Northumbria Research Link

Citation: Brandstätter, Christian and Reimann, Marc (2018) The Line-haul Feeder Vehicle Routing Problem: Mathematical model formulation and heuristic approaches. European Journal of Operational Research, 270 (1). pp. 157-170. ISSN 0377-2217

Published by: Elsevier

URL: https://doi.org/10.1016/j.ejor.2018.03.014 <https://doi.org/10.1016/j.ejor.2018.03.014>

This version was downloaded from Northumbria Research Link: http://nrl.northumbria.ac.uk/id/eprint/34036/

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: http://nrl.northumbria.ac.uk/policies.html

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)

Northumbria University NEWCASTLE

CUSTOMER SERVICE EXCELLENCE CSE UniversityLibrary

The Line-haul Feeder Vehicle Routing Problem: Mathematical model formulation and heuristic approaches

Christian Brandstätter[a], Marc Reimann[a,b]

[a] *Department of Production and Logistics, University of Graz, Austria*
[b] *Newcastle Business School, Northumbria University, UK*

Abstract

This paper deals with a rather new version of the well known Vehicle Routing Problem (VRP) called the Line-haul Feeder Vehicle Routing Problem (LFVRP). It can be described as a VRP with synchronisation constraints where two types of customers are served by two types of vehicles. These vehicles, contrary to the regular VRP, can meet and perform a transshipment to extend their travel. To achieve that, the vehicles need to synchronise - meaning that they have to be at the same place at the same time. The objective of the LFVRP is to minimize the total cost consisting of fixed vehicle cost as well as variable fuel and wage costs for drivers. For this problem we propose the first general mathematical model and derive two heuristics inspired by some structural insights about the problem. Using a thorough and comprehensive computational analysis we show the benefits of the LFVRP over simpler VRP variants, the quality of the heuristics compared with earlier work on the LFVRP and the relative performance of the two heuristics described as a function of different problem characteristics.

Key words: transportation, vehicle routing problem, heuristic approaches, mathematical model, synchronisation

1. Introduction

All major cities around the world have many things in common. Many inhabitants, traffic jams, air pollution and the desire of the people to live in the city are only a few of these things. Naturally, some problems also arise with these challenges. First, given the fact that a lot of people live in the city, a rising demand for supplies arises. These demands need to be satisfied and therefore city logistics becomes more and more important. Second, as so many people want to live in the city, land prices are rapidly increasing. Which means, that space is not only expensive but also limited. In major cities it is common that several inner-city areas exist where hardly any parking spaces exist and narrow streets make it difficult for logistics companies to satisfy all customer demands. In addition to that, the depots for the goods to be supplied are usually located outside the city in rural areas due to the high land prices in the cities. To sum up, a logistics company needs to face several challenges at once. Many customers need to be serviced from a depot that is located far away, which results in longer travel times and

Email addresses: `christian.brandstaetter@edu.uni-graz.at` (Christian Brandstätter), `marc.reimann@uni-graz.at` (Marc Reimann)

distance. Narrow streets make it difficult for large vehicles to drive through the city. And given that it is difficult to use large vehicles, it is only logical to use mainly small vehicles (e.g. car, motorcycle or bicycle). Yet, using small vehicles results in another difficulty. A small vehicle has only limited capacity and therefore it is necessary to reload more often. Thus, new research fields regarding the VRP arise and one of them is the Line-haul Feeder Vehicle Routing Problem (LFVRP).

The LFVRP has been originated from Taiwan and was first introduced in [6]. The problem occurred at the delivery of lunch boxes. Some customers, yet only the minority, could be served by a large vehicle. The remaining majority of customers lives in narrow streets where large vehicles cannot park nor turn and therefore smaller vehicles have to be used. To sum up, we have two types of customers and vehicles classes. Type-1 customers can be delivered by both vehicle classes, whereas type-2 customers can only be delivered by the small vehicle class. Up to this point, this problem can be seen as the Site-Dependent Vehicle Routing Problem (SDVRP- we will describe that problem in more detail in Section 2). But the crucial characteristic of the LFVRP distinguishes it from the other many VRP variants. This characteristic is the synchronisation between the large and small vehicle class. In other words, a large and small vehicle can meet and perform a transshipment of goods. As a result, the small vehicle does not have to return to the depot for a reload. But the two vehicles have to be at the same place at the same time. As mentioned before, type-1 customers can be delivered by both vehicle classes. Thus, it is assumed, that type-1 customers have enough space for both vehicles to meet and perform a transshipment.

The contribution of this paper is twofold. First, our goal is to enhance our understanding of the general problem and we present a mathematical programming formulation to formalize our model description. Second, based on our theoretical insights, we propose two heuristic approaches to find near-optimal solutions. To evaluate their performance we carry out a thorough and comprehensive numerical analysis on test instances from prior research. The structure of the remaining part of this paper is as follows. In Section 2, we review existing papers on the LFVRP and closely related VRP variants. In Section 3, we provide some structural insights that show the potential advantage of the LFVRP over the VRP and will form the basis for the heuristics developed later in the paper. This is followed by the mathematical formulation in Section 4. Our heuristics are described in Section 5 and the computational tests are reported on in Section 6. We close with some summarizing remarks and an outlook on open future questions with respect to solving the problem.

2. Literature Review

The LFVRP can be seen as a variant of the well known Vehicle Routing Problem (VRP), which was first introduced by [11] and extended by [8] with their famous savings algorithm. The VRP basically consists of a defined number of customers with known location, demand and service time, one physical depot and a homogeneous fleet of vehicles. Each customer must be served (therefore only visited once) and all vehicle tours must start from and return to the depot at a minimum cost (usually the shortest distance). Despite its simple description the VRP is a hard problem belonging to computational complexity class NP. This complexity transfers to all the richer problem variants including the LFVRP thereby driving the research for efficient heuristic approaches.

An important extension of the VRP is the Vehicle Routing Problem with Time Windows (VRPTW) where all customers must be served within a certain time window. Usually the time window is described with an earliest starting time (EST) and latest starting time (LST). In [25] some heuristics for the VRPTW are proposed, including his well known I1 heuristic, along with 56 benchmark test instances. A good overview of the solution approaches for the VRPTW can be found in [1] and [2]. For an overview of other VRP variants the reader is advised to refer to conducted surveys by [9] or [18].

Some variants of the VRP have close similarities to the LFVRP. One of them, as already mentioned within the introduction section, is the Site-Dependent Vehicle Routing Problem (SDVRP). The characteristics of the SDVRP are a mixed fleet and a defined vehicle-customer relationship, which means, that some customers can only be visited by a certain vehicle class. Further reading on the SDVRP can be found in [22], [10] and [3]. Another variant with great similarity is the Heterogeneous Fleet Vehicle Routing Problem (HFVRP: see [26] or [17] or [21]) or Fleet Size Mix (FSM: see [15] or [19]) where at least two different classes of vehicles are used, that differ in some characteristics (e.g. capacity and costs).

A rather new field is the VRP research with synchronisation constraints and therefore only a limited number of papers exist. In general, compared to the VRP and its variants, the vehicle tours (at least some) are no longer independent. In other words, two or more vehicles may meet and perform a task together. In [13] a survey is presented, classifying these problems as Vehicle Routing Problem with Multiple Synchronisation Constraints (VRPMSs). In addition to that, [12] also presented the Vehicle Routing Problem with Trailers and Transshipments (VRPTT) which is quite similar to the LFVRP. The VRPTT also has two types of customers (one type can be served by only one vehicle class whereas the other customer type can be serviced by both vehicle classes) and two vehicles classes. The main difference to the LFVRP is that only one vehicle class (lorries) can move without the other. The second vehicle class (trailers) can move only if it is pulled by a lorry. Some branch-and-cut solution algorithms for the VRPTT are presented in [14]. Another type of synchronisation is dealt within the work on the VRPTWMD by [24]. Using a similar motivation of narrow streets and missing parking opportunities in a densely populated city, they propose a two stage routing solution, where service workers serve a subset of customers on walking tours originating and ending at a customer location where the vehicle waits. Once all service workers returned and got on the vehicle, the vehicle moves on.

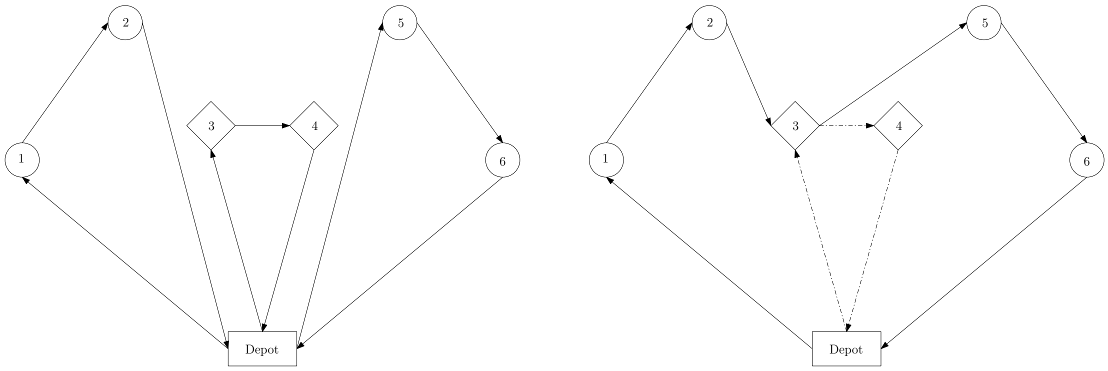
Although these problems are similar to the LFVRP they still differ in crucial aspects like customer distribution, number of vehicle classes and interdependence of vehicle tours which makes a direct comparison difficult. First, except for the VRPTWMD and the VRPTT, the above mentioned approaches use several (small) vehicle classes which differ only slightly in capacity and costs. We, on the other hand, only have two vehicle classes and the difference in vehicle capacity and cost is huge. Moreover, in those previous approaches the tours are independent and the assignment of customers to specific vehicle classes may be unique. Moreover, some of the customers may be served by a subset of (or all) vehicle classes and the vehicles need to be synchronised. Yet, in the VRPTWMD and the VRPTT, at any given time only one of the vehicle classes is moving. In our model both vehicles move and therefore planning for the meeting and synchronisation of those vehicles is much more complicated.

To the best of our knowledge, only four scientific papers exist on the LFVRP. It was introduced in [6] as the Line-haul Feeder Vehicle Routing Problem with Virtual Depots (LFVRPVD). Two heuristics (cost-sharing and threshold method) were proposed and tested on 17 test instances. In [5] time windows were added and a two-stage algorithm using Tabu-Search was developed. For the first two papers, several simplifications have been made, like assuming only one large vehicle with no capacity limit or allowing multiple reloads simultaneously. In the third paper, [7] lifted some of the limitations (e.g. number of large vehicles) mentioned before and proposed a new approach for the initial solution (first stage). In the fourth and final paper, [4] analyses four issues (different solution algorithms, different levels of customer demands, number of VD candidates and looser time windows) on the LFVRPTW and conducted a sensitivity analysis on 15 test instances selected from the Solomon test instances set. While these four papers set the ground for studying the LFVRP, the authors did not provide a mathematical formulation for the studied problem. Thus, we decided to perform our own structural analysis, provide the first general mathematical model for the LFVRP and devise some heuristics inspired by the insights of our structural analysis.

3. Structural Problem Analysis

In order to gain a basic understanding of the dynamics associated with the transshipment between vehicles as well as to understand the potential benefits over the classic VRP we decided to analyse some toy examples. Specifically, we consider a simple instance of six customers, two of type-1 and four of type-2, with one physical depot. All customers have an equal demand of one unit. As for the vehicles, we have small vehicles (SV) with a capacity of two and large vehicles (LV) with a capacity of four. The locations of the customers (type-1 as diamonds, type-2 as circles) and the solution for the VRP and LFVRP are presented in Figure 1.

The optimal VRP solution of the first example with an objective value of 391.59 (shown in Figure 1a) is small vehicle 1: $[Depot \rightarrow 1 \rightarrow 2 \rightarrow Depot]$, small vehicle 2: $[Depot \rightarrow 3 \rightarrow 4 \rightarrow Depot]$ and small vehicle 3: $[Depot \rightarrow 5 \rightarrow 6 \rightarrow Depot]$. Whereas the optimal LFVRP solution with an objective value of 301.20 (shown in Figure 1b) is small vehicle 1: $[Depot \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow Depot]$ and large vehicle 2: $[Depot \rightarrow 3 \rightarrow 4 \rightarrow Depot]$. The gain of almost 25% reduction in distance is given by the fact that the small vehicle 1 does not have to return to the depot to reload but rather can continue its tour after reloading at customer 3. Besides that, an obvious

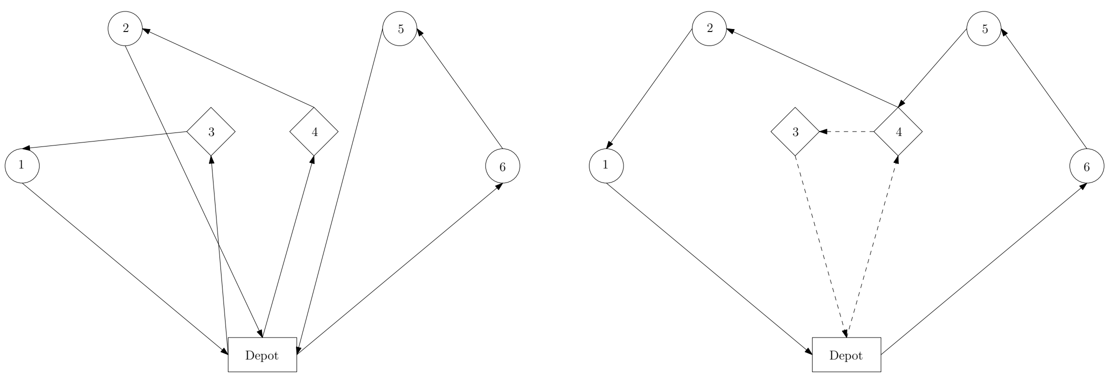


(a) Solution VRP (b) Solution LFVRP

Figure 1: VRP to LFVRP Solution (Example 1)

improvement is that only one small vehicle is used in the LFVRP solution, while the VRP tour of small vehicle 2 has been performed by the large vehicle in the LFVRP enabling the transshipment at customer 3. By looking at that solution we realize that this transshipment breaks the giant tour of the small vehicle into two parts thereby satisfying the capacity constraints. We will use this insight in developing our heuristic *Split-Approach* described in Section 5 below.

However, if we change some of the problem characteristics (capacity of the small vehicle class to three and the demand of customer 4 to two), we can gain a totally different solution as shown in Figure 2.



(a) Solution VRP (b) Solution LFVRP

Figure 2: VRP to LFVRP Solution (Example 2)

The optimal VRP solution of this second example with an objective value of 404.85 (visualized in Figure 2a) is given by: is small vehicle 1: $[Depot \rightarrow 3 \rightarrow 1 \rightarrow Depot]$, small vehicle 2: $[Depot \rightarrow 4 \rightarrow 2 \rightarrow Depot]$ and small vehicle 3: $[Depot \rightarrow 6 \rightarrow 5 \rightarrow Depot]$. The optimal LFVRP solution with an objective value of 301.20 (shown in Figure 2b) is small vehicle 1: $[Depot \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow Depot]$ and large vehicle 2: $[Depot \rightarrow 4 \rightarrow 3 \rightarrow Depot]$. We again save two small vehicles by using one large one. Moreover, the distance gain is again about 25%. Yet the structure of the LFVRP relative to the VRP is very different compared with example 1. Now, the possibility of transshipment at customer 4 seems to enable the linkage of the three small vehicle tours from the VRP. We will use this insight as a basis in our heuristic *Linkage-Approach* described in Section 5 below.

Finally, from both examples (but immediately from example 1) it becomes clear that the more decentralised the depot gets, the greater the benefit of the LFVRP should be. In our numerical analysis below we will investigate this relationship and its link with other problem parameters like transshipment times or small vehicle capacities in detail.

4. Mathematical Model

The LFVRP can be modeled through a graph consisting of a set of nodes V representing one physical depot (node 0) as well as a set of customers C - which is divided into type-1 (set A) and type-2 (set B) customers - and a set of edges E which correspond to the travel links between any two nodes i and j. With each edge a travel cost/time c_{ij}^T is associated. The customers have a demand d_i, service time t_i^S and time windows with earliest starting time a_i and latest starting time b_i. Further a fleet of vehicles F is available. This fleet is divided into small (F_{SV}) and large (F_{LV}) vehicles, respectively. The capacity of a given vehicle k is given by Q^k.

All vehicle tours must start and end at the physical depot (PD) and the duration of each tour must be within the given time limit. Large vehicles are only allowed to visit customers of type-1 whereas small vehicles can visit both customer types. Visiting the customers in the LFVRP is different than in the usual VRP variants. Usually, customers may only be visited once, but for the LFVRP that is only true for customers of type-2. Type-1 customers can be visited multiple times (and by different vehicles) because they provide enough parking space, so that a large vehicle can act as a virtual depot (VD). In other words, a large vehicle can park at a customer of type-1 and meet with different small vehicles to perform multiple transshipments. To model the scheduling of service and transshipment we introduce the task sequence $m \in T$. Below, we summarize the parameters used in our model.

Parameters:

A	set of type-1 customers
B	set of type-2 customers
$C = \{A \cup B\}$	set of all customers excl. depot
$V = \{C \cup 0\}$	set of all customers incl. depot
$i, j \in V$	customer/node index variables
a_i	earliest starting time at node i
b_i	latest starting time at node i
d_i	demand at node i
c_{ij}^T	costs/travelling-time from node i to j
t_i^S	service time at node i
F_{SV}	fleet of small vehicles
F_{LV}	fleet of large vehicles
$F = \{F_{SV} \cup F_{LV}\}$	fleet of vehicles
$k, k' \in F$	vehicle index
Q^k	max. capacity of vehicle k
N_{max}	max. no. of positions
$N = \{0, ..., N_{max}\}$	set of positions
$n, n' \in N$	sequence position index
T_{max}	max. no. of tasks
$T = \{1, ..., T_{max}\}$	set of tasks
$m \in T$	task sequence number
$Time^{UB}$	upper bound for time
μ	time needed for reload of one unit of demand
ν	time needed for transshipping one unit of demand
M	sufficient large number
z	total cost
f_1^{SV}	fixed cost for a small vehicle
f_1^{LV}	fixed cost for a large vehicle
f_2^{SV}	variable fuel cost for a small vehicle
f_2^{LV}	variable fuel cost for a large vehicle
f_3^{SV}	variable labor cost for a small vehicle
f_3^{LV}	variable labor cost for a large vehicle

In our model formulation we need five groups of assignment-type decision variables. The first group of decision variables (x) decides if a vehicle visits a customer i, the second group (y) if a vehicle k is used, the third group is for the service (α) of the customer by vehicle k, the fourth group is for the transshipment (β) between vehicles k and k' and the fifth group (γ) deals with reloads at the depot. Moreover, time (t) and load (q) variables reflect the route progress as result of the service and transshipment tasks. As we allow multiple visits and multiple tasks at a customer of type-1, we have to add some additional auxiliary variables to the model. These auxiliary variables are (ϵ) for time and (ω) for load. Furthermore, a sequence position index $n \in N$ is introduced to manage multiple customer visits. While $n = 0$ represents the starting position of vehicle k at the PD, $n = 1$ means that customer i is the first to be visited after leaving the PD.

Decision Variables:

$$x^k_{n,i} = \begin{cases} 1, & \text{if vehicle } k \text{ visits node } i \text{ at position } n \\ 0, & \text{else} \end{cases}$$

$$y^k = \begin{cases} 1, & \text{if vehicle } k \text{ is used} \\ 0, & \text{else} \end{cases}$$

$$\alpha^k_{n,m,i} = \begin{cases} 1, & \text{if vehicle } k \text{ serves customer } i \text{ at position } n \text{ at sequence } m \\ 0, & \text{else} \end{cases}$$

$$\beta^{k,k',m}_{n,n',i} = \begin{cases} 1, & \text{if large vehicle } k \text{ at position } n \text{ and small vehicle } k' \\ & \text{at position } n' \text{ perform a transshipment at} \\ & \text{customer } i \text{ at sequence } m \\ 0, & \text{else} \end{cases}$$

$$\gamma^k_{n,m,0} = \begin{cases} 1, & \text{if small vehicle } k \text{ reloads at the PD at position } n \\ & \text{at sequence } m \\ 0, & \text{else} \end{cases}$$

t_n^k	end-time at position n of vehicle k
$\epsilon_{n,m}^k$	end-time of task m at position n of vehicle k
q_n^k	end-load at position n of vehicle k
$\omega_{n,m}^k$	end-load of task m at position n of vehicle k
$c_{n-1,n}^k$	costs/distance of vehicle k from position $n-1$ to n
$q_{n,n',i}^{TS,k,k',m}$	transshipment load between large vehicle k at position n and small vehicle k' at position n' at sequence m at customer i.
$t_{n,n',i}^{TS,k,k',m}$	transshipment time between large vehicle k at position n and small vehicle k' at position n' at sequence m at customer i.
$q_{n,0}^{RL,k,m}$	reload quantity of vehicle k at position n at sequence m at the PD
$t_{n,0}^{RL,k,m}$	reload time of vehicle k at sequence m at sequence m at the PD

Our objective is to minimize the total costs, which depend on the utilized fleets of large and small vehicles, the respective fuel costs, which depend on the travelled distance as well as the drivers' labor cost which depend on the duration of the routes.

Objective Function:

$$\begin{aligned}\min z = & f_1^{SV} * \sum_{k\in F_{SV}} y^k + f_1^{LV} * \sum_{k\in F_{LV}} y^k \\ & + f_2^{SV} * \sum_{k\in F_{SV}} \sum_{n\in N|n\geq 1} c_{n-1,n}^k + f_2^{LV} * \sum_{k\in F_{LV}} \sum_{n\in N|n\geq 1} c_{n-1,n}^k \\ & + f_3^{SV} * \sum_{k\in F_{SV}} (t_{N_{max}}^k - t_0^k) + f_3^{LV} * \sum_{k\in F_{LV}} (t_{N_{max}}^k - t_0^k) \qquad (1)\end{aligned}$$

Constraints:

The following constraints (2) to (10) are simple routing constraints. Constraints (2) initialize the vehicles whereas the continuous flow from position n to $n+1$ is secured with (3). As we use position oriented decision variables, we have to make sure, that a vehicle can be at one customer only at a time (4). Constraints (5) and (6) ensure that all vehicle tours must ultimately end at a depot, but that a (small) vehicle can repeatedly return to the depot for a reload. Type-1 customers must be visited at least once but can have multiple visits (7), while customers of type-2 are only allowed to be visited once (8) by a small vehicle (9). Constraints (10) link the customer location oriented cost c_{ij}^T with the sequence oriented cost $c_{n-1,n}^k$, if vehicle k visits customers i and j

successively, whereas constraints (11) and (12) ensure that only used vehicles are considered for the objective function.

$$x^k_{0,0} = 1 \qquad \forall k \in F \tag{2}$$

$$x^k_{n,i} + x^k_{n+1,i} \leq 1 \qquad \forall k \in F, \forall i \in C, \forall n \in N, n < N_{max} \tag{3}$$

$$\sum_{i \in V} x^k_{n,i} = 1 \qquad \forall k \in F, \forall n \in N \tag{4}$$

$$x^k_{N_{max},0} = 1 \qquad \forall k \in F \tag{5}$$

$$x^k_{n,0} - \gamma^k_{n,1,0} + \sum_{i \in C} x^k_{n+1,i} \leq 1 \qquad \forall k \in F, \forall n \in N, 1 \leq n < N \tag{6}$$

$$\sum_{k \in F} \sum_{n \in N} x^k_{n,i} \geq 1 \qquad \forall i \in A \tag{7}$$

$$\sum_{k \in F} \sum_{n \in N} x^k_{n,i} = 1 \qquad \forall i \in B \tag{8}$$

$$\sum_{k \in F_{LV}} \sum_{n \in N} x^k_{n,i} = 0 \qquad \forall i \in B \tag{9}$$

$$c^k_{n-1,n} \geq c^T_{i,j} - M(2 - x^k_{n-1,i} - x^k_{n,j}) \qquad \forall k \in F, \forall n \in N, \forall i,j \in V, n \geq 1 \tag{10}$$

$$y^k \geq \sum_{i \in C} x^k_{n,i} \qquad \forall k \in F, \forall n \in N \tag{11}$$

$$y^k \geq 0 \qquad \forall k \in F \tag{12}$$

Constraints (13) to (23) model the service (α), transshipment (β) and reload (γ) decisions. Customers of type-2 can only have one service task and therefore the decision variable (α) is 0 for all other tasks $m \geq 2$ at the customer according to constraints (13). Analogously, at the depot there can only be a reload and therefore the decision variable (γ) is 0 for all other tasks $m \geq 2$ as modeled by constraints (14). Constraints (15) ensure that there is no service at the depot, while constraints (16) safeguard that every customer is serviced. To perform a service, transshipment or reload it is necessary that the vehicle is present at the respective customer or depot, respectively. That is ensured with constraints (17), (18), (19) and (20). Constraints (21) models that transshipments are only allowed at a customer of type-1, while constraints (22) and (23) secure that only one task (service or transshipment) is performed at a time on a large or small vehicle, respectively.

$$\sum_{m \in T} \alpha^k_{n,m,i} = 0 \qquad \forall k \in F_{SV}, \forall i \in B, \forall n \in N, m \geq 2 \tag{13}$$

$$\sum_{m \in T} \gamma^k_{n,m,0} = 0 \qquad \forall k \in F_{SV}, \forall n \in N, m \geq 2 \tag{14}$$

$$\sum_{k \in F} \sum_{n \in N} \sum_{m \in T} \alpha^k_{n,m,0} = 0 \tag{15}$$

$$\sum_{k \in F} \sum_{n \in N} \sum_{m \in T} \alpha^k_{n,m,i} = 1 \qquad \forall i \in C \tag{16}$$

$$\sum_{m \in T} \alpha^k_{n,m,i} \leq x^k_{n,i} \qquad \forall k \in F, \forall n \in N, \forall i \in V \tag{17}$$

$$\sum_{m \in T} \gamma^k_{n,m,0} \leq x^k_{n,0} \qquad \forall k \in F_{SV}, \forall n \in N \tag{18}$$

$$\beta^{k,k',m}_{n,n',i} \leq x^k_{n,i} \qquad \forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall i \in C, \forall m \in T \tag{19}$$

$$\beta^{k,k',m}_{n,n',i} \leq x^{k'}_{n',i} \qquad \forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall i \in C, \forall m \in T \tag{20}$$

$$\sum_{k \in F_{LV}} \sum_{n \in N} \sum_{k' \in F_{SV}} \sum_{n' \in N} \sum_{m \in T} \sum_{i \in B \cup \{0\}} \beta^{k,k',m}_{n,n',i} = 0 \tag{21}$$

$$\alpha^k_{n,m,i} + \sum_{k' \in F_{SV}} \sum_{n' \in N} \beta^{k,k',m}_{n,n',i} \leq 1 \qquad \forall k \in F_{LV}, \forall n \in N, \forall m \in T, \forall i \in C \tag{22}$$

$$\alpha^{k'}_{n',m,i} + \sum_{k \in F_{LV}} \sum_{n \in N} \beta^{k,k',m}_{n,n',i} \leq 1 \qquad \forall k' \in F_{SV}, \forall n' \in N, \forall m \in T, \forall i \in C \tag{23}$$

The next set of constraints (24) to (35) are necessary for capacity progress with scheduling. The lower and upper bound for the capacity is defined by (24) and (25). The auxiliary capacity variable (ω) models the capacity evolution within a node as a result of service and transshipments. The lower and upper bound for the auxiliary capacity are ensured by (26) and (27). Constraints (28) and (29) ensure that the starting auxiliary capacity is simply the capacity on arriving at the node and the leaving capacity q^k_n is equal to the auxiliary capacity after the last task $m = T_{max}$. Lower and upper bounds on the reload quantity are modeled by constraints (30) and (31).

Constraints (32) and (33) describe the lower and upper bound for the transshipment load. Constraints (34) and (35) consider the capacity progress for the small vehicle (transshipment load or reload quantity is added to the current capacity) and the large vehicle (current capacity is reduced by the transshipment load), respectively.

$$q_n^k \geq 0 \qquad \forall k \in F, \forall n \in N \tag{24}$$

$$q_n^k \leq Q^k \qquad \forall k \in F, \forall n \in N \tag{25}$$

$$\omega_{n,m}^k \geq 0 \qquad \forall k \in F, \forall n \in N, \forall m \in T \tag{26}$$

$$\omega_{n,m}^k \leq Q^k \qquad \forall k \in F, \forall n \in N, \forall m \in T \tag{27}$$

$$\omega_{n,0}^k = q_{n-1}^k \qquad \forall k \in F, \forall n \in N, n \geq 1 \tag{28}$$

$$q_n^k = \omega_{n,T_{max}}^k \qquad \forall k \in F, \forall n \in N, n \geq 1 \tag{29}$$

$$q_{n,0}^{RL,k,m} \geq 0 \qquad \forall k \in F_{SV}, \forall n \in N, \forall m \in T \tag{30}$$

$$q_{n,0}^{RL,k,m} \leq Q^{SV} \gamma_{n,0}^{k,m} \qquad \forall k \in F_{SV}, \forall n \in N, \forall m \in T \tag{31}$$

$$q_{n,n',i}^{TS,k,k',m} \geq 0 \qquad \forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall i \in V, \forall m \in T \tag{32}$$

$$q_{n,n',i}^{TS,k,k',m} \leq Q^{SV} \beta_{n,n',i}^{k,k',m} \qquad \forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall i \in V, \forall m \in T \tag{33}$$

$$\omega_{n',m}^{k'} = \omega_{n',m-1}^{k'} - \sum_{i \in V} (d_i \alpha_{n',m,i}^{k'}) + \sum_{k \in F_{LV}} \sum_{n \in N} \sum_{i \in V} q_{n,n',i}^{TS,k,k',m} + q_{n',0}^{RL,k',m} \tag{34}$$

$$\forall k' \in F_{SV}, \forall n' \in N, \forall m \in T$$

$$\omega_{n,m}^k = \omega_{n,m-1}^k - \sum_{i \in V} (d_i \alpha_{n,m,i}^k) - \sum_{k' \in F_{SV}} \sum_{n' \in N} \sum_{i \in V} q_{n,n',i}^{TS,k,k',m} \tag{35}$$

$$\forall k \in F_{LV}, \forall n \in N, \forall m \in T$$

Finally, the set of constraints (36) to (49) are required for time progress with scheduling. Figure 3 visualizes the time variables at a type-1 customer for a case with two transshipments under the assumption that the service is performed before the two transshipments. Note that this is only one of the twelve possible scenarios (six possible sequences, service performed by small or large vehicle). As it is unlikely that both vehicles arrive at the same time, usually a waiting time for one vehicle has to be considered. To master this complexity we introduced an auxiliary time variable (ϵ).

Constraints (36) and (37) define the lower and upper bound for the time in each node. Analogously, constraints (38) and (39) model these bounds for the auxiliary time variables. Constraints (40) and (41) ensure that the leaving time from the customer is equal to the auxiliary time after the last task $m = T_{max}$ and the starting point of the auxiliary time at a customer is the leaving time t_{n-1}^k from the previous customer plus the driving time $c_{n-1,n}^k$ to the current customer.

The relationship between reload quantity and time as well as transshipment load and time is modelled by constraints (42) and (43), respectively. The earliest and latest starting time for service at a customer i is accounted for in constraints (44) and (45), respectively. Constraints (46) to (49) define the auxiliary time with service and

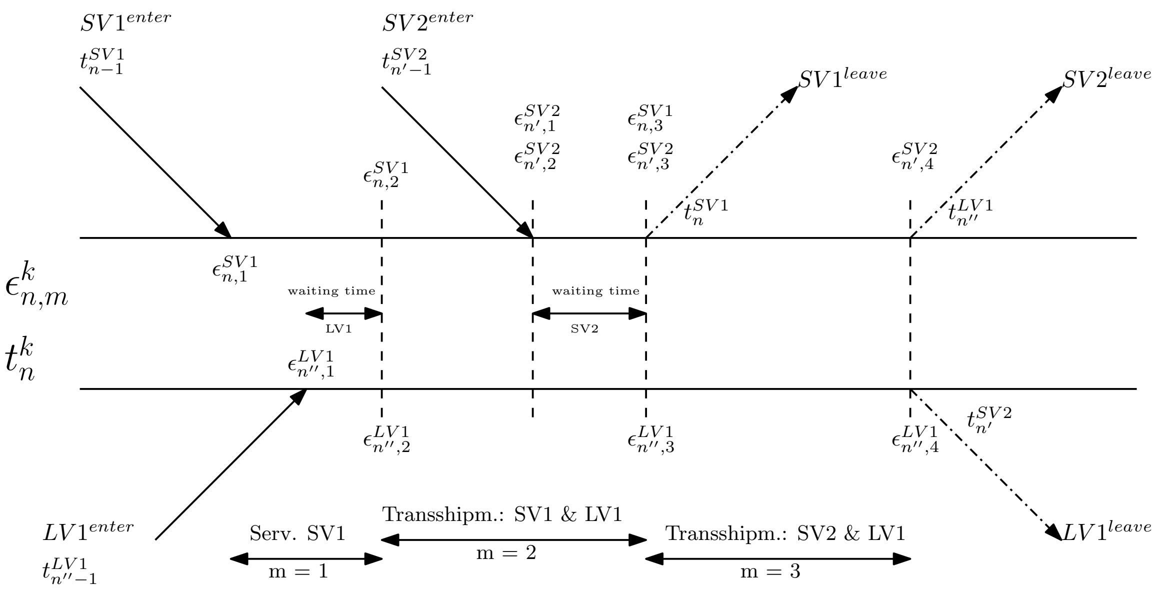


Figure 3: Time constraints explanation

transshipment. Specifically, constraints (46) and (47) model the time conservation for each vehicle individually. Moreover, when a transshipment occurs, it can only start when both vehicles are available and that is modeled through constraints (48) and (49). Note that the model uses M several times to link decision variables. Setting $M = Time^{UB}$ ensures that M is sufficiently large and the model works correctly.

$$t_n^k \geq 0 \qquad \forall k \in F, \forall n \in N \tag{36}$$

$$t_n^k \leq Time^{UB} \qquad \forall k \in F, \forall n \in N \tag{37}$$

$$\epsilon_{n,m}^k \geq 0 \qquad \forall k \in F, \forall n \in N, \forall m \in T \tag{38}$$

$$\epsilon_{n,m}^k \leq Time^{UB} \qquad \forall k \in F, \forall n \in N, \forall m \in T \tag{39}$$

$$t_n^k = \epsilon_{n,T_{max}}^k \qquad \forall k \in F, \forall n \in N, n \geq 1 \tag{40}$$

$$\epsilon_{n,0}^k = t_{n-1}^k + c_{n-1,n}^k \qquad \forall k \in F, \forall n \in N, n \geq 1 \tag{41}$$

$$t_{n,0}^{RL,k,m} = \mu q_{n,0}^{RL,k,m} \qquad \forall k \in F_{SV}, \forall n \in N, \forall m \in T \tag{42}$$

$$t_{n,n',i}^{TS,k,k',m} = \nu q_{n,n',i}^{TS,k,k',m} \qquad \forall k \in F_{LV}, k' \in F_{SV}, \forall n, n' \in N, \forall m \in T, \forall i \in C \tag{43}$$

$$\epsilon_{n,m-1}^k \geq a_i \alpha_{n,m,i}^k \qquad \forall k \in F, \forall n \in N, \forall m \in T, \forall i \in V \tag{44}$$

$$\epsilon_{n,m-1}^k \leq b_i + M(1 - \alpha_{n,m,i}^k) \qquad \forall k \in F, \forall n \in N, \forall m \in T, \forall i \in V \tag{45}$$

$$\epsilon^k_{n,m} \geq \epsilon^k_{n,m-1} + \sum_{i \in C}(t^S_i \alpha^k_{n,m,i}) + \sum_{k' \in F_{SV}} \sum_{n' \in N} \sum_{i \in C} t^{TS,k,k',m}_{n,n',i} \tag{46}$$

$$\forall k \in F_{LV}, \forall n \in N, \forall m \in T$$

$$\epsilon^{k'}_{n',m} \geq \epsilon^{k'}_{n',m-1} + \sum_{i \in C}(t^S_i \alpha^{k'}_{n',m,i}) + \sum_{k \in F_{LV}} \sum_{n \in N} \sum_{i \in C} t^{TS,k,k',m}_{n,n',i} + t^{RL,k',m}_{n',0} \tag{47}$$

$$\forall k' \in F_{SV}, \forall n' \in N, \forall m \in T$$

$$\epsilon^k_{n,m} \geq \epsilon^{k'}_{n',m-1} + \sum_{i \in C} t^{TS,k,k',m}_{n,n',i} - M(1 - \sum_{i \in C} \beta^{k,k',m}_{n,n',i}) \tag{48}$$

$$\forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall m \in T$$

$$\epsilon^{k'}_{n',m} \geq \epsilon^k_{n,m-1} + \sum_{i \in C} t^{TS,k,k',m}_{n,n',i} - M(1 - \sum_{i \in C} \beta^{k,k',m}_{n,n',i}) \tag{49}$$

$$\forall k \in F_{LV}, \forall k' \in F_{SV}, \forall n, n' \in N, \forall m \in T$$

The main contribution of the ILP formulation lies in the formal presentation of the model. We have tested the correctness of the model but only small instances could be solved to optimality. A thorough analysis of a slightly simplified model variant (without reload at the depot) can be found in [20]. For the sake of this paper, Appendix B provides a short comparison of the model results with the results of the heuristics which we developed to handle the problem. We will now present these heuristic approaches in detail.

5. Heuristic Approaches

In our structural analysis (see Section 3) we have already seen that, depending on the instance characteristics, the LFVRP may derive its main structure by either linking some of the optimal VRP tours through transshipments or by making an otherwise infeasible giant tour feasible by allowing transshipments. In this section we present our two heuristics based on these insights.

In developing these heuristics we first abstract from considering customer time windows. Clearly customer time windows are highly relevant in practice and have been very prominent in vehicle routing research. Yet, our aim with this paper is to get a first, basic understanding of the benefit associated with synchronisation. A specific structure of time windows may mask the pure effect and make the analysis less stringent. However, even without customer time windows, time constraints play a role in our model when it comes to synchronisation. In other words, if two vehicles meet, they have to be at the same place at the same time to perform a transshipment. Without time windows, it could be optimal for a vehicle to wait for a long time at a certain node in order to assure a proper synchronization. To avoid this we use a cost function that includes time-based cost, thereby balancing waiting times and travel times (see objective function (1)). Therefore the analysis in this paper can help us in understanding the influence of customer time windows in future work.

5.1. *Linkage-Approach:*

In Figure 1 of Section 3 we could already observe that it may be possible to modify the current VRP solution slightly to gain a LFVRP solution. Specifically, simply replacing two edges from and to the depot ($e_{0,5}$ and $e_{2,0}$ in the example) by two edges to a customer of type-1 ($e_{2,3}$ and $e_{3,5}$ in the example) where a large vehicle acts as a VD can extend a small vehicle tour and reduce the fleet by one small vehicle. In other words, the existing SV tours are linked together. Hence, we will refer to this heuristic as the *Linkage-Approach* for the remainder of this paper.

The *Linkage-Approach* is a very simple heuristic described in four steps as shown by the pseudo code presented in Algorithm 1.

Algorithm 1: Linkage-Approach

Step 1: Solve VRP for all type-2 customers;
Step 2: **for** *all tours* **do**
    while *current tour can be linked through a transshipment* **do**
        if *Inequalities* (50) *and* (51) *are satisfied* **then**
            link tours with a VD/PD;
        else
            return to the Depot (current tour ends at the Depot);
        end
    end
end
Step 3: insert all VDs into LV tours in chronological order;
Step 4: insert all remaining type-1 customers into SV/LV tours;
END;

In the first step we used Solomons I1 heuristic to get the solution for all type-2 customers. We initialize a tour with a randomly chosen seed-customer and add the remaining unserviced customers, using the two defined criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ with the parameters $\alpha = 1$, $\lambda = 2$ and $\mu = 1$ (For a more detailed description of the heuristic please refer to [25] or [1]). If no customer will fit into the tour anymore, a new one will be started and initialized again with a randomly chosen seed-customer. That procedure will be repeated until all type-2 customers are serviced. Without customer time windows we may (and will) also consider the reversed versions of the existing type-2 customer tours.

In the second step we then try to link pairs of existing tours through transshipment with a potential VD or reload at the PD. Clearly, in the latter case it is not necessary that a second (large) vehicle is present. Therefore, we select the first tour (let us call it k) as starting tour and sequentially check the linkage with all the remaining tours, starting with the second tour (let us call it k'), through all possible VDs, i.e. type-1 customers, and the PD. A linkage with a VD $j \in A$ is considered (temporarily saved) if the following two inequalities are satisfied (for a reload at the PD $j = 0$ only inequality (51) must be satisfied).

$$c^T_{i,j} + c^T_{j,i'} < c^T_{i,0} + c^T_{0,i'} \tag{50}$$

$$D^k + D^{k'} + t^{TS} - (c^T_{i,0} + c^T_{0,i'}) + (c^T_{i,j} + c^T_{j,i'}) \leq Time^{UB} \tag{51}$$

In these inequalities, i represents the last customer of the first tour, i' the first customer of the next tour, t^{TS} the transshipment time (or reload time if the PD is chosen), D^k the duration of the tour of vehicle k and $D^{k'}$ the duration for the tour of vehicle k'. As we also consider the reverse tours the following three additional scenarios are computed for each pair of tours and each VD and PD j:

a. i as last customer of the first tour and i' as last customer of the second tour.
b. i as first customer of the first tour and i' as first customer of the second tour.
c. i as first customer of the first tour and i' as last customer of the second tour.

After all pairs of tours have been checked with each VD/PD j, the first best fit solution will be selected and the two tours will we linked with the respective VD/PD. Now this linked tour serves as a starting tour and we again try to link it with yet another tour as above. If no linkage is possible, we select the next single tour as a starting tour and repeat this step until no more linkages are possible.

In the third step we generate the tours for the large vehicles. As we have already linked the small vehicle tours together, the locations and times of the necessary transshipments are already defined. Thus we sort those type-1 customers that have been selected as VDs in their chronological order and insert them into the first large vehicle tour. If no further VD can be inserted, a new large vehicle tour is used until all VDs are inserted.

In the last step we add all remaining type-1 customers into the small or large vehicle tours using once again Solomon's I1 approach like in Step 1. If a customer cannot be inserted into a large vehicle route anymore, an additional vehicle tour (SV or LV depending on the total required demand) will we used. Step 4 will be repeated until all type-1 customers are served.

5.2. Split-Approach:

Although it might be possible to link existing SV tours together, we will not solely rely on that circumstance. The reason for that can be seen in Figure 2 from the structural analysis section. Here a totally different LFVRP solution is presented and the resemblance to the VRP solution is much less pronounced. Consequently, we decided to propose a second approach to master the LFVRP. As the problem remains a VRP at its core, we decided to follow the approach of [23] and divide a giant tour into sub-tours to gain a feasible solution. Therefore, we will refer to it as the *Split-Approach* for the remainder of the paper.

Same as the *Linkage-Approach*, the *Split-Approach* consists of four main steps as given in the pseudo code presented in Algorithm 2. Observe that Steps 3 and 4 are identical to those used in the *Linkage-Approach*. The main difference between the two approaches lies in the first two steps, i.e. in the way, the small vehicle tours are constructed and associated with transshipments. Thus, we will now focus on the discussion of the first two steps of the *Split-Approach*.

Algorithm 2: Split-Approach

```
Step 1: Solve TSP for all type-2 customers;
Step 2: while giant tour is infeasible do
    select a split-customer;
    while sub-tour is infeasible do
        insert all necessary VDs or Depot visits to make sub-tour feasible;
        if feasible sub-tour is found then
            select best sub-tour and split from giant tour;
        else
            select previous customer as split-customer;
        end
    end
end
Step 3: insert all VDs into LV tours in chronological order;
Step 4: insert all remaining type-1 customers into SV/LV tours;
END;
```

In the first step, we create a giant TSP tour of all type-2 customers by ignoring the capacity and route duration restrictions. We employ the I1 heuristic of Solomon (using the same criteria, parameter settings and randomly chosen seed customer) in the same way as in Step 1 of the *Linkage-Approach*. Once again we also consider the reverse giant tour.

In the second step, we need to split the giant tour into feasible sub-tours. To achieve that, we first use a slightly modified variant of the split-procedure presented by [23]. Specifically, we considered only the route duration restriction for finding the optimal split candidates. Thus a route segment between two subsequent split candidates satisfies the route duration constraint, but not necessarily the capacity constraint. The latter problem will be dealt with by adding transshipments at VDs or reloads at the PD. However, since adding transshipments or reloads will increase route duration, the route segment between two subsequent split candidates may no longer be feasible after adding transshipments.

Thus, we look at the splits sequentially. We first consider the first split customer obtained by the modified split-procedure. To know how many transshipments we need to satisfy the capacity restriction, we sum up all the demands to the split customer. After that we try to insert the necessary VDs or PD visits at all possible positions in the sub-tour. When we insert a VD/PD between two customers i and i', we select the type-1 customer or PD j with the smallest distance value of $c^T_{i,j} + c^T_{j,i'}$. A resulting small vehicle tour is feasible, if the capacity limit of the vehicle and the duration of the tour do not exceed the respective limit (max. capacity of the vehicle and max. working time).

Let us consider a small, illustrative example. Assume we have obtained the following route segment by finding an optimal split at customer 5: $0 \to 1 \to 2 \to 3 \to 4 \to 5 (\to 0)$. We have also found that two transshipments will be necessary. Then we have six possible combinations that need to be considered.

a. $0 \to 1 \to VD/PD \to 2 \to VD/PD \to 3 \to 4 \to 5 \to 0$
b. $0 \to 1 \to VD/PD \to 2 \to 3 \to VD/PD \to 4 \to 5 \to 0$
c. $0 \to 1 \to VD/PD \to 2 \to 3 \to 4 \to VD/PD \to 5 \to 0$
d. $0 \to 1 \to 2 \to VD/PD \to 3 \to VD/PD \to 4 \to 5 \to 0$
e. $0 \to 1 \to 2 \to VD/PD \to 3 \to 4 \to VD/PD \to 5 \to 0$
f. $0 \to 1 \to 2 \to 3 \to VD/PD \to 4 \to VD/PD \to 5 \to 0$

The best feasible solution among these six combinations will be used, and the route segment will be separated from the giant tour. If none of the six options is feasible, i.e. they all increase route duration too much, we have to move up and consider the predecessor of the current split customer (in our example customer 4) as the new split customer. By repeating the inclusion of VDs or PD visits we again try to find a feasible solution. Once a feasible solution has been found, we split the route segment from the giant tour. Then Step 2 will be repeated until the remaining giant tour has been split into feasible SV tours in just the same way.

Finally, Steps 3 and 4 already discussed for the *Linkage-Approach* are used to find the tours for the VDs used to satisfy the transshipment requirements and service all customers of type-1.

6. Computational Results

The two proposed approaches were coded in programming language C and compiled with GNU GCC Compiler on a 4 * Intel® Core™ i7-5557U CPU @ 3.1 GHz processor and 16 GB DDR3 RAM (1.6 MHz) under a 64-Bit Operating System (Kubuntu 14.04).

6.1. Test Instances

Our experimental analysis will consist of two main parts. First, we will analyse the performance of our proposed approaches against previous research. To do so, we used the original LFVRP test instances proposed by [6]. They presented 17 test instances with a customer range between 34 and 161 customers and for each test instance 4 customers were selected as VDs (type-1 customer). Furthermore, several parameters have been proposed by [6], which we will describe briefly.

The capacity of the small vehicle depends on the test instance and the large vehicle capacity is unlimited. The costs (in New Taiwan Dollar NTD) for the small vehicle class are f_1^{SV} = 600 NTD/day (rent and insurance), f_2^{SV} = 0,77 NTD/km (fuel cost) and f_3^{SV} = 120 NTD/h (drivers wage). Furthermore the average driving speed for the small vehicle is 40 km/h (presented in [5]). The cost for the large vehicle are f_1^{LV} = 1450 NTD/day (rent and insurance), f_2^{LV} = 3.3 NTD/km (fuel cost) and f_3^{LV} = 270 NTD/h (drivers wage). No speed limit for the large vehicle has been proposed, therefore we assumed the same average speed as for the small vehicle.

The maximum allowed working time per day was presented with max. 8h/day and max. overtime of 4h/day. Since no extra cost for overtime were mentioned in [5] or [6] we decided to use 12 hours as the max. time limit for our algorithm.

According to [6], service and transshipment times are small and were therefore neglected. However, we designed our algorithm to include service and transshipment times, but to be able to compare our results we set them to 0.

Second, we will analyse the performance of our approaches and their benefit over the more traditional HFVRP on an extended set of benchmark instances. Those instances are derived from the 17 original test instances of [6] by systematically varying the following four parameters using three different values for each. To test the impact of the depot location we use: (i) the original depot location as given by [6], (ii) a depot located further away from the customer cluster at coordinate location (0,0), and (iii) a depot located far afield at coordinate location (-50,-50). To evaluate the impact of service and transshipment times we consider the three scenarios: (i) service and transshipment times are zero, (ii) service and transshipment times of 15 minutes for completely uploading the small vehicle, and (iii) service and transshipment times of 30 minutes for completely uploading the small vehicle. To study the effect of small vehicle capacity we distinguish the following three settings: (i) small vehicle capacity is 100 units, (ii) 50 units or (iii) very tight with 25 units. Finally, to understand the impact of customer distribution we analyse (i) a setting with few (12.5 %) type-1 customers, (ii) a setting with 25 % type-1 customers and (iii) a setting with 50 % type-1 customers.

6.2. Benchmark comparison

Since both of our approaches feature some stochastic aspects we first want to understand the robustness of the *Linkage-* and *Split-Approach* by comparing their results for a sample size of 10, 100 and 1000 (samples are independent) and conducting a statistical analysis afterwards. Besides the minimum (columns *Min.*) and mean

Instance	Sample	Linkage				Split			
		Mean	Std-Dev.	Min.	CPU	Mean	Std-Dev.	Min.	CPU
Average	10	11519.44	638.81	10778.10	12.393	12357.84	861.52	11374.50	43.083
Average	100	11541.01	668.67	10448.24	123.900	12478.99	918.24	11108.64	391.646
Average	1000	11533.83	682.91	9971.83	1245.300	12482.60	918.04	11071.92	3981.875
Average	total	11531.42	663.46	10399.39	460.531	12439.81	899.27	11185.02	1472.201

Table 1: Average results over the 17 problem instances obtained with the *Linkage-Approach* and the *Split-Approach* for 10, 100 and 1000 repetitions

values (columns *Mean*) as well as the standard deviation (columns *Std-Dev.*) of total cost, we also present the total required CPU-Time (columns *CPU*) in seconds in Table 1. The disaggregated results for each of the 17 instances are presented in Table 6 in Appendix A.

If we have a look at the mean values, we can observe that even with an increasing sample size, the average values hardly change. Therefore we can conclude that we will not gain a better average solution if we increase the no. of calculation cycles. Contrary to that, we find that the sample size has a substantial influence on the minimum values and the best solutions are always found within the sample of 1000 repetitions. Hence, we can regard these minimum values as benchmark for our further analysis as well as future research. Another interesting aspect can be observed if we look at the calculation times. Not only does the *Linkage-Approach* consume significantly less computation time than the *Split-Approach* on average, it is also less sensitive with respect to the no. of customers per instance as shown in Table 6 in Appendix A.

To compare the *Linkage-* and *Split-Approaches* in terms of solution quality, we performed the Mann-Whitney-U-Test for all three sample sizes and each instance. An equal solution quality of both approaches will be assumed by the Null-Hypothesis H_0 and a significant difference by the Alternative-Hypothesis H_1. The null hypothesis can be rejected when the p-value is less than 0.05 and Table 2 shows the results of the test (performed with R).

Overall, we observe that the *Linkage-approach* produces significantly better results for all but three (all but one) instances under 100 (1000) repetitions. For 10 repetitions, the results are more mixed, yet the *Linkage-Approach* still has an edge over the *Split-Approach*. Thus, from this statistical analysis we can conclude that over the 17 benchmark instances, the *Linkage-Approach* provides significantly better results than the *Split-Approach* in most cases.

Instance	p-value 10	p-value 100	p-value 1000
A32	**0,467**	**0,398**	**0,245**
A34	0,016	0,000	0,000
A38	0,000	0,000	0,000
A45	**0,269**	**0,459**	0,000
A46	**0,305**	**0,245**	0,000
A60	**0,131**	0,000	0,000
A61	0,001	0,000	0,000
A64	**0,730**	0,000	0,000
A65	0,002	0,000	0,000
A69	0,017	0,000	0,000
A80	0,002	0,000	0,000
C51	**0,466**	0,008	0,000
C81	**0,970**	0,000	0,000
C161	0,030	0,000	0,000
F45	0,000	0,000	0,000
F72	0,014	0,000	0,000
F135	0,004	0,000	0,000

Table 2: Statistical analysis of comparison between *Linkage*- and *Split-Approach*

Building on that analysis we next want to compare our results with the original proposed results provided by [6]. To also keep computation times comparable with those presented by [6] we decided to use 10 repetitions for each instance and choose the best solution among them. The results are presented in Table 3.

The bold values in Table 3 are the minimum cost values for the superior approach. What we can observe again is that the *Linkage-Approach* is superior to the other approaches most of the time. Moreover, both of our approaches seem to clearly outperform the previous work from [6]. Although our approaches use more large vehicles on average, we can perform more transshipments and therefore reduce the no. of small vehicles substantially. [6] on the other side, does not use a single transshipment in most cases. To statistically verify the relative performance of our approaches against the approach by Chen presented in [6] we performed a Wilcoxon-Signed-Rank test (WSRT: originally presented by [27]) and the results are shown in Table 4. These results clearly highlight that both our LFVRP approaches provide a significantly better solution quality than [6].

6.3. Sensitivity analysis

To get an even deeper insight into the performance of our *Linkage*- and *Split-Approaches* we performed a second series of tests on a larger set of benchmark instances as mentioned above.

Specifically, we were interested in the influence of the following four problem parameters (depot location,

	Chen[a]					Linkage					Split				
Instance	Cost	Dist[b]	SV	LV[c]	CPU[d]	Cost	Dist	SV	LV	CPU[d]	Cost	Dist	SV	LV	CPU[d]
A32	**7942**	n.a.	5	0	0.10	8065	1003.83	2	1	0.033	8337	949.76	2	1	0.048
A34	8106	n.a.	4	1	0.02	**7358**	897.92	2	1	0.042	8321	943.04	2	1	0.058
A38	**6909**	n.a.	4	0	0.06	7113	863.14	2	1	0.064	7635	761.31	2	1	0.154
A45	8852	n.a.	6	0	0.35	8686	1131.68	2	1	0.109	**8207**	964.85	2	1	0.173
A46	8852	n.a.	6	0	0.27	**7712**	1022.32	2	1	0.110	8597	1082.52	2	1	0.185
A60	15553	n.a.	9	1	1.77	11384	1601.55	4	1	0.248	**11313**	1450.10	3	1	0.534
A61	11155	n.a.	5	1	0.07	**8463**	1064.06	2	1	0.266	9579	1145.20	3	1	0.571
A64	12205	n.a.	9	0	1.45	**10519**	1432.19	3	1	0.394	10574	1270.45	3	1	0.729
A65	12204	n.a.	8	0	0.66	**10093**	1386.58	3	1	0.349	11368	1420.68	3	1	0.738
A69	13242	n.a.	7	1	1.33	**10079**	1405.63	3	1	0.453	10361	1262.94	3	1	0.950
A80	16837	n.a.	9	1	2.22	**12430**	1814.71	4	1	0.792	15984	1856.29	4	2	3.270
C51	7122	n.a.	3	1	0.31	**4969**	595.14	2	1	0.373	5735	572.03	1	1	0.392
C81	21142	n.a.	13	0	2.38	**15653**	2402.20	5	1	0.867	15894	2445.54	5	1	1.973
C161	82930	n.a.	48	0	57.79	37571	5994.15	12	2	6.913	**37536**	6000.06	13	2	28.406
F45	8971	n.a.	4	1	0.29	8707	1013.36	2	1	0.152	**7888**	966.36	2	1	0.207
F72	**3760**	n.a.	1	1	0.18	3874	350.79	1	1	1.446	4329	299.76	1	1	1.557
F135	11262	n.a.	8	0	14.72	**10551**	1557.30	3	1	8.636	11710	1422.12	3	1	15.944
Average	15120.24	n.a.	8.76	0.47	4.94	10778	1502.15	3.18	1.06	1.250	11375	1459.59	3.18	1.12	3.288

[a],[b] Data reported (Distance not reported) by [6].

[c] According to [6], only one LV is used but only if a VD is used. Hence, if no VD is used, all customers are serviced by the small vehicle class.

[d] CPU-Time in sec. for the best solution. As [6] used a different hardware configuration, CPU times are not comparable.

Table 3: Best Solution Comparison

service and transshipment times, small vehicle capacity and no. of type-1 and type-2 customers) on the solution quality in general, the relative performance of the two approaches compared with each other as well as with the simple benchmark of the Heterogeneous Fleet VRP (HFVRP) where customers of type-2 are served solely by small vehicles and type-1 by large vehicles.

As mentioned in our structural analysis, the LFVRP approaches should (compared with the simpler HFVRP) benefit from a more remote depot location since the possibility of transshipments will reduce the burden of the small vehicles returning to the physical depot for reloading.

We also varied the service and transshipment times. Clearly, shorter transshipment times should benefit our LFVRP approaches. Thirdly, we studied the impact of the small vehicle capacity. The smaller this capacity is, the more reloads/transshipments should be necessary, thereby requiring more synchronisation in the LFVRP. On the other hand, smaller capacity of type-2 vehicles should in general imply that the HFVRP needs more of those vehicles. Thus, it is not a-priori clear whether smaller capacity benefits or hurts the LFVRP approaches over the HFVRP. Finally, the number of type-1 and type-2 customers should play an important role. The conjecture would

Comparison	p-value[a]
Linkage-Chen	0.000
Split-Chen	0.006

Table 4: Statistical Analysis of Best Solution Comparison

be that an increase in type-1 customers should magnify the benefits of the LFVRP over the HFVRP. When there are too little type-1 customers it may be difficult to find effective transshipment possibilities of the small vehicles with the large vehicle(s).

Table 7 in Appendix A presents the associated disaggregated results. Specifically, we let all three approaches (HFVRP, *Linkage* and *Split*) run once for each instance and each parameter constellation. For each constellation of the four parameters, we then averaged over all 17 instances to obtain the results of the three approaches.

Let us now summarize the main findings here. To start out we present single-factor effects in Table 5, i.e. we focus on one parameter and average over all the other parameters. Specifically, we show the relative percentage deviation (RPD) of the *Linkage-Approach* and the *Split-Approach* over the HFVRP. The RPD is calculated as $RPD_* = 100\frac{Cost_{HFVRP} - Cost_*}{Cost_{HFVRP}}$, where $* \in \{Linkage, Split\}$. Clearly, the expected general trends hold. For example, when transshipment time increases, the benefit of the LFVRP vanishes, while a more remote depot location pronounces the LFVRP advantage over the HFVRP. We can also find our conjecture about the benefit of having more type-1 customers confirmed. A first interesting and strong result concerns small vehicle capacity. An increase in small vehicle capacity greatly improves the relative performance of the LFVRP over the HFVRP for the *Linkage-Approach*, while for the *Split-Approach* the benefit is largest when SV capacity is not too high. It seems that this result is driven by the structural differences between the solutions obtained with the two aproaches. The *Split-Approach*, starting from the giant tour, allows more options for synchronisation since any pair of successive clients on the giant tour yields a reasonable breakpoint for a detour to a virtual depot. This implies that the synchronisation is more efficient in the *Split-Approach*, such that smaller transshipment volumes are already beneficial. In the *Linkage-Approach*, the routes are predefined and can only be appended to each other at their respective ends, leaving less room for effective transshipments. Yet, when small vehicle capacities are larger to begin with the initial tours are more effective and the transshipment benefits magnify that.

Finally, we can observe another interesting result when comparing the *Linkage-* with the *Split-Approach.* While we did find that the former approach outperforms the latter one for the benchmark instances above, the effect is now reversed when the depot lies far afield. In that case, the *Split-Approach* produces an average improvement of almost 10% over the HFVRP, where the *Linkage-Approach* yields only around 6% cost reduction. This result is driven by a similar structural characteristic of the heuristics as discussed above. While the *Linkage-Approach* will build tours in a way, where first and last customers are relatively close to the physical depot, the benefits of transshipments at a virtual depot are relatively smaller than in the *Split-Approach*, where transshipment between any pair of customers, which may be very far from the physical depot, can be established.

		(i)	(ii)	(iii)
Depot Location	Linkage	2.56%	2.77%	6.13%
	Split	-2.13%	3.19%	9.72%
#t1 customers	Linkage	2.06%	4.81%	4.59%
	Split	0.54%	3.70%	6.53%
SV capacity	Linkage	-4.11%	4.37%	11.20%
	Split	-1.45%	6.97%	5.25%
Service and Transshipment time	Linkage	7.25%	3.75%	0.46%
	Split	8.35%	2.64%	-0.22%

(i) {Standard, 12.5%, 25, 0}
(ii) {(0,0), 25%, 50, 15}
(iii) {(-50,-50), 50%, 100, 30}

Table 5: Single Factor Effects – RPD

Summarizing, we find that – as expected – smaller transshipment times, a not too skewed distribution of type-1 and type-2 customers, a not too small SV capacity and a more remote physical depot location benefit the LFVRP the most. Moreover, the *Linkage-Approach* outperforms the *Split-Approach* when the depot is less remote, or the SV capacity is large, while the *Split-Approach* is preferable when the depot is far a field or the SV capacity is small.

7. Conclusion

In times of rising land prices, highly populated cities with narrow streets and increasing customer demands, new VRP problems in the context of city logistics arise and solution approaches need to be developed to master these problems. One of these problems is the LFVRP. The LFVRP allows a synchronisation between vehicles, which allows us to use small and large vehicles to satisfy all customer demands and a depot located outside of town.

With this paper, we provided an overview of the existing literature and presented a structural analysis to the problem at hand. After that we gave a mathematical formulation to formally describe the problem in general. The correctness of the model formulation was verified on small test instances. Our main aim in solving the problem was to focus on heuristic solution approaches that can be used for practically relevant problem sizes.

To master the LFVRP we developed two heuristic approaches based on insights from our structural analysis. The first approach, called the *Linkage-Approach*, starts with a VRP solution and tries to connect/link the tours together with a possible VD or the depot. The second approach, called the *Split-Approach*, starts with a TSP solution (one giant tour neglecting capacity and time restrictions) which we have to split into feasible sub tours using VDs or the depot.

Using a thorough numerical analysis and statistical testing we could show that our approaches clearly outperform the existing heuristics in terms of solution quality. We also investigated the influence of the random aspects of our algorithms by repeatedly running our approaches for 10, 100 and 1000 times. The results show that increasing the number of repetitions does not alter the average results significantly, but clearly helps to find new best solutions. These best solutions will be used as benchmarks in future work to compare more sophisticated meta- and matheuristic approaches against.

We finally performed a thorough sensitivity analysis to understand the effects of a (more) decentralised depot location, varying capacity of the small vehicles, efficiency of transshipment and reload operations as well as customer distribution in terms of their accessibility by small and large vehicles.

Summarizing, we find that only when the problem constellation is very unfavourable for transshipment, i.e. transshipment times are very long, there are only few transshipment options in terms of type-1 customers, small vehicle capacity is very small, and the physical depot is located centrally, the HFVRP may compete with the LFVRP. In all other cases, the benefits of synchronisation are substantial and can easily exceed 10%. Moreover, an interesting structural insight concerns the relative performance of our two proposed approaches. While the *Linkage-Approach* outperforms the *Split-Approach* when the depot is less remote, or SV capacity is large, we find that the *Split-Approach* benefits more strongly from its larger synchronisation potential when the depot is further afield or small volumes can be transshipped.

The work in this paper serves as a starting point for deeper investigation of the LFVRP. In a next step we have started to develop metaheuristic approaches to provide even better solutions to the problem. Another promising direction of research will be to include time windows, and more specifically to analyse the width and location of time windows on the potential of the synchronisation approach key to the LFVRP. Insights on that should help improving the crucial link between the operational routing decisions with the consumer related service agreements in terms of delivery timing.

Appendix A

Instance	Sample	Linkage				Split			
		Mean[a]	Std-Dev.[a]	Min. [a]	CPU[b]	Mean[a]	Std-Dev.[a]	Min. [a]	CPU[b]
A32	10	8694.47	411.51	8064.92	0.357	8563.53	194.74	8337.28	0.495
	100	8628.75	492.33	7712.61	3.257	8526.01	188.99	8337.28	4.831
	1000	8593.49	523.98	7225.64	32.760	8539.53	189.61	8031.89	47.705
A34	10	7932.74	471.02	7357.98	0.422	8588.91	348.26	8320.93	0.591
	100	7877.35	506.69	7337.85	3.984	8502.21	307.70	8320.93	5.882
	1000	7941.36	506.01	7205.71	40.850	8481.66	299.09	8094.19	59.230
A38	10	7355.87	192.95	7112.62	0.631	7940.05	210.56	7634.61	1.439
	100	7392.30	302.28	6970.31	6.350	7922.80	251.97	7146.50	13.939
	1000	7396.19	346.84	6921.60	62.969	7901.92	261.99	7146.50	138.953
A45	10	9331.21	483.47	8686.02	1.040	9039.84	902.59	8206.55	1.733
	100	9386.90	660.87	8258.01	10.273	9459.57	1239.53	8206.55	17.372
	1000	9313.34	563.54	7966.40	103.934	9279.86	1138.59	8206.55	173.921
A46	10	8482.23	391.49	7712.04	1.083	9013.87	738.11	8596.78	1.868
	100	8745.38	582.91	7822.62	10.869	8896.23	592.99	8434.45	18.845
	1000	8674.42	430.77	7710.23	108.806	8970.96	657.76	8422.96	188.286
A60	10	11929.05	453.53	11383.80	2.565	12801.71	1027.42	11312.84	5.377
	100	12021.97	858.79	10784.44	25.827	13186.41	820.81	11312.84	53.906
	1000	12003.17	823.57	10177.00	257.397	13154.59	829.96	11312.84	539.460
A61	10	9640.92	710.09	8463.47	2.650	11786.43	1567.20	9578.63	6.377
	100	9665.18	638.35	8296.36	26.513	11773.99	1372.63	9392.37	62.705
	1000	9770.00	732.03	8275.57	265.095	11843.78	1473.32	9392.37	637.496
A64	10	11125.32	620.53	10518.67	3.924	10871.56	313.47	10574.18	7.417
	100	11242.01	811.64	10105.14	39.153	10716.12	250.45	10574.18	73.664
	1000	11064.13	725.86	9913.29	391.868	10684.90	228.03	10574.18	738.066
A65	10	10869.85	778.44	10093.04	3.446	11593.91	587.65	11367.60	7.423
	100	11277.67	972.64	9916.08	34.517	12158.34	965.41	10798.35	88.822
	1000	11283.76	1020.65	9741.87	344.421	12062.17	860.45	10798.35	862.842
A69	10	10467.20	293.17	10079.42	4.580	12343.03	1887.20	10360.61	9.579
	100	10569.19	607.01	9700.54	45.445	12421.26	1833.53	10047.24	97.172
	1000	10504.83	589.67	9557.96	454.629	12402.12	1749.14	10047.24	966.246
A80	10	13868.64	1353.86	12430.29	7.850	16575.96	781.46	15983.88	26.034
	100	14297.32	1423.27	12263.54	79.522	17055.78	911.42	14560.65	241.716
	1000	14282.36	1390.60	12045.61	793.148	17057.02	912.01	14560.65	2468.065
C51	10	5543.37	352.93	4969.04	3.722	5736.60	1.95	5734.75	3.931
	100	5393.66	391.66	4952.37	37.376	5774.77	115.67	5734.75	40.318
	1000	5373.25	411.65	3412.16	371.306	5786.53	129.48	5734.75	404.859
C81	10	16768.53	1172.76	15652.76	8.816	16874.96	1052.51	15893.85	22.140
	100	16602.35	857.33	15294.68	87.516	17474.15	2059.79	15893.85	221.950
	1000	16670.78	974.14	14973.48	877.624	17412.38	1965.97	15893.85	2215.933
C161	10	39104.44	1595.93	37571.41	69.170	41301.45	2250.54	37536.37	289.783
	100	38643.85	1208.25	35900.01	692.640	41788.64	2040.52	37536.37	2848.494
	1000	38583.31	1322.60	32503.40	6941.851	42086.61	2222.47	37536.37	28585.727
F45	10	8762.18	97.54	8707.37	1.602	8249.41	380.67	7888.28	2.124
	100	8820.57	147.36	8453.86	16.069	8281.04	366.75	7776.12	21.226
	1000	8822.41	131.14	8398.44	160.805	8316.91	358.41	7776.12	212.037
F72	10	4076.22	225.16	3874.33	14.694	4359.72	96.54	4329.14	16.124
	100	4051.34	247.61	3716.93	145.934	4332.30	30.53	4329.14	160.161
	1000	4049.80	245.51	3702.22	1469.608	4332.61	31.85	4329.14	1608.239
F135	10	11878.16	1255.36	10550.60	84.126	14442.25	2305.01	11710.25	329.977
	100	11581.37	658.43	10134.69	841.055	13873.27	2261.45	10445.31	2686.973
	1000	11748.45	870.95	9790.50	8493.021	13890.62	2298.49	10364.63	27844.811

Table 6: Disaggregated results over the 17 problem instances obtained with the *Linkage-Approach* and the *Split-Approach* for 10, 100 and 1000 repetitions

		SV capacity								
		100			50			25		
		Transshipment times			Transshipment times			Transshipment times		
# t1 customers	depot location	0 mins.	15 mins.	30 mins.	0 mins.	15 mins.	30 mins.	0 mins.	15 mins.	30 mins.
12.5%	Standard	13.570	14.056	14.396	19.872	20.657	21.221	33.061	34.425	35.458
		(*) 11.374	12.095	12.466	18.630	19.326	20.321	31.204	36.501	40.741
		12.196	13.091	14.008	18.437	21.670	22.075	40.362	38.473	41.264
	(0,0)	16.401	16.739	17.346	24.995	25.541	26.623	41.522	43.592	45.036
		13.923	15.063	15.648	24.968	25.137	28.508	48.364	50.255	52.524
		18.060	17.711	19.089	23.407	24.309	25.481	39.729	45.999	48.183
	(-50,-50)	21.957	22.657	23.149	34.132	35.192	36.221	58.711	60.951	62.452
		18.866	19.272	20.433	32.592	34.323	36.237	63.695	60.517	69.134
		19.216	20.309	23.163	29.602	32.137	33.448	54.292	58.456	61.496
25%	Standard	13.758	14.313	14.615	19.249	20.006	21.173	30.260	31.611	34.327
		12.104	12.819	13.122	16.883	19.604	21.193	31.306	32.264	38.380
		11.973	14.209	14.006	17.467	19.727	21.825	36.599	37.518	39.576
	(0,0)	16.029	16.446	17.282	23.397	24.746	25.912	37.601	40.467	42.475
		13.820	14.380	15.583	21.367	23.248	25.025	36.603	46.683	48.090
		17.195	18.487	16.306	20.311	22.262	24.786	34.618	38.984	43.004
	(-50,-50)	22.716	23.548	24.276	33.374	34.703	35.682	54.110	56.550	59.492
		19.151	19.801	20.106	29.354	32.147	32.864	55.576	58.926	57.853
		18.992	19.915	23.668	26.323	29.257	32.000	47.524	52.245	56.891
50%	Standard	13.818	14.788	15.577	17.504	19.123	20.274	24.921	27.527	30.529
		13.539	14.378	15.820	16.856	18.951	22.401	23.671	27.097	33.463
		11.915	13.062	14.724	16.083	20.302	21.630	25.050	30.096	34.193
	(0,0)	17.090	17.671	17.886	21.632	22.582	23.997	30.945	33.704	36.848
		13.712	15.028	17.045	18.817	22.471	22.971	29.337	31.738	37.939
		14.259	16.699	18.362	17.376	19.975	22.896	24.120	31.749	37.670
	(-50,-50)	21.819	22.442	25.208	28.688	31.836	33.393	42.621	46.854	51.668
		19.326	21.515	23.356	23.748	29.917	31.795	38.148	45.442	51.694
		18.219	20.497	22.275	23.591	28.968	31.808	35.112	44.795	51.235

(*) HFVRP
Linkage
Split

Table 7: Sensitivity Analysis

Appendix B

For our analysis of the mathematical model we used a small test instance which is presented in Table 8. The parameter setting for the solver was $Q^{SV} = 20$, $Q^{LV} = 160$, $\mu = \nu = 0.1$, $Time^{UB} = 1000$, $F_{SV} = \{1\}$, $F_{LV} = \{1\}$ and the time limit for the solver was set to 3600 sec. We coded the mathematical formulation with the programming language Python (Version 2.7.6) and used the optimization software Gurobi Optimizer (Version 6.5.1: see[16]). The hardware configuration can be found in Section 6.

In Table 9 we present the results for the mathematical formulation as well as the heuristic results for a better comparison. In the first column we present the size of the instance (no. of customers) followed by the best found objective of the LP solver. In the third column we present the nodes explored by the branch-and-cut procedure along with the remaining optimality gap and the required cpu time. The last two columns show the best found objective values by our heuristic approaches.

If we have a look at Table 9 we can observe four interesting findings. First, with 6 customers (customers 1-6 in the instance) we are able to find an optimal solution by the mathematical formulation within 0.22 sec and also with the heuristic (Split) approach. Second, if we increase the number of customers to 9 we are able to find a feasible solution within the time limit of 3600 sec. but with a remaining optimality gap of 54.38%. Furthermore, our heuristic solutions are very close to the results gained by the mathematical formulation, with gap of less than 0.75% indicating the quality of those solutions. Third, if we further increase the number of customers to 16 we were still able to find a feasible solution but with a higher optimality gap of 65.47%. However, the obtained heuristic results of both approaches are far better than those of the mathematical formulation. Finally, we used the smallest instance out of our benchmark set (A32) and were not able to find a fesible solution within the given time limit of 3600 sec.

Overall, this short comparison helps to understand the complexity of the model as well as the value of the heuristics. Not only do those heuristics provide close to optimal results (as shown for the tiny instances), but they also provide results in a fraction of a second in those tested instances.

cust.	x-coord.	y-coord	service time	demand	type
0	0	0	0	0	0
1	60	60	10	10	1
2	80	60	10	10	2
3	100	60	10	10	1
4	60	80	10	10	2
5	80	80	10	10	2
6	100	80	10	10	2
7	60	100	10	10	1
8	80	100	10	10	2
9	100	100	10	10	1
10	120	60	10	10	2
11	120	80	10	10	2
13	60	120	10	10	2
14	80	120	10	10	2
15	100	120	10	10	2
16	120	120	10	10	2

Table 8: Instance for model analysis

	mathematical formulation				heuristics[a]	
cust.	objective	nodes[b]	gap	cpu[c]	Linkage	Split
6	5869	608	0.00%	0.22	5994	5869
9	7011	183127	54.38%	3600	7062	7034
16	12635	108957	65.47%	3600	10019	9865
A32[d]	-	95604	-	3600	8065	8337

[a] Heuristic results (objective) for comparison.

[b] Branch-and-cut nodes explored by the problem solver.

[c] CPU time in sec. with time limit of 3600 sec.

[d] Smallest instance set of [6].

Table 9: Computational results for model formulation

References

[1] Bräysy, O., Gendreau, M., 2005a. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Science 39, 104–118.

[2] Bräysy, O., Gendreau, M., 2005b. Vehicle routing problem with time windows, part ii: Metaheuristics. Transportation Science 39, 119–139.

[3] Chao, I.M., Golden, B., Wasil, E., 2016. A computational study of a new heuristic for the site-dependent vehicle routing problem. INFOR: Information Systems and Operational Research 37, 319–336.

[4] Chen, H.K., 2015. Issues for the linehaul-feeder vehicle routing problem with virtual depots and time windows. Journal of the Eastern Asia Society for Transportation Studies 11, 678–692.

[5] Chen, H.K., Chou, H.W., Hsu, C.Y., 2011a. The linehaul-feeder vehicle routing problem with virtual depots and time windows. Mathematical Problems in Engineering , 1–15.

[6] Chen, H.K., Chou, H.W., Hsueh, C.F., Ho, T.Y., 2011b. The linehaul-feeder vehicle routing problem with virtual depots. IEEE Transactions on Automation Science and Engineering 8, 694–704.

[7] Chen, H.K., Wang, H., 2012. A two-stage algorithm for the extended linehaul-feeder vehicle routing problem with time windows. International Journal of Shipping and Transport Logistics 4, 339–356.

[8] Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581.

[9] Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., 2002. A guide to vehicle routing heuristics. Journal of the Operational Research Society 53, 512–522.

[10] Cordeau, J.F., Laporte, G., 2016. A tabu search algorithm for the site dependent vehicle routing problem with time windows. INFOR: Information Systems and Operational Research 39, 292–298.

[11] Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. Management Science 6, 80–91.

[12] Drexl, M., 2012a. Applications of the vehicle routing problem with trailers and transshipments. European Journal of Operational Research .

[13] Drexl, M., 2012b. Synchronization in vehicle routing–a survey of vrps with multiple synchronization constraints. Transportation Science 46, 297–316.

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

[15] Golden, B., Assad, A., Levy, L., Gheysens, F., 1984. The fleet size and mix vehicle routing problem. Computers & Operations Research 11, 49–66.

[16] Gurobi Optimization, I., 2016. Gurobi optimizer reference manual. URL: `http://www.gurobi.com`.

[17] Kritikos, M.N., Ioannou, G., 2013. The heterogeneous fleet vehicle routing problem with overloads and time windows. International Journal of Production Economics 144, 68–75.

[18] Laporte, G., 2009. Fifty years of vehicle routing. Transportation Science 43, 408–416.

[19] Liu, F.H., Shen, S.Y., 1999. The fleet size and mix vehicle routing problem with time windows. The Journal of the Operational Research Society 50, 721.

[20] Loibnegger, M., 2016. Practical Analysis of the Linehaul-feeder Vehicle Routing Problem: Master Thesis. Graz.

[21] Penna, P.H.V., Subramanian, A., Ochi, L.S., 2013. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. Journal of Heuristics 19, 201–232.

[22] Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers & Operations Research 34, 2403–2435.

[23] Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002.

[24] Pureza, V., Morabito, R., Reimann, M., 2012. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the vrptw. European Journal of Operational Research 218, 636–647.

[25] Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35, 254–265.

[26] Subramanian, A., Penna, P.H.V., Uchoa, E., Ochi, L.S., 2012. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 221, 285–295.

[27] Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biometrics Bulletin 1, 80–83.