the first time, which showed that it is not at all clear that the instances used in this thesis are suitable to show the benefits of trailer sharing. Perhaps these benefits become more apparent as the size of the problem instance grows.

The aim of this thesis was to develop a heuristic that can handle larger instances than exact methods and that can be extended such that it can be used to develop a solution for TransMission's problem. This goal has been achieved with the VNS heuristic. The heuristic found in a short period of time reasonably good solutions for all tested instances, even for the instances for which the exact method could not find a solution. An extension of the model that reduces costs has already been demonstrated by allowing lorries to unload multiple times. Furthermore, since the heuristic is fairly simple and unoptimized, there is reason to believe that it can be optimized for better performance.

Before the results of this thesis can be used by TransMission to produce transport plans some work still has to be done. The model has to be extended such that it supports multiple commodities, multiple depots and split deliveries.

6 Future Work

There are improvements that can be made to the model:

- The amount of symmetry in the model could be reduced by adding certain constraint to the model. In the current representation of a solution, multiple representations can map to the same solution.
- The need to create η transshipment vertex pairs for each transshipment location instead of generating them when needed by the optimization method can be removed.

There are ways that the model can be extended:

- The model can be extended from a vehicle routing problem to a pickup and delivery problem.
- The model can be extended such that it can represent a set of customers that can change during the time horizon.
- The model can be extended such that it supports multiple depots.
- The model can be extended such that it can model split deliveries.
- The model can be extended such that the maximum amount of a customer's supply is unrelated to the capacity of the vehicles.

There are improvements that can be made to the optimization method:

- More constraints could be turned into soft constraints like the precedence of decoupling a trailer over coupling a trailer at a transshipment vertex.
- A solution can be decomposed into several sub-solutions which are independent of each other with respect to each others load and time table. The current method recalculates the sub-solutions for each new solution even if the sub-solutions have already been seen before in other solutions. This can be resolved by caching the sub-solutions of solutions.
- New operators can be developed to speed up the search heuristic.
- The parameters used by the optimization method can be tuned by performing a hyperparameter optimization.
- The time table algorithm can be improved for the VRPTTMU which potentially leads to a reduction of trailers' time related cost.
- A probabilistic datastructure like a bloom filter can be used for the tabu table to reduce its memory consumption.
- The method could be adapted such that it can be executed in parallel.

There are improvements that can be made to the testing method:

- The influence that neighborhood operators have on the results could be tested.
- The influence that turning a constraint into a soft constraint has on the results could be tested.
- The method can be tested for more iterations and on larger instances.
- Different types of instances could be used to test the relation between instance characteristics and the benefits of trailer sharing.

A Symbols and Abbreviations

Many symbols and abbreviations are used throughout this thesis. They are enumerated with their meaning in Table 6.

Symbol	Meaning
VRP	The vehicle routing problem.
RVRP	The rich vehicle routing problem.
VRPMS	The vehicle routin problem with multiple synchronization constraints.
VRPTT	The vehicle routing problem with trailers and transshipments.
VRPTTMU	The vehicle routing problem with trailers, transshipments and multiple unloads.
🚚	The lorry icon.
🛒	The trailer icon.
⊸	The transshipment couple vertex icon.
⟜	The transshipment decouple vertex icon.
🥛	The supply collection vertex icon.
🕒	The time window icon.
▶	The start of vehicle path vertex icon.
⏹	The end of vehicle path vertex icon.
⊥	The transshipment location icon.
🧑‍🌾	The customer icon.
🏭	The depot icon.
⊕	The supersource vertex icon.
⊖	The supersink vertex icon.
$\mathcal{L}$	The set of locations.
$\mathcal{D}$	The mapping from a pair of locations to the distance between them.
$\mathcal{F}$	The set of vehicles in the fleet.
$\mathcal{L}^{\text{types}}$	The set of location types.
$locationType_i$	The location type of location i.
α_i	The start of the time window of location i.
β_i	The end of the time window of location i.
$\mathcal{L}^{\text{customer}}$	The set of customer locations.
$locationSupply_i$	The supply of customer location i.
$\mathcal{F}^{\text{types}}$	The set of vehicle types.
$\mathcal{F}^{\text{lorry}}$	The set of vehicle lorries.
$\mathcal{F}^{\text{trailer}}$	The set of trailers.
$\mathcal{F}^{\text{axle-types}}$	The set of axle types of vehicles.
f_k^{type}	The vehicle type of vehicle k.
f_k^{capacity}	The capacity of vehicle k.
f_k^{fixed}	The fixed cost of vehicle k.
f_k^{distance}	The variable distance cost of vehicle k per kilometer.
f_k^{time}	The variable time cost of vehicle k per hour.
$f_k^{\text{axle-type}}$	The axle type of vehicle k.
f_k^{speed}	The maximum speed of vehicle k on distances longer than ϕ.

Symbol	Meaning
$f_k^{\text{shortspeed}}$	The maximum speed of vehicle k on distances shorter than or equal to ϕ.
$f_k^{\text{loadspeed}}$	The load speed of lorry k in minutes per unit.
ϕ	The cut-off distance that determines the maximum speed of the vehicles.
τ^{D}	The duration of coupling or decoupling a trailer at the depot in minutes.
τ^{R}	The duration of coupling or decoupling a trailer at a transshipment location in minutes.
$\mathcal{R}^-$	The set of transshipment decouple vertices.
$\mathcal{R}^+$	The set of transshipment couple vertices.
$\mathcal{L}^{\text{transshipment}}$	The set of transshipment locations.
$r_m^{j,-}$	The decouple vertex of the mth transshipment vertex pair at transshipment location j.
$r_m^{j,+}$	The couple vertex of transshipment vertex pair m at transshipment location j.
η	The amount of transshipment vertex pairs per transshipment location.
$\mathcal{S}$	The set of the customers' supply collection vertices
s_i	the supply collection vertex of customer location i.
$\mathcal{S}^{\text{trailer}}$	The set of supply collection vertices of the the trailer customers.
$\mathcal{S}^{\text{lorry}}$	The set of supply collection vertices of the lorry customers.
$\mathcal{M}^-$	The set of end vertices of the lorries.
$\mathcal{M}^+$	The set of start vertices of the lorries.
m_k^-	The end vertex of lorry k.
m_k^+	The start vertex of lorry k.
$\mathcal{N}^-$	The set of end vertices of the trailers.
$\mathcal{N}^+$	The set of start vertices of the trailers.
n_l^-	The end vertex of trailer l.
n_l^+	The start vertex of trailer l.
$\mathcal{V}$	The union of the set of vehicle start and end vertices, the set of transshipment decouple and couple vertices and the set of customer supply collection vertices.
v^{loc}	The location of vertex v.
τ_v^{start}	The start of vertex v's time window.
τ_v^{end}	The end of vertex v's time window.
τ^{min}	The start of the time horizon.
τ^{max}	The end of the time horizon.
$\delta_{u,v}$	The distance between vertex u and v.
$\tau_{u,v,k}^{\text{travel}}$	The travel time of vehicle k between vertices u and v.
$\tau_{u,v,k,l}^{\text{extra}}$	The extra time it takes for lorry l to traverse edge (u,v) if it is coupled with trailer l.
$travelTime$	The function that takes as input an edge and a path assignment and returns the amount of time it took for vehicles to traverse that edge.
$visitationStart$	The function that takes as input a vertex, a path assignment and a time table and returns the time at which the the vertex' visitation started.
$\mathcal{E}$	The set of edges $\{(u,v) : u, v \in \mathcal{V}, u \neq v\}$.
$\mathcal{G}$	The graph $(\mathcal{V}, \mathcal{E})$.
$x_{u,v}^k$	The binary decision variable that models whether lorry k traverses edge (u,v).
$y_{u,v}^{k,l}$	T'he binary decision variable that models whether lorry k traverses edge (u,v) whilst coupled to trailer l.
$t_{u,v}^k$	The real-valued decision variable that models at which time lorry k finishes traversing edge (u,v).
$z_{u,v}^k$	The real-valued decision variable that models the amount of load with which vehicle k traverses edge (u,v).
C	The objective function.

Symbol	Meaning
C^{distance}	The function that calculates the distance-related costs of a solution.
C^{time}	The function that calculates the time-related costs of a solution.
C^{fixed}	The function that calculates the fixed costs of a solution.
X	The lorries' paths of a solution.
Y	The trailers' paths of a solution.
T	The time table of a solution.
L	The load table of a solution.
XY	The path assignment (X, Y).
VNS	The variable neighborhood search .
$\omega^{\text{unserved-customer}}$	The penalty for not serving a customer.
$U^{\text{unserved-customer}}$	The function that calculates the penalty of a solution due to unserved customers.
$\omega^{\text{lorry-customer}}$	The penalty for visiting a lorry customer with a trailer.
$U^{\text{lorry-customer}}$	The function that calculates the penalty of a solution due to lorry customers visited with trailers.
$\omega^{\text{time-window}}$	The penalty per minute for a lorry violating the end of a vertex' time window.
$U^{\text{time-window}}$	The function that calculates the penalty of a solution due to violated time windows.
$\omega^{\text{capacity-shortage}}$	The penalty per supply unit left uncollected of a visited customer.
$U^{\text{capacity-shortage}}$	The function that calculates the penalty of a solution due to uncollected supply of visited customers.
U	The functional that calculates the penalty of a solution.
$nSamples$	The amount of path assignments that the operation *mergelorries* returns.
$\mathcal{G}^{\text{c}}$	The capacity graph $(\mathcal{V}^{\text{c}}, \mathcal{E}^{\text{c}})$.
$\mathcal{V}^{\text{c}}$	The set of vertices of the capacity graph $\mathcal{G}^{\text{c}}$.
$\mathcal{E}^{\text{c}}$	The set of edges of the capacity graph $\mathcal{G}^{\text{c}}$.
$capacity$	A mapping from edges in the capacity graph to their capacities.
$\mathcal{P}$	The set of vertices used to model load transfers to trailers when constructing the load table.
p_l	The vertex used to model load transfers to trailer l when constructing the load table.
$source$	The vertex used as a supersource when constructing the time and load table.
$sink$	The vertex used as a supersink when constructing the time and load table.
f^{c}	A mapping that represent the solution to the maximum flow problem on the capacity graph from the supersource to the supersink.
$\lvert f^{\text{c}} \rvert$	The total amount of customer supply collected.
$\mathcal{G}^{\text{trailer}}$	The graph ($\mathcal{V}$ used to determine trailers' trailer paths.
$trailerPath_l$	The trailer path of trailer l which is used to determine the load of a trailer during each edge it traverses.
$\mathcal{E}^{\text{pair}}$	The set of edges that connects each transshipment decouple vertex with the couple vertex it is paired with.
$\mathcal{E}^{\text{used-trailer}}$	The set of edges traversed by any trailer.
$\mathcal{V}^{\text{precedence}}$	The set of vertices used to model the precedence graph.
$\mathcal{E}^{\text{earliest}}$	The set of edges used to calculate the earliest arrival times.
$\mathcal{E}^{\text{latest}}$	The set of edges used to calculate the latest time at which trailers start can start their paths.
$duration^{\text{earliest}}$	A function that maps from edges to their weights.
$duration^{\text{latest}}$	A function that maps from edges to their weights.
$duration^{\text{result}}$	A function that maps from edges to their weights.
$\mathcal{G}^{\text{earliest}}$	$(\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{earliest}}, duration^{\text{earliest}})$
$\mathcal{G}^{\text{latest}}$	$(\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{latest}}, duration^{\text{latest}})$
$\mathcal{G}^{\text{result}}$	$(\mathcal{V}^{\text{precedence}}, \mathcal{E}^{\text{earliest}}, duration^{\text{result}})$
$\mathcal{V}^{\text{start}}$	The set of edges used to model the start of vertex visits in a precedence graph.
$\mathcal{V}^{\text{end}}$	The set of edges used to model the end of vertex visits in a precedence graph.

Symbol	Meaning
$\mathcal{E}^{\text{latest-start}}$	The set of edges used to constrain the time at which lorries start their paths.
$score$	The function that calculates the score of a solution, which is the sum of its costs and its penalties.
$TABU$	The collection that is used to cache path assignments.
PQ	The collection that keeps solutions with unexplored neighborhoods in sorted order.
p	The probability that a random item is removed from the PQ instead the one with the highest priority.
$VRPTT\_UB$	Upper bounds for the VRPTT excluding time costs.
$VRPTT\_LB$	Lower bounds for the VRPTT excluding time costs.
$TTRP\_OPT$	The optimal values for the TTRP including time costs.

Table 6: Symbols and abbreviations with their meanings.

B Results and Bounds

B.1 Nominal Values

Table 7 describes for each instance the lowest cost solution found by the VNS for the VRPTT and the VRPTTMU and the optimal value for the TTRP provided in [19] which include time costs and the lower and upper bound of the VRPTT provided in [9] which exclude time costs.

	including time costs			excluding time costs			
instance	VRPTT	VRPTTMU	TTRP_OPT	VRPTT	VRPTTMU	VRPTT_LB	VRPTT_UB
1_1_1_00	32660.0	32660.0	32480.0	25280.0	25280.0	25280.0	25280.0
1_1_1_01	43440.0	42080.0	43260.0	32040.0	29960.0	32040.0	32040.0
1_1_1_02	40200.0	40200.0	40020.0	30360.0	30360.0	30360.0	30360.0
1_1_1_03	39030.0	39030.0	38850.0	29730.0	29730.0	29730.0	29730.0
1_1_1_04	58467.0	50840.0	58993.0	39567.0	34640.0	39575.0	39575.0
1_1_1_05	36960.0	35030.0	36900.0	28680.0	26450.0	28680.0	28680.0
1_1_1_06	47695.0	47695.0	47515.0	33535.0	33535.0	33535.0	33535.0
1_1_1_07	41450.0	41450.0	41270.0	31130.0	31130.0	31130.0	31130.0
1_1_1_08	37140.0	37140.0	36960.0	28680.0	28680.0	28680.0	28680.0
1_1_1_09	48065.0	48065.0	47885.0	33665.0	33665.0	33665.0	33665.0
1_1_1_10	39540.0	39540.0	39900.0	29040.0	29040.0	29117.0	29117.0
1_1_1_11	37428.0	37428.0	37852.0	28128.0	28128.0	28205.0	28205.0
1_1_1_12	48860.0	45840.0	48680.0	35120.0	32040.0	35120.0	35120.0
1_1_1_13	38422.0	38422.0	38878.0	28582.0	28582.0	28659.0	28659.0
1_1_1_14	46631.0	46560.0	47069.0	32882.0	32040.0	32890.0	32890.0
1_1_1_15	45450.0	45450.0	45270.0	32430.0	32430.0	32430.0	32430.0
1_1_1_16	30420.0	30420.0	30240.0	24240.0	24240.0	24240.0	24240.0
1_1_1_17	43724.0	40070.0	44016.0	31364.0	28790.0	31372.0	31372.0
1_1_1_18	67855.0	56400.0	68445.0	44635.0	37500.0	44643.0	44643.0
1_1_1_19	36220.0	36220.0	36100.0	28120.0	28120.0	28120.0	28120.0
1_1_1_20	48310.0	42130.0	48610.0	33850.0	29830.0	33858.0	33858.0
1_1_1_21	40020.0	37280.0	39840.0	30360.0	27620.0	30360.0	30360.0

	including time costs			excluding time costs			
instance	VRPTT	VRPTTMU	TTRP_OPT	VRPTT	VRPTTMU	VRPTT_LB	VRPTT_UB
1_1_1_22	42610.0	42610.0	42430.0	31690.0	31690.0	31690.0	31690.0
1_1_1_23	50350.0	47590.0	50170.0	35890.0	32950.0	35890.0	35890.0
1_1_1_24	44180.0	41780.0	44000.0	32493.0	29960.0	32501.0	32501.0
1_1_1_25	52700.0	49570.0	53380.0	36440.0	33730.0	36448.0	36448.0
1_1_1_26	44510.0	42660.0	44330.0	32810.0	30480.0	32810.0	32810.0
1_1_1_27	42100.0	42100.0	41980.0	31480.0	30870.0	31480.0	31480.0
1_1_1_28	33800.0	31320.0	33620.0	26720.0	24240.0	26720.0	26720.0
1_1_1_29	45740.0	43590.0	45560.0	33440.0	30870.0	33440.0	33440.0
2_2_2_00	56280.0	54220.0	55980.0	39180.0	36460.0	39180.0	39180.0
2_2_2_01	74787.0	56290.0	75193.0	56007.0	38007.0	56084.0	56084.0
2_2_2_02	65742.0	56720.0	65180.0	43122.0	38900.0	43217.0	43217.0
2_2_2_03	71718.0	53798.0	71298.0	55338.0	37418.0	55420.0	55420.0
2_2_2_04	56709.0	54171.0	56409.0	38169.0	36871.0	38252.0	38252.0
2_2_2_05	42260.0	41890.0	41960.0	31340.0	29830.0	31340.0	31340.0
2_2_2_06	71444.0	66230.0	71084.0	45966.0	42830.0	45974.0	45974.0
2_2_2_07	85527.0	56775.0	86251.0	61647.0	37474.0	59499.0	59499.0
2_2_2_08	72580.0	72580.0	72340.0	48280.0	47250.0	48280.0	48280.0
2_2_2_09	80945.0	62945.0	81275.0	59105.0	41105.0	59182.0	59182.0
2_2_2_10	75040.0	55040.0	74740.0	57640.0	37640.0	57640.0	57640.0
2_2_2_11	82157.0	64157.0	82493.0	59897.0	41897.0	59974.0	59974.0
2_2_2_12	67055.0	63400.0	67125.0	43655.0	41660.0	43728.0	43728.0
2_2_2_13	56122.0	54510.0	55702.0	38302.0	37350.0	38314.0	38314.0
2_2_2_14	58264.0	50150.0	53438.0	38884.0	35330.0	38026.0	38026.0
2_2_2_15	55428.0	52220.0	55068.0	38792.0	35550.0	38877.0	38877.0
2_2_2_16	105591.0	84520.0	104160.0	72891.0	53880.0	72968.0	72968.0
2_2_2_17	63456.0	60820.0	62976.0	41796.0	41140.0	41891.0	41891.0
2_2_2_18	49750.0	48145.0	49450.0	35470.0	33145.0	35470.0	35470.0
2_2_2_19	66033.0	62140.0	65613.0	43653.0	41270.0	43665.0	43665.0
2_2_2_20	61099.0	60050.0	60739.0	40639.0	39710.0	40799.0	40799.0
2_2_2_21	44870.0	44145.0	44570.0	32810.0	31065.0	32810.0	32810.0
2_2_2_22	53920.0	53920.0	53620.0	37780.0	37780.0	37780.0	37780.0
2_2_2_23	84051.0	66051.0	83691.0	61491.0	43491.0	61574.0	61574.0
2_2_2_24	80947.0	62947.0	80527.0	59707.0	41454.0	59790.0	59790.0
2_2_2_25	69843.0	65850.0	69363.0	45363.0	44010.0	45458.0	45458.0
2_2_2_26	49050.0	46650.0	48810.0	35190.0	32430.0	35190.0	35190.0
2_2_2_27	59522.0	55810.0	59102.0	40368.0	37045.0	40380.0	40380.0
2_2_2_28	69097.0	66330.0	68737.0	45397.0	43675.0	45409.0	45409.0
2_2_2_29	85435.0	67435.0	85135.0	61615.0	43615.0	61698.0	61698.0
3_3_3_00	78232.0	70890.0	75172.0	49672.0	44910.0	48033.0	48033.0
3_3_3_01	73197.0	64970.0	72597.0	47037.0	42180.0	47061.0	47061.0
3_3_3_02	107945.0	96850.0	106984.0	75202.0	57910.0	71290.7	75209.0
3_3_3_03	58361.0	55840.0	58659.0	39461.0	37240.0	39469.0	39469.0
3_3_3_04	64572.0	60940.0	64032.0	43872.0	41140.0	43880.0	43880.0
3_3_3_05	110244.0	93283.0	108285.0	75144.0	56623.0	64918.2	75464.0
3_3_3_06	86800.0	77154.0	86260.0	63940.0	48714.0	64018.0	64018.0
3_3_3_07	81330.0	70134.0	80850.0	60930.0	44154.0	60930.0	60930.0
3_3_3_08	98000.0	79529.0	98388.0	69610.0	48929.0	66850.6	68676.0

	including time costs			excluding time costs			
instance	VRPTT	VRPTTMU	TTRP_OPT	VRPTT	VRPTTMU	VRPTT_LB	VRPTT_UB
3_3_3_09	95352.0	68149.0	87091.0	67092.0	42440.0	59008.7	61207.0
3_3_3_10	93029.0	73160.0	92740.0	66914.0	47720.0	66940.0	66940.0
3_3_3_11	81927.0	62910.0	81336.0	60807.0	41490.0	60344.0	60344.0
3_3_3_12	65993.0	60910.0	65857.0	42953.0	40815.0	42961.0	42961.0
3_3_3_13	68244.0	65535.0	68506.0	44563.0	42375.0	44648.0	44648.0
3_3_3_14	102645.0	89160.0	100560.0	70845.0	53880.0	68338.0	69527.0
3_3_3_15	105995.0	90583.0	105626.0	72670.0	55423.0	69275.3	72680.0
3_3_3_16	80811.0	61561.0	80391.0	59391.0	40621.0	59133.1	59403.0
3_3_3_17	96331.0	83732.0	95971.0	69211.0	51150.0	67813.0	69353.0
3_3_3_18	73548.0	70662.0	73008.0	47628.0	45942.0	47640.0	47640.0
3_3_3_19	90797.0	73150.0	90790.0	64637.0	46244.0	61289.6	64578.0
3_3_3_20	99197.0	91090.0	93265.0	69617.0	64690.0	65981.0	66840.0
3_3_3_21	117307.0	102810.0	115094.0	80467.0	61290.0	73564.2	78791.0
3_3_3_22	113547.0	99160.0	107303.0	76824.0	59920.0	69179.4	74467.0
3_3_3_23	80213.0	60320.0	79900.0	60473.0	40580.0	60473.0	60473.0
3_3_3_24	101044.0	78849.0	100914.0	70264.0	49629.0	66863.2	70385.0
3_3_3_25	91031.0	74906.0	91011.0	64971.0	47006.0	64616.0	64616.0
3_3_3_26	100723.0	102304.0	97668.0	70224.0	70783.0	67394.5	69038.0
3_3_3_27	118849.0	98370.0	116382.0	78877.0	61230.0	75777.0	78318.0
3_3_3_28	92806.0	74806.0	93064.0	65506.0	47506.0	65514.0	65514.0
3_3_3_29	97835.0	79835.0	97020.0	68435.0	50435.0	68232.0	68232.0
4_4_4_00	146800.0	136688.0	135871.0	103360.0	98468.0	NaN	NaN
4_4_4_01	115281.0	95999.0	110753.0	78981.0	58499.0	74558.8	76707.0
4_4_4_02	141037.0	124908.0	136253.0	101077.0	84408.0	NaN	NaN
4_4_4_03	117899.0	95481.0	114071.0	79919.0	59181.0	66956.7	76895.0
4_4_4_04	127460.0	132394.0	120540.0	92780.0	96132.0	NaN	NaN
4_4_4_05	97818.0	78753.0	96648.0	69258.0	49533.0	64597.6	68227.0
4_4_4_06	110840.0	103170.0	103612.0	75800.0	70600.0	62338.3	76362.0
4_4_4_07	129985.0	136806.0	129205.0	95125.0	99006.0	NaN	NaN
4_4_4_08	102880.0	82880.0	102280.0	72760.0	52760.0	70684.3	72219.0
4_4_4_09	121588.0	101566.0	113233.0	80608.0	61126.0	70914.6	77696.0
4_4_4_10	99331.0	84804.0	98671.0	70171.0	52524.0	63722.6	70055.0
4_4_4_11	102593.0	83450.0	100360.0	71033.0	51770.0	64698.2	71088.0
4_4_4_12	114548.0	95125.0	106835.0	76868.0	56545.0	66390.1	74653.0
4_4_4_13	109563.0	103714.0	104913.0	74343.0	71540.0	66476.6	72117.0
4_4_4_14	119274.0	105900.0	111229.0	79734.0	72120.0	65135.0	77077.0
4_4_4_15	120662.0	95160.0	113695.0	80293.0	59340.0	76965.0	76965.0
4_4_4_16	175018.0	168612.0	150907.0	127261.0	123792.0	NaN	NaN
4_4_4_17	142948.0	124966.0	127415.0	102088.0	85126.0	NaN	NaN
4_4_4_18	124732.0	108460.0	111766.0	82252.0	74860.0	67322.2	77964.0
4_4_4_19	144305.0	150951.0	130335.0	103385.0	83811.0	NaN	NaN
4_4_4_20	153522.0	151160.0	142142.0	108822.0	100628.0	NaN	NaN
4_4_4_21	125003.0	110935.0	119501.0	82523.0	66355.0	70915.0	81154.0
4_4_4_22	175856.0	141955.0	149116.0	128276.0	91735.0	NaN	NaN
4_4_4_23	147357.0	143300.0	128958.0	103917.0	101480.0	NaN	NaN
4_4_4_24	99269.0	98907.0	98129.0	68189.0	58887.0	63792.2	69677.0
4_4_4_25	104514.0	87998.0	103582.0	71694.0	54278.0	63440.8	81368.0

	including time costs			excluding time costs			
instance	VRPTT	VRPTTMU	TTRP_OPT	VRPTT	VRPTTMU	VRPTT_LB	VRPTT_UB
4_4_4_26	101805.0	101047.0	101347.0	70005.0	61207.0	67108.1	70250.0
4_4_4_27	136832.0	111991.2	124090.0	97772.0	76580.0	NaN	NaN
4_4_4_28	109664.0	85300.0	102658.0	74787.0	52180.0	62286.9	70983.0
4_4_4_29	117740.0	104626.0	114586.0	79695.0	66404.0	61872.4	91197.0

B.2 Relative Values

Table 8 describes for each instance the gap in percentages between the lowest cost solution found by the VNS for the VRPTT and the VRPTTMU and the optimal value for the TTRP provided in [19] which include time costs and the lower and upper bound of the VRPTT provided in [9] which exclude time costs.

	TTRP_OPT, time cost incl.		VRPTT_LB, time cost excl.		VRPTT_UB, time cost excl.	
instance	VRPTT	VRPTTMU	VRPTT	VRPTTMU	VRPTT	VRPTTMU
1_1_1_00	0.6	0.6	0.0	0.0	0.0	0.0
1_1_1_01	0.4	-2.7	0.0	-6.5	0.0	-6.5
1_1_1_02	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_03	0.5	0.5	0.0	0.0	0.0	0.0
1_1_1_04	-0.9	-13.8	-0.0	-12.5	-0.0	-12.5
1_1_1_05	0.2	-5.1	0.0	-7.8	0.0	-7.8
1_1_1_06	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_07	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_08	0.5	0.5	0.0	0.0	0.0	0.0
1_1_1_09	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_10	-0.9	-0.9	-0.3	-0.3	-0.3	-0.3
1_1_1_11	-1.1	-1.1	-0.3	-0.3	-0.3	-0.3
1_1_1_12	0.4	-5.8	0.0	-8.8	0.0	-8.8
1_1_1_13	-1.2	-1.2	-0.3	-0.3	-0.3	-0.3
1_1_1_14	-0.9	-1.1	-0.0	-2.6	-0.0	-2.6
1_1_1_15	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_16	0.6	0.6	0.0	0.0	0.0	0.0
1_1_1_17	-0.7	-9.0	-0.0	-8.2	-0.0	-8.2
1_1_1_18	-0.9	-17.6	-0.0	-16.0	-0.0	-16.0
1_1_1_19	0.3	0.3	0.0	0.0	0.0	0.0
1_1_1_20	-0.6	-13.3	-0.0	-11.9	-0.0	-11.9
1_1_1_21	0.5	-6.4	0.0	-9.0	0.0	-9.0
1_1_1_22	0.4	0.4	0.0	0.0	0.0	0.0
1_1_1_23	0.4	-5.1	0.0	-8.2	0.0	-8.2
1_1_1_24	0.4	-5.0	-0.0	-7.8	-0.0	-7.8
1_1_1_25	-1.3	-7.1	-0.0	-7.5	-0.0	-7.5
1_1_1_26	0.4	-3.8	0.0	-7.1	0.0	-7.1
1_1_1_27	0.3	0.3	0.0	-1.9	0.0	-1.9
1_1_1_28	0.5	-6.8	0.0	-9.3	0.0	-9.3
1_1_1_29	0.4	-4.3	0.0	-7.7	0.0	-7.7
2_2_2_00	0.5	-3.1	0.0	-6.9	0.0	-6.9
2_2_2_01	-0.5	-25.1	-0.1	-32.2	-0.1	-32.2
2_2_2_02	0.9	-13.0	-0.2	-10.0	-0.2	-10.0
2_2_2_03	0.6	-24.5	-0.1	-32.5	-0.1	-32.5
2_2_2_04	0.5	-4.0	-0.2	-3.6	-0.2	-3.6
2_2_2_05	0.7	-0.2	0.0	-4.8	0.0	-4.8
2_2_2_06	0.5	-6.8	-0.0	-6.8	-0.0	-6.8
2_2_2_07	-0.8	-34.2	3.6	-37.0	3.6	-37.0
2_2_2_08	0.3	0.3	0.0	-2.1	0.0	-2.1
2_2_2_09	-0.4	-22.6	-0.1	-30.5	-0.1	-30.5

	TTRP_OPT, time cost incl.		VRPTT_LB, time cost excl.		VRPTT_UB, time cost excl.	
instance	VRPTT	VRPTTMU	VRPTT	VRPTTMU	VRPTT	VRPTTMU
2_2_2_10	0.4	-26.4	0.0	-34.7	0.0	-34.7
2_2_2_11	-0.4	-22.2	-0.1	-30.1	-0.1	-30.1
2_2_2_12	-0.1	-5.5	-0.2	-4.7	-0.2	-4.7
2_2_2_13	0.8	-2.1	-0.0	-2.5	-0.0	-2.5
2_2_2_14	9.0	-6.2	2.3	-7.1	2.3	-7.1
2_2_2_15	0.7	-5.2	-0.2	-8.6	-0.2	-8.6
2_2_2_16	1.4	-18.9	-0.1	-26.2	-0.1	-26.2
2_2_2_17	0.8	-3.4	-0.2	-1.8	-0.2	-1.8
2_2_2_18	0.6	-2.6	0.0	-6.6	0.0	-6.6
2_2_2_19	0.6	-5.3	-0.0	-5.5	-0.0	-5.5
2_2_2_20	0.6	-1.1	-0.4	-2.7	-0.4	-2.7
2_2_2_21	0.7	-1.0	0.0	-5.3	0.0	-5.3
2_2_2_22	0.6	0.6	0.0	0.0	0.0	0.0
2_2_2_23	0.4	-21.1	-0.1	-29.4	-0.1	-29.4
2_2_2_24	0.5	-21.8	-0.1	-30.7	-0.1	-30.7
2_2_2_25	0.7	-5.1	-0.2	-3.2	-0.2	-3.2
2_2_2_26	0.5	-4.4	0.0	-7.8	0.0	-7.8
2_2_2_27	0.7	-5.6	-0.0	-8.3	-0.0	-8.3
2_2_2_28	0.5	-3.5	-0.0	-3.8	-0.0	-3.8
2_2_2_29	0.4	-20.8	-0.1	-29.3	-0.1	-29.3
3_3_3_00	4.1	-5.7	3.4	-6.5	3.4	-6.5
3_3_3_01	0.8	-10.5	-0.1	-10.4	-0.1	-10.4
3_3_3_02	0.9	-9.5	5.5	-18.8	-0.0	-23.0
3_3_3_03	-0.5	-4.8	-0.0	-5.6	-0.0	-5.6
3_3_3_04	0.8	-4.8	-0.0	-6.2	-0.0	-6.2
3_3_3_05	1.8	-13.9	15.8	-12.8	-0.4	-25.0
3_3_3_06	0.6	-10.6	-0.1	-23.9	-0.1	-23.9
3_3_3_07	0.6	-13.3	0.0	-27.5	0.0	-27.5
3_3_3_08	-0.4	-19.2	4.1	-26.8	1.4	-28.8
3_3_3_09	9.5	-21.7	13.7	-28.1	9.6	-30.7
3_3_3_10	0.3	-21.1	-0.0	-28.7	-0.0	-28.7
3_3_3_11	0.7	-22.7	0.8	-31.2	0.8	-31.2
3_3_3_12	0.2	-7.5	-0.0	-5.0	-0.0	-5.0
3_3_3_13	-0.4	-4.3	-0.2	-5.1	-0.2	-5.1
3_3_3_14	2.1	-11.3	3.7	-21.2	1.9	-22.5
3_3_3_15	0.3	-14.2	4.9	-20.0	-0.0	-23.7
3_3_3_16	0.5	-23.4	0.4	-31.3	-0.0	-31.6
3_3_3_17	0.4	-12.8	2.1	-24.6	-0.2	-26.2
3_3_3_18	0.7	-3.2	-0.0	-3.6	-0.0	-3.6
3_3_3_19	0.0	-19.4	5.5	-24.5	0.1	-28.4
3_3_3_20	6.4	-2.3	5.5	-2.0	4.2	-3.2
3_3_3_21	1.9	-10.7	9.4	-16.7	2.1	-22.2
3_3_3_22	5.8	-7.6	11.1	-13.4	3.2	-19.5
3_3_3_23	0.4	-24.5	0.0	-32.9	0.0	-32.9
3_3_3_24	0.1	-21.9	5.1	-25.8	-0.2	-29.5
3_3_3_25	0.0	-17.7	0.5	-27.3	0.5	-27.3
3_3_3_26	3.1	4.7	4.2	5.0	1.7	2.5

	TTRP_OPT, time cost incl.		VRPTT_LB, time cost excl.		VRPTT_UB, time cost excl.	
instance	VRPTT	VRPTTMU	VRPTT	VRPTTMU	VRPTT	VRPTTMU
3_3_3_27	2.1	-15.5	4.1	-19.2	0.7	-21.8
3_3_3_28	-0.3	-19.6	-0.0	-27.5	-0.0	-27.5
3_3_3_29	0.8	-17.7	0.3	-26.1	0.3	-26.1
4_4_4_00	8.0	0.6	NaN	NaN	NaN	NaN
4_4_4_01	4.1	-13.3	5.9	-21.5	3.0	-23.7
4_4_4_02	3.5	-8.3	NaN	NaN	NaN	NaN
4_4_4_03	3.4	-16.3	19.4	-11.6	3.9	-23.0
4_4_4_04	5.7	9.8	NaN	NaN	NaN	NaN
4_4_4_05	1.2	-18.5	7.2	-23.3	1.5	-27.4
4_4_4_06	7.0	-0.4	21.6	13.3	-0.7	-7.5
4_4_4_07	0.6	5.9	NaN	NaN	NaN	NaN
4_4_4_08	0.6	-19.0	2.9	-25.4	0.7	-26.9
4_4_4_09	7.4	-10.3	13.7	-13.8	3.7	-21.3
4_4_4_10	0.7	-14.1	10.1	-17.6	0.2	-25.0
4_4_4_11	2.2	-16.8	9.8	-20.0	-0.1	-27.2
4_4_4_12	7.2	-11.0	15.8	-14.8	3.0	-24.3
4_4_4_13	4.4	-1.1	11.8	7.6	3.1	-0.8
4_4_4_14	7.2	-4.8	22.4	10.7	3.4	-6.4
4_4_4_15	6.1	-16.3	4.3	-22.9	4.3	-22.9
4_4_4_16	16.0	11.7	NaN	NaN	NaN	NaN
4_4_4_17	12.2	-1.9	NaN	NaN	NaN	NaN
4_4_4_18	11.6	-3.0	22.2	11.2	5.5	-4.0
4_4_4_19	10.7	15.8	NaN	NaN	NaN	NaN
4_4_4_20	8.0	6.3	NaN	NaN	NaN	NaN
4_4_4_21	4.6	-7.2	16.4	-6.4	1.7	-18.2
4_4_4_22	17.9	-4.8	NaN	NaN	NaN	NaN
4_4_4_23	14.3	11.1	NaN	NaN	NaN	NaN
4_4_4_24	1.2	0.8	6.9	-7.7	-2.1	-15.5
4_4_4_25	0.9	-15.0	13.0	-14.4	-11.9	-33.3
4_4_4_26	0.5	-0.3	4.3	-8.8	-0.3	-12.9
4_4_4_27	10.3	-9.8	NaN	NaN	NaN	NaN
4_4_4_28	6.8	-16.9	20.1	-16.2	5.4	-26.5
4_4_4_29	2.8	-8.7	28.8	7.3	-12.6	-27.2

References

[1] U.S. Environmental Protection Agency. Inventory of u.s. greenhouse gas emissions and sinks: 1990-2013. Technical report, 2015.

[2] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[3] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[4] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*, volume 18. Siam, 2014.

[5] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673, 2014.

[6] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.

[7] Michael Drexl. Rich vehicle routing in theory and practice. *Logistics Research*, 5(1-2):47–63, 2012.

[8] Michael Drexl. Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, 227(2):275–283, 2013.

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

[10] Jean-Francois Cordeau, Michel Gendreau, Gilbert Laporte, Jean-Yves Potvin, and François Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research society*, pages 512–522, 2002.

[11] Renaud Masson, Fabien Lehuédé, and Olivier Péton. The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12–23, 2014.

[12] Michael Drexl et al. A generic heuristic for vehicle routing problems with multiple synchronization constraints. Technical report, 2014.

[13] Michael Drexl, Julia Rieck, Thomas Sigl, and Bettina Press. Simultaneous vehicle and crew routing and scheduling for partial-and full-load long-distance road transport. *BuR-Business Research*, 6(2):242–264, 2013.

[14] Renaud Masson, Fabien Lehuédé, and Olivier Péton. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355, 2013.

[15] Michael Drexl and Hans-Jürgen Sebastian. On some generalized routing problems. Technical report, Deutsche Post Lehrstuhl für Optimierung von Distributionsnetzwerken (NN), 2007.

[16] G van Rossum et al. Python programming language. *URL http://www. python. org*, 1989.

[17] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2016-08-11].

[18] Continuum Analytics. Numba. *Version 0.14. url: http://numba. pydata. org/numba-doc/0.14/index. html*, 2014.

[19] Michael Drexl. 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*, 12:5–38, 2011.

[20] Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

[21] Robert W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 35(5):345, 1962.