



HAL
open science

The Vehicle Routing Problem with FTL and LTL carriers

Gauthier Soleilhac, Fabien Lehuédé, Juliette Medina, Olivier Péton

► **To cite this version:**

Gauthier Soleilhac, Fabien Lehuédé, Juliette Medina, Olivier Péton. The Vehicle Routing Problem with FTL and LTL carriers. IMT Atlantique, Département Automatique, Productique et Informatique (DAPI), Campus de Nantes, 4 rue Alfred Kastler, 44300 Nantes Cedex 3. 2022. hal-03860741

HAL Id: hal-03860741

<https://hal.science/hal-03860741v1>

Submitted on 18 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Vehicle Routing Problem with FTL and LTL carriers

Gauthier Soleilhac^{a,b}, Fabien Lehuédé^b, Juliette Médina^a, Olivier Péton^b

^a*CRC Services, Rueil-Malmaison, France.*

^b*IMT Atlantique, Laboratoire de Sciences du Numérique de Nantes (LS2N, UMR CNRS 6004), Nantes, France.*

Abstract

We consider the distribution problem faced by a company that ships goods from a depot to multiple customers. This distribution is outsourced to multiple carriers offering different transportation rates and limited heterogeneous fleets of vehicles. Two shipping modes are used: in Less-Than-Truckload (LTL) mode, a shipment cost depends on its size and destination; in Full-Truckload (FTL) mode, routes should be designed for several fleets of heterogeneous vehicles, minimizing routing costs. This problem is formulated as a variant of a vehicle routing problem with time windows with private fleet and common carrier. To solve this problem, we use a matheuristic approach combining a Large neighborhood Search (LNS) metaheuristic with a Set Partitioning component. This algorithm embeds filtering strategies to efficiently integrate specific constraints in the LNS operators. We show that our method is competitive on less general problems benchmarks from the literature and we introduce new instances. We finally propose a case study arising from the French retail industry. We explore how the combination of FTL and LTL shipments can achieve substantial savings compared to pure FTL or LTL approaches.

Keywords: Vehicle routing, freight transportation, large neighborhood search

1. Introduction

Most industrial companies collaborate with several Logistics Service Providers (LSP) and carriers to ship their products from their industrial facilities to their customers or points of sale. They are traditionally called *shippers*. Designing a distribution network consists of defining all transportation flows starting at the production centers, while optimizing one or several criteria (cost, service level, environmental footprint). In practice, when a shipper collaborates with several carriers, the operational routing decisions are meant to simultaneously optimize the routes and allocate these routes to a set of carriers.

This paper studies the distribution policy of a shipper from one production facility or one warehouses to a set of customer facilities. We consider a single origin, hereafter called the depot, that frequently delivers to a set of customers spread over hundreds of kilometers. The shipper does not have its own fleet of vehicles for product distribution and the quantities to be shipped vary from one day to another. Thus, on each shipping day, all the transportation orders of the day are dispatched to several carriers, each having its own transportation rates, fleet of vehicles, and managerial constraints.

In this context, two very different shipping modes can be distinguished:

In *Full-Truckload* (FTL), the shipper books an entire vehicle to the carrier for a given predefined route. FTL routes all start at the depot and deliver several orders to one or several customers within their time windows. The shipper defines the route and the schedule of this vehicle while respecting a set of operational constraints that can differ from one carrier to another. The most classical constraints concern vehicle capacity or maximal load, and compatibility between vehicles and their cargo (e.g. refrigerated transport, specific containers, transportation of hazardous material, etc.). The carrier

Email addresses: gauthier.soleilhac@crc-services.com (Gauthier Soleilhac),
fabien.lehuede@imt-atlantique.fr (Fabien Lehuédé), juliette.medina@crc-services.com (Juliette Médina),
olivier.peton@imt-atlantique.fr (Olivier Péton)

may specify a maximal detour as an upper limit on out-of-route travel either as an absolute bound or as a ratio with respect to the original lane distance (Lindsey et al., 2013). Then, the legislation imposes constraints on the maximum duration of the routes. Finally, carriers and customers have their own constraints related to the number of stopovers in a route, or to the satisfaction of time windows and service times.

The FTL rates are either publicly available or have previously been established in a collaboration contract. The FTL rate proposed by a carrier may be based on a cost per kilometer but several other calculation methods are frequently used. For example, the cost may be determined by the farthest area reached by the itinerary plus a constant stopover cost for each intermediate stop.

All carriers have their own calculation rules that depend on many internal factors (vehicle amortization, drivers cost, fuel, tolls, etc.) as well as competitiveness factors (probability to find cargo for the return trip, presence of competing carriers, etc.). The cost of a given route also depends on the list of nodes visited and on the vehicle type.

Figure 1 shows an example of a zone-based fare. The cost is determined by the farthest area crossed (425 €). Three stopovers' costs (30 €) are added so the total cost of the route is 515 €.

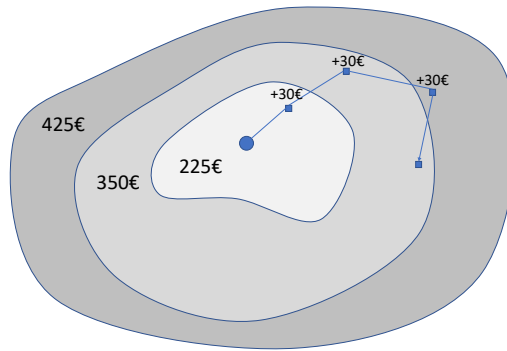


Figure 1: Example of an FTL rate determined by concentric areas around the depot and stopover costs

In *Less-Than-Truckload* (LTL) the carrier proposes a cost for each shipment according to the distance from the depot to its destination and to the quantity (weight, volume) to be delivered. Within this framework, the carrier organizes the optimization of the transports in its own network. As a result, all constraints, such as time windows, related to delivery are handled by the carrier. The LTL rate covers all shipment sizes but it is particularly attractive for small shipments. In return, the cost of shipping the first few units is higher than the average unit cost in FTL. In practice, the LTL rates are decreasing functions of the volume shipped. Figure 2 shows an example of such an LTL rate for one given destination.

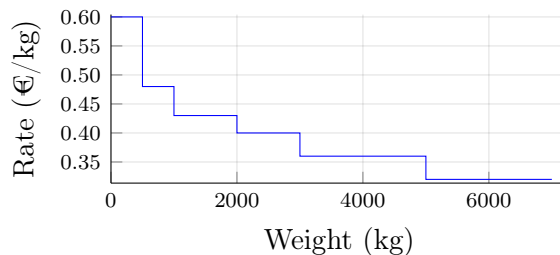


Figure 2: LTL rate example based on a shipping cost per kg which varies depending on the shipment weight.

Each shipping mode has advantages and disadvantages for the shipper. FTL shipping is more advantageous if the load is close to the capacity of a vehicle. On the other hand, it requires the transportation of large quantities which can be incompatible with the quantities ordered by customers. It also requires the shipper to have defined the routes to be taken and to manage the execution of these routes. Therefore, a certain expertise in vehicle routing and load planning is required. In contrast, LTL shipping allows the shipper to outsource the entire transport to its carriers but a full LTL strategy is often quite expensive.

The relationship between shippers and carriers establish a set of business constraints which are related to fleet management and practical transportation constraints. One of the main challenges for shippers is to find available trucks in increasingly tight transportation markets. The challenge for carriers is to ensure that a certain quantity of flow is entrusted to them in a repetitive manner and to optimally plan the necessary vehicles. Thus, both parties generally agree on a maximal number of trucks that can be guaranteed on every shipping day. For each carrier, a *carrier limit* defines the maximal number of vehicles that can be used by the shipper for each day of a given period. This carrier limit guarantees shippers that a number of trucks are reserved for them. It also helps carriers to control the number of vehicles chartered to their customers. It can also be used to model complex transportation rates. For example, when in a given fleet, the first vehicles are available at a low price and the following vehicles at a higher price. The OR literature generally considers the use of non-binding subcontracted resources (infinite capacity, no volume commitments) and a single fleet of homogeneous or heterogeneous vehicles (see e.g. [Chu \(2005\)](#); [Bolduc et al. \(2008\)](#); [Dabia et al. \(2019\)](#)). In this paper, we consider multiple fleets each one having its own limited number of vehicles. Another feature is that carriers own heterogeneous fleets of vehicles, i.e. fleets composed of different types of vehicles. Each type of vehicle is available in a limited quantity; it has with its own cost model and may be subject to its own specific constraints.

This work relies on a real case encountered by the CRC Services company which helps shippers to optimize their transport organization through organizational tools and optimization algorithms. Our challenge is to integrate these real-life constraints as well as realistic rates arising from real-life practices between shippers and carriers. The key contributions of this paper are the following:

- We introduce the *VRP-FLC* in which orders have to be served via FTL or LTL routes and where FTL shipping is subcontracted to several carriers with a limited heterogeneous fleet.
- We propose an LNS-SPP matheuristic that combines Large neighborhood Search (LNS) and a set-partitioning problem (SPP).
- Multiple route cost models and realistic constraints are integrated as an external code combining both incremental and static information to efficiently evaluate insertions with respect to their costs and feasibility.
- The LNS-SPP algorithm maintains a list of all vehicles compatible with a given route and embeds filtering strategies to efficiently integrate specific constraints in the LNS operators.
- We show that our method is competitive on less general problems benchmarks from the literature and introduce a new benchmark integrating multiple carriers and carrier limits.
- We explore how the combination of FTL and LTL shipments can achieve substantial savings compared to pure FTL or LTL approaches.

This paper is structured as follows: A review of the literature is proposed in Section 2, focusing on two features: private fleet and common carrier and limited heterogeneous fleet. Section 3 presents the problem settings and the corresponding mathematical model. Section 4 describes the proposed solution method. In Section 5, the proposed algorithm is evaluated on benchmark instances for very similar optimization problems. Then we present numerical results on new instances. Finally, Section 7 provides concluding remarks and suggests future research directions.

2. Literature review

The optimization problem considered in this paper belongs to the very broad category of Vehicle Routing Problems (VRP). It addresses two major features of shipper distribution problems: the presence of a limited heterogeneous fleet and the possibility of choosing between two shipping modes (FTL or LTL). In the literature, VRPs with the latter feature are called VRP with Private Fleet and Common Carrier (VRPPC). The heterogeneous fleet VRP and the VRPPC are detailed in Sections 2.1 and 2.2, respectively.

2.1. Heterogeneous Fleet Vehicle Routing Problems

There are two main variants of VRPs with heterogeneous fleets: the Fleet Size and Mix VRP (FSMVRP) and the Heterogeneous Fixed Fleet VRP (HFFVRP). The FSMVRP, introduced by [Golden et al. \(1984\)](#), considers several vehicle types which are available in an unlimited quantity. The objective is to define a set of routes and to assign them to vehicle types in order to lexicographically minimize the number of vehicles used and then the operating costs of the fleet. In the HFFVRP ([Taillard, 1999](#)), the number of vehicles of each type is limited. Since the composition of the fleet is already determined, the objective to be minimized is no longer the number of vehicles but the cost of operating the fleet to deliver the goods. The literature on vehicle routing problems with heterogeneous fleet is abundant. For a survey on FSMVRP and HFFVRP one can refer to [Koç et al. \(2016\)](#). Recently, several FSMVRP and HFFVRP variants have been solved in [Penna et al. \(2019\)](#) by a matheuristic which combines Iterated Local Search and a Set Partitioning component. Below, we focus on contributions that integrate time windows.

Many authors consider the presence of time windows in the FSMVRP. In contrast, very few papers consider HFFVRP with Time Windows (HFFVRPTW). [Paraskevopoulos et al. \(2008\)](#) propose a two-phase heuristic hybridizing Variable Neighborhood Search (VNS) and Tabu Search. The authors use the instances of [Liu and Shen \(1999\)](#), limiting the fleet according to the solutions reported by [Liu and Shen \(1999\)](#) on the FSMVRPTW. The objective is to minimize the total distribution time. [Koç et al. \(2015\)](#) propose a meta-heuristic called “Hybrid Evolutionary Algorithm”. The solutions are represented by a *giant tour* and the SPLIT algorithm proposed by [Prins \(2009\)](#) is enriched to assign a vehicle type to each route. The authors use the instances of [Paraskevopoulos et al. \(2008\)](#) and consider two objectives: minimization of the total en-route duration or minimization of the travelled distance. [Fachini and Armentano \(2020\)](#) propose two exact methods (Logic-Based Benders Decomposition and Branch-and-Check) based on a Benders decomposition to solve the HFFVRPTW where the total distance is minimized.

2.2. Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC)

The VRPPC, introduced by [Chu \(2005\)](#) is a generalization of the VRP in which orders can be delivered either by the private fleet or by an external carrier. Two factors can influence the choice of orders that are entrusted to the external carrier: the limited size of the private fleet and the potential savings of not using private vehicles for the delivery of remote customers. In our case, the *private fleet* is actually made up of different fleets offered by different FTL carriers and the external carrier corresponds to LTL rates. The VRPPC can be assimilated to a special case of the VRP with profits (see [Archetti et al. \(2014\)](#) for a survey): decision makers can choose either to deliver to their customers through FTL routes or not to deliver them by outsourcing the transportation (LTL routes).

Heuristic methods

[Chu \(2005\)](#) proposes an adaption of the savings heuristic of [Clarke and Wright \(1964\)](#) followed by a local search. This method is tested on five instances. [Bolduc et al. \(2007\)](#) propose a heuristic called Selection-Routing-Improvement to solve the VRPPC. It consists of selecting the requests to be served by the external carrier then constructing an initial solution before improving it with local search. [Bolduc et al. \(2008\)](#) show that the VRPPC can be reformulated as a VRP with Heterogeneous Fixed Fleet (HFFVRP) by replacing the external carrier by a dummy vehicle. The cost of an arc to a customer is equal to the cost of delivery by the external carrier. A meta-heuristic called Randomized Construction - Improvement - Perturbation (RIP) is proposed and applied on new instances with homogeneous or heterogeneous fleet. [Côté and Potvin \(2009\)](#) propose a Tabu Search (TS) algorithm to solve the VRPPC with a homogeneous fleet using two neighborhoods: 0-1 exchange and 1-1 exchange. [Potvin and Naud \(2011\)](#) and [Euchi \(2017\)](#) use neighborhoods based on ejection chains within a Tabu Search. [Liu et al. \(2010\)](#) study a variant of an arc-routing problem faced by a carrier. A first set of orders coming from shippers must be served by the private fleet of the carrier or outsourced to an external carrier at a given cost. A second set of orders that comes from other carriers can be served by the private fleet for a *compensative payment*. The problem is reformulated as an equivalent full-truckload node routing problem and solved with a memetic algorithm. Solutions are represented

as a giant tour and the split procedure is adapted to determine which requests are outsourced to an external carrier. Experiments are conducted on new instances featuring up to 400 requests.

Ceschia et al. (2011) study a rich VRPPC variant in which the heterogeneous fleet is composed of vehicles proposed by various carriers. In order to handle the different cost models proposed by the carriers, the authors suggest a Tabu Search approach in which the cost of the routes can be evaluated by calling external code. The authors identify and implement four different types of cost functions that they encountered in practice (cost based on the traveled distance or on the farthest point to the depot, costs depending on vehicle fill rate).

Vidal et al. (2015) propose a model unifying several variants of VRP with profits. A so-called exhaustive solution representation is used: customers are always placed on one of the private fleet routes, even if they are delivered by the external carrier. To determine which customers of a route are delivered by the external carrier, the authors solve a Resource-Constrained Shortest Path Problem (RCSPP). This solution representation is integrated with the UHGS meta-heuristic (Vidal et al. (2014)) and applied to VRPPC instances with homogeneous fleet. Gahm et al. (2017) investigate a variant of the VRPPC with several LTL carriers which offer discounts based on the total amount of cargo they are in charge of. It is also possible to charter vehicles from an external carrier to perform routes. The authors propose an ILP modeling and three variants of a Variable Neighborhood Search (VNS), which is tested on instances adapted from Bolduc et al. (2008). Bulhões et al. (2018) introduce the VRP with Service Level Constraints (VRPSL) in which service level constraints require delivery to part of the customers with the private fleet. The VRPPC can be described as a VRPSL with a zero service level constraint. A branch-and-price algorithm and a genetic algorithm based on the Unified Hybrid Genetic Search (UHGS) framework (Vidal et al. (2012)) are proposed.

Goeke et al. (2019) propose an LNS meta-heuristic with a decomposition strategy such that the removal and repair operators are only applied to a subproblem. A subproblem P_r is defined for each route r of the current solution. Each subproblem P_r only considers customers currently assigned to route r , customers of a random number of routes that are closest to r , and customers close to r that are currently assigned to the private carrier. A Variable Neighborhood Descent (VND) is also included and applied on promising solutions.

Exact Methods

Some exact methods have also been proposed to address the VRPPC. Goeke et al. (2019) use a branch-and-cut-and-price (BCP) approach. The master problem is a route-based Set Partitioning Problem (SPP). The pricing problem is an Elementary Shortest-Path Problem with Resource Constraints (ESPPRC) ; it is solved with a dynamic programming algorithm. This method can find optimal solutions for 8 of out 34 instances with homogeneous fleets and for 5 out of 34 instances with heterogeneous fleets.

Dabia et al. (2019) study a rich variant of the VRPPC with a limited heterogeneous fleet, time windows, and discounts offered by the external carrier. These discounts take the form of a piecewise linear function depending on the total quantity of goods entrusted to the external carrier. The authors propose an exact method relying on a route-based Set Partitioning Problem (SPP). Two formulations are proposed: with or without variables indicating the customers delivered by the external carrier. Both models are solved with a Branch-and-Cut-and-Price algorithm, which is evaluated on instances of Liu and Shen (1999), that were adapted to handle the case of subcontracting costs and a fleet limited to three units of each vehicle type.

Baller et al. (2020) study the VRP with Partial Outsourcing (VRPPO), in which delivery to any customer can be split between the private fleet and external carriers. Two route-based SPP formulations are proposed as well as a Branch-and-Cut-and-Price to solve them. The interest of partial delivery is evaluated on instances of Dabia et al. (2019), with or without partial outsourcing.

Table 1 summarizes the main references listed in this section and positions the present paper. Column 2 classifies the references according to the problem variant, with a distinction between problems with a homogeneous fleet or heterogeneous fleet. Columns 3 and 4 list the exact methods or heuristic methods used to solve these problems. Column 5 indicates when time windows are considered. All the papers cited in this table model subcontracted LTL shipments with the VRP with profit paradigm.

Reference	Problem variant	Exact methods	Heuristic methods	TW
<i>Homogeneous fleet</i>				
Côté and Potvin (2009)	VRPPC		TS	
Liu et al. (2010)	ARP-PC		MA	
Vidal et al. (2015)	VRPPC		UHGS	
Bulhões et al. (2018)	VRP-SL		HGS	
<i>Heterogeneous fleet</i>				
Chu (2005)	VRPPC		Savings + LS	
Bolduc et al. (2007)	VRPPC		SRI	
Bolduc et al. (2008)	VRPPC		RIP	
Potvin and Naud (2011)	VRPPC		TS	
Ceschia et al. (2011)	VRPPC [1]		TS	✓
Gahm et al. (2017)	VRPPCdR [2], [3]		RVND	
Euchi (2017)	VRPPC		TS	
Dabia et al. (2019)	VRPPC [2]	BPC		✓
Goeke et al. (2019)	VRPPC	BPC	LNS+VND	
Baller et al. (2020)	VRP-PO	BPC		✓
<i>This work</i>	VRP-FLC [1]		LNS-SPP	✓

LS = Local Search, TS = Tabu Search, MA = Memetic Algorithm, HGS = Hybrid Genetic Search, SRI = Selection, Routing and Improvement, RIP = Randomized construction - Improvement - Perturbation, (U)HGS = (Unified) Hybrid Genetic Search, RVND = Randomized Variable Neighborhood Descent, BPC = Branch and Price and Cut

[1] With multiple FTL Carriers, [2] With volume discounts, [3] With multiple LTL Carriers

Table 1: Papers related to the VRP with Private Fleet and Common Carrier (VRPPC).

One contribution of the problem introduced in this paper is the consideration of multiple FTL carriers. For LTL carriers we consider that a preliminary analysis allows to select which carrier is better for each order. Hence, the multiple LTL carriers case is not treated explicitly in the optimization. The new variant of the VRPPC problem introduced in this paper is called the *Vehicle Routing Problem with FTL and LTL carriers* (VRP-FLC).

3. Problem settings and mathematical formulation

The VRP-FLC considers a shipper and a set of customers to be delivered to. Let $\mathcal{N} = \{1, \dots, |\mathcal{N}|\}$ be the set of orders placed by these customers. An order is a set of products which must be delivered at once. It is characterized by the address of the delivery location, a size (number of pallets, and/or weight), a release date, and a time window in which the customer can be delivered to. Each order can be shipped either in FTL or in LTL mode. In LTL mode, the shipping cost c_n^{LTL} is given *a priori* for each order $n \in \mathcal{N}$.

When shipped in FTL mode, orders are delivered by a set of routes. A route $r \in \Omega$ starts from the depot and delivers a given subset of orders to their respective customers. All routes satisfy a number of constraints imposed by shippers, carriers, and the legislation (e.g., maximum number of stopovers, maximum duration or detours, access to customers' facilities). They also satisfy the time window constraints. Let us denote by Ω the set of all feasible routes. The total number of routes in the set Ω can be large and potentially lead to untractable MIP models. This is not problematic since our solution method does not explicitly handle the whole set Ω .

Let $\mathcal{K} = \{1, \dots, |\mathcal{K}|\}$ be the set of FTL carriers, each offering a heterogeneous fleet. For each carrier $k \in \mathcal{K}$, the set $\mathcal{V}^k = \{1, \dots, |\mathcal{V}^k|\}$ describes the different available vehicles types. Additionally, we define $\mathcal{V} = \bigcup_{k \in \mathcal{K}} \mathcal{V}^k$ as the whole set of vehicle types. We note k^v the carrier associated with the vehicle type $v \in \mathcal{V}$. All vehicle types are assumed to have the same traveling speed.

Due to the capacity constraints, and the various real-life and business constraints, the routes are not compatible with every vehicle type. We denote by Ω_v the subset of routes that can be performed by a vehicle of type $v \in \mathcal{V}$ and \mathcal{V}_r the *compatibility list* of a route r , i.e., the list of vehicle types that

can perform the route r . In addition, we denote by Ω^n the subset of routes that can deliver an order $n \in \mathcal{N}$.

The cost of route $r \in \Omega$ operated by a vehicle of type $v \in \mathcal{V}$ is denoted $c_{r,v}$. As in [Ceschia et al. \(2011\)](#), our proposed matheuristic method calls external code to compute this value.

We consider a limited *fleet size* for any carrier $k \in \mathcal{K}$ and any vehicle type $v \in \mathcal{V}$. The total number of available vehicles of type $v \in \mathcal{V}$ is denoted Q_v .

We also consider the *carrier limit*, defined as follows.

Definition 1 (Carrier limit). *For any carrier $k \in \mathcal{K}$, the total number of vehicles belonging to k used in the solution is upper bounded by the quantity ub_k , called the "carrier limit".*

From the definition it is straightforward that $ub_k \leq \sum_{v \in \mathcal{V}^k} Q_v$.

The VRP-FLC considers two types of decision variables. For any route $r \in \Omega$ and any vehicle type $v \in \mathcal{V}^r$, the binary variable $x_{r,v}$ takes the value 1 if a vehicle of type v is assigned to route r , and 0 otherwise. For any order $n \in \mathcal{N}$, the variable y_n takes the value 1 if the order n is delivered in LTL mode, and 0 otherwise.

The VRP-FLC is modeled by the following linear programming formulation with binary variables (1)–(6).

$$\min \quad z = \sum_{r \in \Omega} \sum_{v \in \mathcal{V}_r} c_{r,v} x_{r,v} + \sum_{n \in \mathcal{N}} c_n^{LTL} y_n \quad (1)$$

s.t.

$$\sum_{r \in \Omega^n} \sum_{v \in \mathcal{V}_r} x_{r,v} + y_n = 1 \quad \forall n \in \mathcal{N} \quad (2)$$

$$\sum_{r \in \Omega_v} x_{r,v} \leq Q_v \quad \forall v \in \mathcal{V} \quad (3)$$

$$\sum_{v \in \mathcal{V}^k} \sum_{r \in \Omega_v} x_{r,v} \leq ub_k \quad \forall k \in \mathcal{K} \quad (4)$$

$$x_{r,v} \in \{0, 1\} \quad \forall r \in \Omega, \forall v \in \mathcal{V}_r \quad (5)$$

$$y_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \quad (6)$$

This compact route-based model highlights the two main contributions (two types of routes and carrier limit). The objective function minimizes the set of transportation costs. The first sum represents the cost of FTL routes. The second sum represents the cost of LTL shipments.

Constraints (2) state that each order is delivered exactly once. This delivery can be done either in LTL or in FTL mode. Constraints (3) model the fleet size limit for each type of vehicle. Constraints (4) model the carrier limits for each carrier.

A solution to the *VRP-FLC* is a set of FTL routes starting from the depot and delivering a subset of orders, such that each route is assigned to a vehicle type in a way that the fleet size limits and the carrier limits are satisfied. Note that there is at least one feasible solution consisting of using LTL mode for all orders.

4. LNS-SPP algorithm

This section describes the LNS-SPP matheuristic and its adaptation to solve the VRP-FLC. LNS has been first introduced by [Shaw \(1998\)](#). It consists of exploring large neighborhoods by iteratively applying so-called *removal* and *repair* operators ([Ropke and Pisinger, 2006a](#)) also known as *ruin* and *recreate* heuristics ([Schrimpf et al., 2000](#)). The removal operators are heuristic algorithms that select a certain quantity of orders and simply remove them from the current solution. Their output is an *incomplete solution* and a *request bank*, where the removed orders are stored. The repair operators are heuristic algorithms that re-insert orders contained in the request bank into the incomplete solution with the objective to obtain a new feasible solution. The removal and repair operators are called until

a stopping criterion is met, e.g., a number of iterations or a time limit. The potential of LNS for solving a broad variety of VRPs was emphasized by Ropke and Pisinger (2006a,b); Pisinger and Ropke (2007), and in the survey (Pisinger and Ropke, 2019).

We highlight three major characteristics of our implementation. First, one source of inspiration is the recent ruin-and-recreate meta-heuristic of Christiaens and Vanden Berghe (2020), based on small removals and fast greedy insertion heuristics as repair operators. Thanks to these two factors, LNS can perform a very large number of iterations which somehow compensates the lack of a local search. Christiaens and Vanden Berghe’s algorithm competes with state-of-the-art algorithms on many VRPs including the fundamental and intensively-studied VRPTW. These operators are listed in Section 4.2.

Second, one specificity of the proposed algorithm is that the repair operators do not directly assign the routes to some vehicle type. Following the approach of Tellez et al. (2018), we maintain a list of compatible vehicle types for each route. In order to account for the carrier limit and the fleet size limit, the list of possible vehicle types for each route is filtered by extending a filtering algorithm proposed in Régim (1996). Once all orders have been reinserted, each route is assigned to a vehicle type with a heuristic assignment algorithm. The fleet management is detailed in Section 4.3.

Third, a Set Partitioning Problem (SPP) is solved at regular intervals during the LNS execution in order to recombine routes that are produced at different iterations. The use of SPP as a post-optimization technique is reported in (Mancini, 2017; Gschwind and Drexl, 2019). The hybridization with heuristics is reported in many papers including (Prescott-Gagnon et al., 2009; Groër et al., 2011; Mendoza and Villegas, 2013; Subramanian et al., 2013; Parragh and Schmid, 2013; Yıldırım and Çatay, 2015; Grangier et al., 2017; Tellez et al., 2018; Dumez et al., 2021).

4.1. Detail of the LNS-SPP algorithm

The proposed LNS-SPP algorithm is described in Algorithm 1.

Algorithm 1 : LNS-SPP algorithm

```

Parameters :  $\Sigma^-, \Sigma^+$  // Sets of ruin and repair operators
Input :  $s$  // Initial solution
1:  $\Omega \leftarrow routes(s)$  // Pool of routes
2:  $s_{best} \leftarrow s$ 
3: while termination criterion is not met do
4:    $\sigma^- \leftarrow selectRemovalOperator(\Sigma^-)$ 
5:    $\sigma^+ \leftarrow selectRepairOperator(\Sigma^+)$ 
6:    $\phi \leftarrow U(\phi_{min}, \phi_{max})$  // Randomly select a destruction size
7:    $s' \leftarrow \sigma^+(\sigma^-(s, \phi))$ 
8:    $lb \leftarrow lowerBound(s')$ 
9:   if AcceptanceCriterion( $lb, s_{best}$ ) then
10:      $s' \leftarrow AssignVehiclesToRoutes(s')$ 
11:     if AcceptanceCriterion( $z(s'), s_{best}$ ) then
12:        $\Omega \leftarrow \Omega \cup routes(s')$ 
13:        $s \leftarrow s'$ 
14:       if  $z(s) < z(s_{best})$  then
15:          $s_{best} \leftarrow s$ 
16:       end
17:     end
18:   end
19:   if  $|\Omega| \geq \eta$  then
20:      $s_{best} \leftarrow SPP(\Omega)$ 
21:      $\Omega \leftarrow routes(s_{best})$ 
22:      $s \leftarrow s_{best}$ 
23:   end
24: end
25: return  $s_{best}$ 

```

An initial solution s is obtained by using the *Farthest Insertion* repair operator (see Section 4.2.2) from an empty solution. The routes of the initial solution are saved in the pool Ω . (line 1). The best solution s_{best} is initialized as a copy of the initial solution (line 2).

The main loop (lines 3–24) is repeated until a termination criterion is satisfied. The termination criterion can be based on computation time, a maximum number of iterations, or a maximum number of iterations without improvement of the best solution. At each iteration, a removal and a repair operator are randomly chosen with a biased roulette wheel (lines 4 and 5). A destruction size ϕ is also randomly selected according to a uniform distribution in the interval $[\phi_{min}, \phi_{max}]$ (line 6). Then, the removal and the repair operator are applied to the current solution and the result is saved in a temporary copy s' (line 7). During this phase, all insertions in a route r are evaluated using the cheapest available vehicle that is compatible with r after insertion.

Before calling the assignment heuristic, a naive lower bound lb is computed in line 8. It is calculated by considering an idealistic assignment of routes to vehicles where each route is assigned to its cheapest compatible vehicle type, regardless of the fleet size limit and carrier limit.

We then check whether this lower bound passes the acceptance criterion. The acceptance criterion in lines 9 and 11 is based on the record-to-record principle (Dueck, 1993). If the objective function $z(s')$ of the new solution s' is not $\mathcal{T}\%$ worse than that of s_{best} , then s' is accepted as the new current solution. As suggested in Dumez et al. (2021), \mathcal{T} is periodically updated in order to keep the ratio of accepted solutions between $\alpha_{min}\%$ and α_{max} . If the ratio of accepted solutions in the last μ iterations is lower than $\alpha_{min}\%$, \mathcal{T} is multiplied by 1.5. If it is larger than $\alpha_{max}\%$, \mathcal{T} is divided by 1.5.

If s' passes this preliminary test, then all routes in s' are assigned to a vehicle type so that all VRP-FLC constraints are satisfied. The assignment heuristic will be detailed in Section 4.3. If this full solution passes the acceptance criterion, then routes in s' are appended to Ω (line 12) and the values of the current and best solutions are updated (lines 13–16).

The lines 19–23 correspond to the Set Partitioning component. When the size of the pool Ω reaches a predetermined number of routes η , an SPP instance containing all routes of the pool is built. Instead of solving an SPP formulation, we solve a Set Covering Problem (SCP) formulation. The solution obtained is repaired if a customer is served more than once. Yıldırım and Çatay (2015) showed that solving an SCP instead of an SPP slightly shortens solving time. Additionally, an exact reparation is rarely needed and a greedy procedure generally finds the optimal reparation. The SPP is solved by a MIP solver with a time limit of 30 seconds. Afterwards, the pool is then cleared and re-initialized with the routes of the best solution found.

4.2. LNS operators

The efficiency of LNS algorithms relies on a set of well-chosen removal and repair operators. We implemented twelve removal operators and ten repair operators. After preliminary experiments on a representative set of instances, this set was reduced to the six removal operators presented in Section 4.2.1 and the five repair operators presented in Section 4.2.2. The selection of these operators is based on experiments with the *irace* package (López-Ibáñez et al. (2016)), allowing the weight of each operator to take an integer value between zero and two. Only operators with a positive weight were kept. A second round of tuning with *irace* was used to set the weight of each operator. The corresponding values, between 1 and 4, are reported in Table 2. At each iteration, these weights are used in a biased roulette to determine which operator will be used.

4.2.1. Removal operators

Removal operators are parametrized with a destruction size ϕ drawn in the interval $[\phi_{min}, \phi_{max}] \subset [0, 1]$ at each iteration. For all removal operators listed below except the string removal and random removal, the orders are ranked according to some criterion and the $\lceil \phi \times |\mathcal{N}| \rceil$ first orders are removed from the current solution and placed into a *request bank*.

The removal operators used in line 4 of Algorithm 1 are the following:

- Random Removal: random removal of orders;
- (Split) String Removal (Christiaens and Vanden Berghe (2020)): This operator selects several small sequences of order in different close routes. For each selected sequence, either the sequence is removed from the solution or it is kept (split case) and the other customers of the route are removed. The method was implemented as described by Christiaens and Vanden Berghe (2020)

with the suggested parameters except for \bar{c} , the average number of customers to remove, which is set to $\lceil \phi \times |\mathcal{N}| \rceil$;

- Least LTL Cost Removal : removal of orders having the cheapest LTL cost;
- Distance Related Removal (Shaw (1998)): removal of a randomly selected order and its nearest orders according to the traveled distance;
- Largest Request Removal: removal of orders with the largest shipment size;
- Greatest Waiting Time Removal: removal of orders which generate the greatest waiting time, i.e., that are served the most ahead of their time window.

4.2.2. Repair operators

Since all orders can be shipped in LTL mode, it is always possible (and easy) to rebuild feasible solutions. We use fast repair operators taken from or inspired by those proposed by Christiaens and Vanden Berghe (2020). They consist of sorting the request bank once at the beginning of the reconstruction phase, then, according to this sorting, all orders in the request bank are sequentially inserted in the solution at their best possible position. Another advantage of using these heuristics is that they necessitate an order of magnitude fewer evaluations of routing costs than traditional heuristics such as greedy or regret-based insertions.

To further reduce the time spent on calculating routing costs, we designed the *cost evaluation* and the *route feasibility* functions to accept a list of parameters that are incrementally calculated. The parameters that need to be calculated depend on the types of costs for the instance at hand. For the costs found in instances of the literature, only the route’s load, length, and duration suffice. Other information, such as product types or the list of nodes visited, is computed if needed for more complex cost functions.

The repair operators used in line 5 of Algorithm 1 are the following:

- Random: the sequence of orders insertion is random;
- Request Size: the orders with larger shipment size are inserted first;
- Farthest: the orders with the farthest delivery location from the depot are inserted first;
- Nearest: the orders with the nearest delivery location are inserted first;
- Time Window Width: the orders with the tightest time windows are inserted first.

As in Christiaens and Vanden Berghe (2020), when evaluating the cost of inserting an order at different positions in a route, each position has a probability p^{blink} not to be considered.

In the repair step (line 7 of Algorithm 1), inserting an order in an incomplete solution requires evaluating the cost increase generated by the use of each vehicle type at each insertion position. Checking the feasibility of an order insertion in an existing route $r \in \Omega$ amounts to check that, after insertion, (i) the route is still valid with respect to time windows, (ii) the list of vehicles types compatible with r is not empty and, (iii) constraints (3)–(4) hold.

The feasibility of an order insertion with respect to time windows can be checked in constant time by using the forward time slacks (Savelsbergh (1985)). Note that all vehicles travel at the same speed, thus the satisfaction of time windows does not depend on the vehicle that performs the tour.

Then, we check if the new solution is still compatible with the vehicle fleet. The calculation of the cost of a route is based on the cheapest compatible vehicle. After an order insertion, this calculation includes a possible change of vehicle type. This is why Algorithm 1 needs to carefully manage vehicle types, as explained in Section 4.3.

4.3. Management of carriers with limited fleet

This section details the feasibility checking of order insertion with respect to the fleet and to the carrier limit constraints. Example 1 illustrates the case where some route - vehicle type assignment is rendered impossible by the combination of the fleet size and the carrier limit.

Example 1. Let us consider two carriers k_1 and k_2 . Carrier k_1 has a fleet of 10 vehicles: 9 of type A and 1 of type B. Carrier k_2 has a fleet of 4 vehicles: 2 of type C and 2 of type D but has only 2 drivers available; therefore only 2 vehicles can be used.

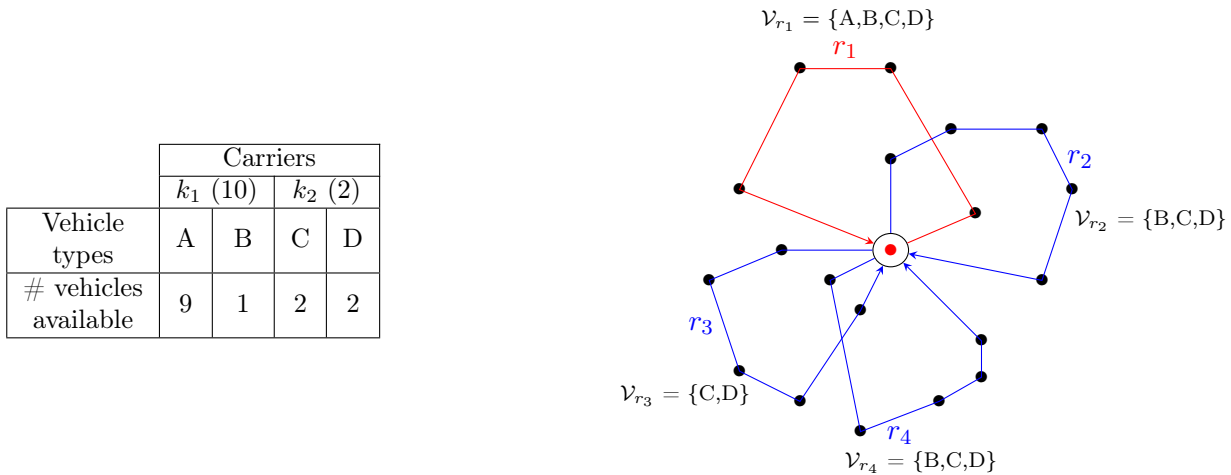


Figure 3: Example of vehicle type filtering during the repair phase

Let us consider the incomplete solution depicted in Figure 3, with four routes denoted r_1, r_2, r_3 , and r_4 . The unserved orders, placed in the request bank, are not represented. Route r_1 can be executed by all vehicle types ($\mathcal{V}_{r_1} = \{A, B, C, D\}$). The other routes have a limited set of compatible vehicles: $\mathcal{V}_{r_2} = \{B, C, D\}$, $\mathcal{V}_{r_3} = \{C, D\}$ and $\mathcal{V}_{r_4} = \{B, C, D\}$. Since routes r_2, r_3 , and r_4 are compatible with vehicle types B, C, and D only, and that at most three vehicles of these types can be used, these routes will take all available vehicles B, C, and D. Consequentially even though route r_1 is compatible with all vehicle types, it can only be performed by a vehicle of type A. Therefore, route r_1 must be assigned to a vehicle of type A.

In Section 4.3.1, we describe the filtering algorithm used to define a restricted set of possible vehicle types for each route. In Section 4.3.2, we detail the heuristic algorithm used to assign one vehicle type for each route.

4.3.1. Filtering algorithm

As illustrated in the previous example, some filtering of the routes compatibility lists can be deduced for each route, depending on the other routes compatibility lists and from the fleet sizes and carrier limits. The objective of this filtering is twofold: first, it allows the prevention of insertions that would result in an infeasible solution (in the example, performing an insertion that would remove A from \mathcal{V}_{r_1}) or to early detect an infeasible solution (i.e., empty compatibility list). Second, it supports a better estimation when calculating the cost lower bound in the algorithm.

During the repair phase of Algorithm 1, let us consider an incomplete solution s defined by a set of routes $\Omega(s)$ and a request bank containing a set of orders that are still not served.

Inserting an order in a route $r \in \Omega(s)$ is likely to restrict its compatibility list \mathcal{V}_r . Assigning a vehicle type to route r can also restrict the list of vehicle types compatible with all other routes $r' \neq r$. We introduce the notation \mathcal{V}_r^s to define the *filtered compatibility list* of a route r in solution s , with $\mathcal{V}_r^s \subseteq \mathcal{V}_r$. This list is obtained by a filtering algorithm that is adapted from Régim (1996) in order to jointly constrain the number of vehicles of each type and the maximum number of vehicles chartered to each carrier. Régim (1996) proposed a filtering algorithm for the *global cardinality* constraint which enforces that a minimum and maximum number of occurrences of a set of values can be assigned to a

list of variables. Our filtering algorithm relies on a layered graph $G = (X, E)$, where X contains three layers of vertices representing the carriers $t \in T$, the vehicle types $v \in \mathcal{V}$ and the routes $r \in \Omega(s)$, respectively. It also includes a source node \mathbf{o} and a sink node \mathbf{o}' . The arc set E contains an arc (\mathbf{o}, k) of capacity ub_k for each carrier $k \in \mathcal{K}$, an arc (k, v) of capacity \mathcal{Q}_v for each vehicle type v of a carrier k , and arcs (v, r) of capacity 1 between routes and compatible vehicle types in the solution $((v, r) \in E \iff v \in \mathcal{V}_r^s)$. The nodes corresponding to the routes are connected to the sink node by arcs (r, \mathbf{o}') of capacity 1.

Example 2. The filtering graph corresponding to the case presented in Example 1 is represented by Figure 4.

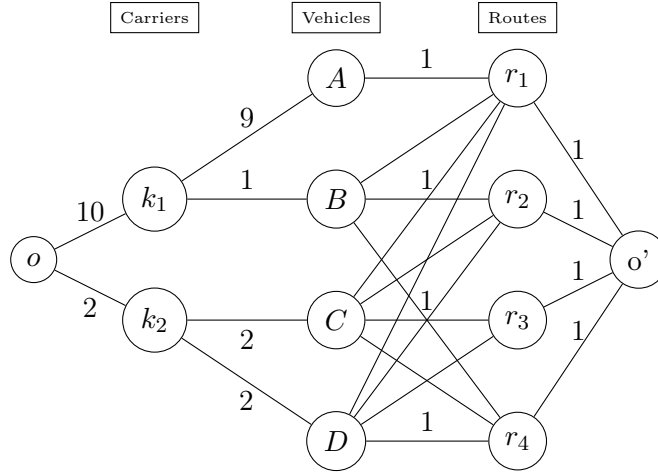


Figure 4: Graph representing the possible assignments of routes to vehicle types

Let \mathbf{f} be a maximal flow from source to sink in graph G . Because there is one arc with a capacity of 1 per route that is connected to the sink \mathbf{o}' , the value of \mathbf{f} is less than or equal to the number of routes. If this value is equal to the number of routes, then, by construction, each route is associated with one single vehicle type. Therefore, the arcs used in the flow between the vehicles and routes vertices define a feasible assignment of routes to vehicle types.

Indeed, from left to right in the example, the first and second layers of arcs guarantee respectively that the carrier limit and fleet size constraints are verified. The third layer guarantees that each route is assigned to a vehicle from its compatibility list. Given a maximal flow \mathbf{f} , we build a residual graph $\mathcal{R}(\mathbf{f})$. It is a directed graph for which the vertex set is the same as that of graph G and where the arcs capacities are the quantities by which the flow \mathbf{f} can be either increased or decreased.

Régin (1996) shows the following property:

Property 1. *If the flow on an arc connecting two vertices is null and if these two vertices do not belong to the same strongly connected component in $\mathcal{R}(\mathbf{f})$, then there cannot exist another maximal flow with a non-zero flow on this arc.*

Given Property 1, if for a vehicle type $k \in \mathcal{K}$ and a route $r \in \Omega(s)$ there exists a maximal flow \mathbf{f} with $\mathbf{f}(k, u) = 0$ and if k and u don't belong to the same strongly connected component of $\mathcal{R}(\mathbf{f})$, then we cannot find an assignment of all the routes to a vehicle type if route r is assigned to vehicle type k . The filtering method consists of computing a maximal flow, its residual graph, and identifying the strongly connected components on the residual graph. Then, for each route $r \in \Omega(s)$ and each vehicle $v \in \mathcal{V}_r$, we check if Property 1 is verified on arc (v, r) . If it is the case, v is removed from the route compatibility list \mathcal{V}_r^s .

Figure 5 shows an example of a maximal flow that corresponds to the assignment of a vehicle of type A (resp. B, C, D) to routes r_1 (resp. r_2, r_3, r_4).

Figure 6 shows the residual graph of this flow. It has four strongly connected components - $\{\mathbf{o}, k_1, A\}$, $\{k_2, B, C, D, r_2, r_3, r_4\}$, $\{r_1\}$, and $\{\mathbf{o}'\}$, which are indicated by different colors on the graph. Since r_1 and B (resp. C and D) do not belong to the same strongly connected component in $\mathcal{R}(\mathbf{f})$ and the flow on arcs connecting them is null in \mathbf{f} , r_1 can only be assigned to vehicle type A.

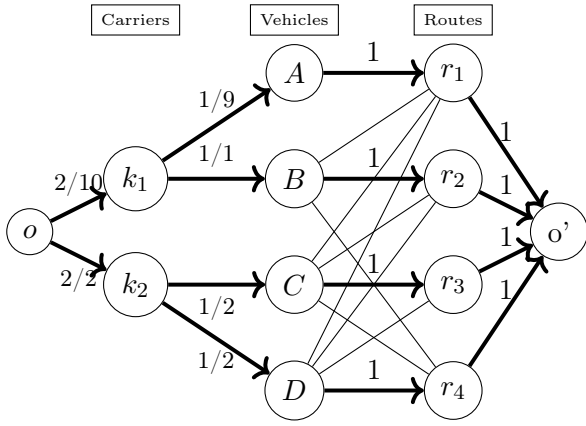


Figure 5: Maximal flow corresponding to a feasible assignment

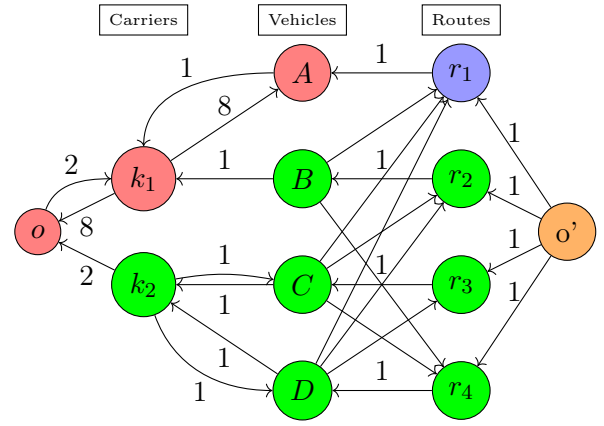


Figure 6: Residual graph and strongly connected components of a maximum flow

4.3.2. Assignment of routes to vehicle types

After inserting all orders from the request bank, line 10 of Algorithm 1 assigns all routes in the solution s' to a vehicle type. This amounts to solving the *VRP-FLC* (1)-(6) with Ω restricted to the set of routes $\Omega(s')$.

Before looking for an assignment, we check that a lower bound on the value of the assignment verifies the acceptance criterion. This bound consists of solving the *VRP-FLC* without considering the fleet size and carrier limit constraints (3) and (4). In such a solution, each route is automatically assigned to its cheapest compatible vehicle type. Then, routes are selected if and only if their cost does not exceed the sum of the LTL delivery cost of its orders.

If this bound verifies the acceptance criterion, then we solve the generalized assignment problem with the MTHG regret-based heuristic (Martello, 1990) to obtain a complete feasible solution to the *VRP-FLC*. We define the regret associated to a route as the difference in the route cost between the two cheapest compatible vehicles. If a route can be assigned to a single vehicle type, its regret is defined as the difference between the cost of outsourcing all of its orders to the LTL carrier minus the cost of the route with its vehicle type. Iteratively, we assign the route with the highest regret to its cheapest vehicle type. When a vehicle type cannot be assigned anymore because of its limited availability or because of the limits on the carrier's fleet, the regrets for routes that relied on this vehicle type are recalculated.

5. Numerical experiments

This section presents the computational results conducted on a cluster running with Ubuntu 20.04.2 LTS and using Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz. The LNS-SPP algorithm is coded as a single-threaded program in Julia. IBM Ilog CPLEX 20.1.0 is used as a MIP solver, with standard tuning and parameter `CXPPARAM_Threads` set to value 1. Algorithm 1 contains a number of parameters that were tuned with the irace package (López-Ibáñez et al., 2016) on a representative set of instances. Table 2 lists the value of these parameters.

Since there are no instances for the *VRP-FLC* problem in the literature, we evaluate the efficiency of our approach on instances of closely related problems. Section 5.1 relates the computational experiments on the Heterogeneous Fixed Fleet Vehicle Routing Problem with Times Windows (HFF-VRP-TW). Section 5.2 presents results on the HFF-VRP-TW with Private fleet and Common carrier (HFF-VRP-TW-PC). Section 5.3 introduces some new instances generated for the *VRP-FLC*. These instances are solved by the LNS-SPP algorithm and used to provide managerial insights on the problem.

5.1. Results on HFF-VRP-TW benchmark instances

Paraskevopoulos et al. (2008) built HFF-VRP-TW instances from a subset of the FSM-VRP-TW instances proposed by Liu and Shen (1999). They consider 4 instances in each category of Liu and

	Parameter	Value
General parameters	initial acceptance threshold	1%
	α_{min}	4%
	α_{max}	14%
	ϕ_{min}	10%
	ϕ_{max}	20%
	p^{blink}	0.01
	μ	300
Set partitioning	η	3000
Weight of removal operators	Random Removal	2
	(Split) String Removal	4
	Least LTL Cost Removal	1
	Distance Related Removal	1
	Largest Request Removal	1
	Greatest Waiting Time Removal	2
Weight of repair operators	Random	3
	Request Size	1
	Farthest	2
	Nearest	2
	Time Window Width	2

Table 2: List of LNS-SPP parameters

Shen’s instances. In these instances, the fleet size is set equal to that found in the best known solutions of the FSM-VRP-TW. The vehicle fixed costs are high (type A). The objective to be minimized is a sum of vehicles fixed costs and total travel and waiting time. Service time, which is constant for any solution is not counted in the objective. Using the same instances, [Koç et al. \(2015\)](#) use a Hybrid Evolutionary Algorithm (HEA) and present results for two objective functions: time minimization and distance minimization.

In order to assess the impact of the filtering procedure described in Section 4.3.1, we ran two sets of experiments using these instances. In the first one, vehicle compatibility lists are filtered after an insertion is performed in a route that makes it infeasible for at least one vehicle type. In the second one, vehicle compatibility lists are not filtered and always contain all the vehicle types that are compatible with the route. On average, the solutions obtained using the filtering procedure are 0.40% better than those obtained without filtering. If we compare the best solutions obtained out of 10 runs, the filtering procedure improves the results by 0.10% on average. The results presented in the remainder of this paper integrate the filtering procedure.

Table 3 compares our results with those obtained by [Koç et al. \(2015\)](#) for distance minimization.

The first column contains the instance names. Then, for each method, the columns entitled **Mix** and z^{best} describe the number of vehicles of each type used in the best found solution and the value of the objective function, respectively. The column entitled **gap (%)** gives the relative gap between the proposed LNS-SPP algorithm and the best known solutions for each instance.

According to these experiments, we find that with smaller solving times, LNS-SPP outperforms HEA on instances with short routes (instance groups R1, C1 and RC1) and that it is very close to HEA on instances with long routes. The results were improved in 9 out of 24 instances ; identical solutions are found for 8 instances and slightly worse results were obtained for 7 instances. Most of the improvements are obtained through a modification of the vehicle mix. Besides, for all instances where HEA outperforms LNS-SPP, both algorithms find the same vehicle mix.

Table 4 compares our results with those obtained by [Paraskevopoulos et al. \(2008\)](#) and by [Koç et al. \(2015\)](#) for time minimization. Here again, LNS-SPP obtains competitive results with respect to [Paraskevopoulos et al. \(2008\)](#) and [Koç et al. \(2015\)](#). The best known solutions were improved in 14 out of 24 instances ; identical solutions are found for 5 instances and slightly worse results were obtained for 5 instances.

Instance	HEA Koç et al. (2015)			LNS-SPP		
	Mix	z^{best}	Time (min)	Mix	z^{best}	gap (%)
R101A	$B^{10}C^{11}D^1$	4355.41	5.19	$A^1B^{11}C^{10}D^1$	4333.36	-0.51
R102A	$B^4C^{13}D^2$	4356.44	6.24	$A^1B^4C^{14}D^1$	4187.93	-3.87
R103A	B^6C^{15}	4080.16	6.57	B^7C^{14}	4038.72	-1.02
R104A	B^7C^{14}	3954.72	5.89	B^7C^{14}	3952.75	-2.58
C101A	B^{10}	8828.94	4.25	B^{10}	8828.94	0.0
C102A	A^{19}	7080.17	3.97	A^{19}	7080.17	0.14
C103A	A^{19}	7079.21	3.99	A^{19}	7079.21	0.0
C104A	A^{19}	7075.06	2.98	A^{19}	7075.06	0.0
RC101A	$A^4B^7C^7$	5162.28	6.41	$A^4B^7C^7$	5162.25	0.0
RC102A	$A^2B^6C^8$	5018.05	5.24	$A^4B^5C^8$	4986.61	-0.63
RC103A	$A^{10}B^2C^8$	4926.55	4.39	$A^{10}B^2C^8$	4897.08	-0.60
RC104A	$A^2B^{13}C^3D^1$	4995.91	4.88	$A^1B^{13}C^3D^1$	4950.62	-0.91
R201A	A^5	3448.76	6.74	A^5	3463.27	0.42
R202A	A^5	3308.16	8.13	A^5	3314.89	0.20
R203A	A^4B^1	3382.39	7.49	A^4B^1	3392.79	0.31
R204A	A^5	3018.14	5.47	A^5	3022.23	0.14
C201A	A^4B^1	6082.38	4.21	A^4B^1	6082.38	0.0
C202A	A^1C^3	7618.62	3.69	A^1C^3	7618.62	0.0
C203A	C^2D^1	7303.37	3.67	C^2D^1	7303.37	0.0
C204A	A^5	5677.66	5.11	A^5	5677.66	0.0
RC201A	C^1E^3	5344.47	6.72	C^1E^3	5294.8	-0.93
RC202A	$A^1C^1D^1E^2$	4856.02	6.48	$A^1C^1D^1E^2$	4865.98	0.21
RC203A	$A^1B^1C^5$	4246.25	6.93	$A^1B^1C^5$	4212.44	-0.80
RC204A	$A^{14}B^2$	4195.32	6.17	$A^{14}B^2$	4198.68	0.08
Average		5224.77	5.45		5209.16	-0.43
Runs		10			10	
Processor		Xe 2.6 GHz			Xe Gold 6230 2.10GHz	

Table 3: Results on HFF-VRP-TW instances, minimization of the distance traveled, 2 minutes per run

Instance	ReVNTS		HEA		LNS-SPP		
	Paraskevopoulos et al. (2008)	z	Koç et al. (2015)	z^{best}	Mix	z^{best}	gap (%)
R101A	$B^{10}C^{11}D^1$	4583.99	$B^{10}C^{11}D^1$	4588.76	A^1, B^9, C^{11}, D^1	4626.66	0.931
R102A	$B^3C^{14}D^2$	4420.680	$A^1B^4C^{13}D^2$	4376.54	A^1, B^4, C^{14}, D^1	4394.16	0.403
R103A	B^6C^{15}	4195.05	B^6C^{15}	4201.71	B^7, C^{14}	4149.94	-1.075
R104A	B^8C^{14}	4065.52	B^9C^{13}	4027.69	B^7, C^{14}	3970.2	-1.427
C101A	B^{10}	8828.93	B^{10}	8828.93	B^{10}	8828.94	0.0
C102A	A^{19}	7137.79	A^{19}	7153.13	A^{19}	7119.35	-0.258
C103A	A^{19}	7143.88	A^{19}	7122.57	A^{19}	7102.86	-0.277
C104A	A^{19}	7104.96	A^{19}	7083.74	A^{19}	7081.51	-0.031
RC101A	$A^4B^7C^7$	5279.92	$A^4B^7C^7$	5266.36	A^4, B^7, C^7	5221.81	-0.846
RC102A	$A^4B^5C^8$	5149.95	$A^4B^5C^8$	5099.55	A^4, B^5, C^8	5043.57	-1.098
RC103A	$A^{10}B^2C^8$	5002.41	$A^{10}B^2C^8$	4991.29	A^{10}, B^2, C^8	4924.08	-1.347
RC104A	$A^2B^{13}C^3D^1$	5024.25	$A^2B^{13}C^3D^1$	5016.97	A^1, B^{13}, C^3, D^1	4952.61	-1.283
R201A	A^5	3779.12	A^5	3782.49	A^5	3839.82	1.606
R202A	A^5	3578.91	A^5	3583.92	A^5	3556.41	-0.629
R203A	A^4B^1	3582.54	A^4B^1	3553.92	A^4, B^1	3550.46	-0.097
R204A	A^5	3143.68	A^5	3081.80	A^5	3079.58	-0.072
C201A	A^4B^1	6140.64	A^4B^1	6140.64	A^4, B^1	6140.64	0.0
C202A	A^1C^3	7752.88	A^1C^3	7623.96	A^1, C^3	7623.96	0.0
C203A	C^2D^1	7303.37	C^2D^1	7303.37	C^2, D^1	7303.37	0.0
C204A	A^5	5721.09	A^5	5680.46	A^5	5680.46	0.0
RC201A	C^1E^3	5523.15	C^1E^3	5534.59	C^1, E^3	5536.76	0.246
RC202A	$A^1C^1D^1E^2$	5132.08	$A^1C^1D^1E^2$	5150.23	A^1, C^1, D^1, E^2	5189.71	1.123
RC203A	$A^1B^1C^5$	4508.27	$A^1B^1C^5$	4471.92	A^1, B^1, C^5	4466.48	-0.122
RC204A	$A^{14}B^2$	4252.87	$A^{14}B^2$	4241.83	A^{14}, B^2	4229.25	-0.297
Average		5348.16		5326.74		5317.19	-0.19
Average time (min)		20		5.61		2	
Runs		1		10		10	
Processor		PIV 1.5GHz		Xe 2.6GHz		Xe Gold 6230 2.10GHz	

Table 4: Results on HFF-VRP-TW instances, minimization of the time, 2 minutes per run

5.2. Results on HFF-VRP-TW-PC benchmark instances

We benchmarked the results of the LNS-SPP algorithm against the most recent works on the HFF-VRP-TW-PC. [Baller et al. \(2020\)](#) derive instances from those of [Dabia et al. \(2019\)](#) and propose two categories of instances with high and low outsourcing costs, respectively. As in the original instances, they were adapted from Solomon’s instances ([Solomon, 1987](#)). They feature different fleets as defined by [Liu et al. \(2010\)](#) but the fleet size of each type of vehicle is set to 3. The original instance have 100 customers. Smaller instances built from the first 25 or 50 customers of each instance are also considered.

Table 5 compares our results to the exact results obtained by [Baller et al. \(2020\)](#) on the VRPPC. Each line aggregates the results for one category of instances. The letters A, B, and C represent instances with high, medium, and low fixed vehicle cost, respectively. The values 25, 50, and 100 represent the number of customers. Each category of instances is made up of 56 instances.

Instances	BCP			LNS-SPP				
	#	z_{avg}^*	t_{avg} (s)	# opt	z_{avg}^{best}	$gap_{avg}^{best}(\%)$	$gap_{avg}^{avg}(\%)$	time (s)
<i>High Outsourcing Costs</i>								
A/25	22	1233.03	10	22	1233.03	0	0.06	30
A/50	15	2402.88	185	14	2402.81	-0.002	0.41	60
A/100	5	5050.96	149	3	5051.52	0.01	0.14	300
B/25	34	586.36	9	34	586.36	0	0.26	30
B/50	25	1115.96	262	24	1116.35	0.037	0.28	60
B/100	11	2597.60	420	7	2599.27	0.07	0.25	300
C/25	37	470.84	9	34	471.36	0.11	0.14	30
C/50	28	890.25	330	26	890.82	0.06	0.15	60
C/100	10	2129.36	127	8	2129.80	0.02	0.24	300
<i>Low Outsourcing Costs</i>								
A/25	13	1012.79	2	12	1012.86	0.0	0.25	30
A/50	12	2107.9	153	7	2109.85	0.094	0.47	60
A/100	3	4297.67	228	1	4323.60	0.6	1.06	300
B/25	25	585.23	4	25	585.23	0	0.19	30
B/50	19	1096.34	179	16	1096.91	0.05	0.22	60
B/100	6	2243.65	532	4	2244.70	0.05	0.25	300
C/25	29	483.61	23	28	483.72	0.023	0.29	30
C/50	20	920.57	289	19	921.125	0.060	0.23	60
C/100	6	1949.58	562	4	1951.60	0.12	0.45	300

Table 5: Comparison to optimal solutions: results of LNS-SPP on VRP-PC instances compared to the branch-and-cut-and-price of [Baller et al. \(2020\)](#)

The column entitled # shows the number of instances (out of 56) solved to optimality by the BCP algorithm of [Baller et al. \(2020\)](#) within the one hour time limit. Note that these results ignore instances for which the optimal solution consists in shipping all orders in LTL. Full results for instances with 100 clients are available in the Appendix (Tables A.8 and A.9)¹ Column 4 indicates the average CPU time (in seconds) necessary to get optimal solutions, for all instances that could be solved to optimality (otherwise, the CPU is 1h.

The LNS-SPP algorithm is run five times on each instance. The column entitled #opt indicates the number of instances for which the optimal solution was found on at least one of the runs. Column 5 (z_{avg}^{best}) represents the average of the best objective function (out of 5 runs) obtained on each instance.

¹The slightly negative gap comes from different rounding procedure in [Dabia et al. \(2019\)](#) and in this work, but the solutions found in both works are the same ([Baller et al., 2021](#)). Despite a constructive discussion with the authors and a careful examination of the optimal solutions we have not been able to find the precise origin of the difference.

The column entitled gap_{avg}^{best} indicates the optimality gap associated with the value z_{avg}^{best} while the column gap_{avg}^{avg} indicates the average gap across all runs. The computation time (column 8) was set to 30, 60, and 300 seconds respectively for the 25, 50, and 100 customer instances.

This table shows that the LNS-SPP algorithm is able to find near optimal or optimal solutions for most instances in all categories, including half of the instances with 100 customers.

5.3. Instances and results for the VRP-FLC

We propose new VRP-FLC instances adapted from those of [Dabia et al. \(2019\)](#) by considering two new features of the VRP-FLC: the presence of several carriers and a limitation on the total number of vehicles chartered to each carrier. The vehicle fleets are composed of 3 to 6 types of vehicles with high, medium, and low fixed costs (instances of type A, B, and C) and fleet size set to 3. We assign the first half of the vehicle types (rounded up to the nearest integer value) to one carrier and the remaining vehicle types to a second carrier. The carrier limits are set as follows: carriers that have 2 vehicle types can use at most 5 out of their 6 vehicles ; carriers that have 3 vehicle types can use at most 7 out of their 9 vehicles. A VRP-FLC instance is generated for each VRPPC instance with high or low outsourcing cost, amounting to a total of 336 instances.

Tables 6 and 7 summarizes the results on these new instances, for instances with high outsourcing costs and low outsourcing costs, respectively.

In these tables, we compare the results of the LNS-SPP algorithm on the two following variants of the VRPPC: the original VRPPC instances (columns 2-4) and the VRP-FLC including the carrier limit (columns 5-8). For each category of instance, we report the value of the best objective function z_{min} found out of 5 runs and the average value z_{avg} . Columns 4 and 8 show the cost breakdown in three categories: from left to right the percentage of fixed costs of the FTL fleet, the percentage of variable costs which depend on the distance traveled, and the percentage of LTL shipping costs.

Column 6 reports the relative gap between the best solutions found in both cases. This indicator shows the impact of carrier constraints on the routing costs.

When the vehicle fixed cost is high (type A instances), most optimal solutions use less than the carrier limit which has a limited impact in this case. The VRP-FLC solution uses more LTL shipping but a part of the additional cost is offset by the fixed vehicle costs that are not paid anymore.

When the vehicle fixed costs are low (type C instances), the fleet is fully utilized and routing costs represent the main costs. In this case, the carrier limit has a significant impact.

Little impact is observed on clustered instances. When the outsourcing cost is low, the whole flow is shipped in LTL mode so that the fleet limitation has no impact (clustered instances in Table 7).

6. Case study : combining FTL and LTL shipments

In this section we study how the combination of FTL and LTL shipments can achieve savings compared with a pure FTL or LTL approach.

6.1. Description

The instances were generated based on the experience of CRC Services to reflect typical situations faced by their clients. We consider the distribution of a shipper over an 18000 km² territory in France. The instances feature around 60 orders per day on average, distributed over 25 customer locations. The total average demand represents approximately 300 pallets per day. This quantity represents a little less than 9 full truckloads, but more vehicles are needed due to tight time windows and release dates.

For this set of experiments, we consider a single LTL carrier and a single FTL carrier. We use realistic LTL and FTL rates with two types of vehicles of capacity 33 pallets and 19 pallets, respectively. The cost of an FTL route follows the following model: (i) a route can serve at most three customers, (ii) an FTL delivery cost is given for each customer, (iii) the cost of a route is equal to the highest FTL delivery cost among the customers served by the route (plus a stopover cost per intermediate customer between the shipper and the last served customer), (iv) the total length of a route is equal to the distance traveled from the depot to the last customer of the route (i.e., we consider open routes), and (v) the maximum possible detour (difference between the route length and the distance between

Inst	VRPPC			VRP-FLC			
	z min	z avg	Cost distribution Fix Var LTL	z min	Δ (%)	z avg	Cost distribution Fix Var LTL
R1A	4426.8	4456		4622.3	4.4	4648.2	
R1B	1967.6	2001.8		2463.2	25.2	2491.9	
R1C	1661.5	1692.1		2194.2	32.1	2221.9	
RC1A	4987.8	5000.4		5090.4	2.1	5098.2	
RC1B	2685.3	2696.5		3086.2	14.9	3096.9	
RC1C	2400.2	2410.8		2830.6	17.9	2842	
C1A	3620	3620		3620	0.0	3620	
C1B	2519.5	2528.7		2584.2	2.6	2598.4	
C1C	1783.7	1794.7		1882.3	5.5	1896.3	
R2A	3425.2	3463.6		3434.1	0.3	3466.9	
R2B	1413.2	1450.9		1415	0.1	1455.8	
R2C	1154.8	1187		1154.6	0.0	1186.4	
RC2A	4160.7	4211.3		4222.4	1.5	4283.2	
RC2B	1705.7	1729		1708	0.1	1733	
RC2C	1365.1	1389.1		1364	-0.1	1385.5	
C2A	3620	3620		3620	0.0	3620	
C2B	1648.7	1668.7		1648.5	0.0	1668.9	
C2C	1167.9	1184.1		1168.4	0.0	1185	

Table 6: Results for high outsourcing cost instances

Inst	VRPPC			VRP-FLC			
	z min	z avg	Cost distribution Fix Var LTL	z min	Δ (%)	z avg	Cost distribution Fix Var LTL
R1A	4230	4249.7		4239	0.2	4253.6	
R1B	1897.2	1919.4		2144.3	13.0	2161.7	
R1C	1592.5	1612.5		1872.5	17.6	1891.9	
RC1A	3448	3448		3448	0.0	3448	
RC1B	2194.2	2200.5		2313.2	5.4	2319	
RC1C	1907	1913.1		2061.6	8.1	2068.9	
C1A	905	905		905	0.0	905	
C1B	905	905		905	0.0	905	
C1C	905	905		905	0.0	905	
R2A	3307.4	3363.9		3306.1	0.0	3359.8	
R2B	1412.7	1456.3		1413.8	0.1	1453	
R2C	1153.4	1188.1		1153.2	0.0	1189.7	
RC2A	3448	3448		3448	0.0	3448	
RC2B	1682	1722		1684.1	0.1	1725.8	
RC2C	1342.6	1377.8		1343.7	0.1	1370.2	
C2A	905	905		905	0.0	905	
C2B	905	905		905	0.0	905	
C2C	905	905		905	0.0	905	

Table 7: Results for low outsourcing cost instances

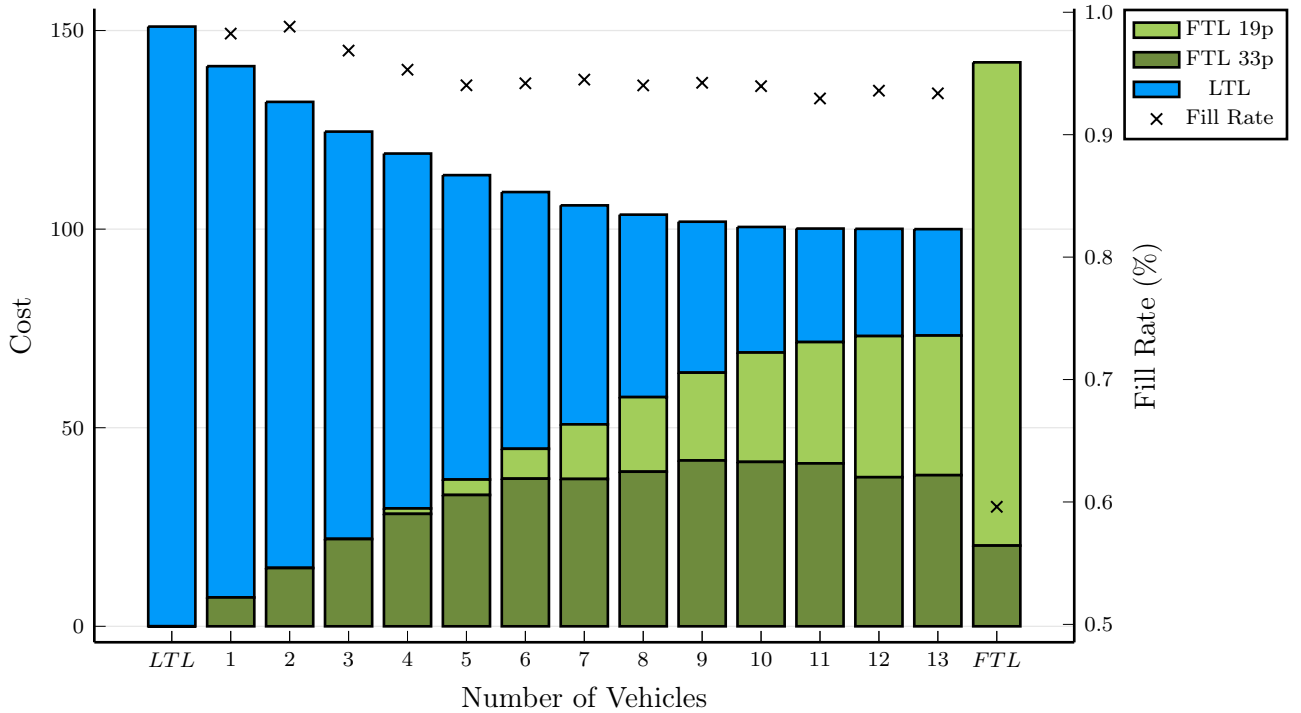


Figure 7: Evolution of the total cost and FTL trucks fill rate according to shipping policy and carrier limit. The cost is expressed in percentage with the minimum cost scenario (carrier limit 13) taken as a reference.

the shipper and the farthest customer in the route) is limited to 50 km. The LTL rate is a concave increasing function of the number of pallets to be shipped. It differs for each region. Distances and travel times between each pair of sites are obtained with OSRM (Open Source Routing Machine) using a truck routing profile.

6.2. Experiments

We first compare three different shipping strategies: shipping all orders with LTL rates, or with FTL rates, or allowing a mix of both. Figure 7 synthesizes the results.

We first run the experiments without any carrier limit; the minimum cost was obtained with a fleet of 13 vehicles. Thus, we set a carrier limit (represented on the horizontal axis) to values ranging from 1 to 13. The horizontal axis also includes two extreme scenarios: full LTL distribution on the left and full FTL distribution on the right. The vertical axis represents costs. The cost with a carrier limit set to 13 is considered as a reference cost (100%) and all other cost are expressed as percentages with respect to this reference cost.

In the full LTL scenario (on the left), the high LTL rates cause a 51% increase of the shipping costs. The full FTL scenario (on the right) suffers from the transportation of small shipments over a relatively large territory: many small vehicles are necessary to respect the time windows and route duration constraints. With this strategy, between 20 and 24 trucks are used per day, 88% of which have a capacity of 19 pallets. This causes a 42% increase of the total shipping cost, with a 60% average fill rate of the FTL trucks.

When mixing FTL and LTL shipping modes without limiting the number of FTL vehicles, the average number of trucks used for FTL routes drops to 11.5, the percentage of 19-pallets trucks drops to 66% and the average fill rate of FTL trucks increases to 94%.

For each scenario, the blue part represents the LTL costs and the green parts represents the FTL costs of the 19- and 33-pallets vehicles.

The figure shows that selecting an appropriate mix of FTL and LTL shipments is critical in terms of cost. Once a good vehicle mix has been determined, it seems to remain quite robust with respect to variations in the carrier limit. Indeed, decreasing the carrier limit from 13 to 10 has very limited

impact on the overall cost but it slightly modifies the vehicle mix. Below ten, the carrier limit drastically reduces the use of 19-pallet vehicles and the shipper resorts to more expensive LTL shipments.

7. Conclusions

In this paper, we investigate an extension of the vehicle routing problem with heterogeneous fleet arising in distribution networks, when shippers work with different FTL and LTL carriers. We introduce the VRP-FLC and propose an integer linear programming formulation of this problem as a variant of the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC), in which the private fleet consists of different fleets offered by various FTL carriers. In addition to the classical features of the VRP problem with heterogeneous fleet, we introduce the carrier limit bounding the maximum number of vehicles available at a carrier.

We propose a Large Neighborhood Search based matheuristic to solve this problem. In order to manage vehicles and carriers related constraints, a key point in this algorithm is to maintain the list of vehicle types compatible with each route of the current solution during the rebuild phase of the LNS. The numerical experiments show that our algorithm produces good results on several variants of the VRP with heterogeneous fleet. It shows that mixing FTL and LTL shipments in a distribution plan clearly outperforms pure FTL or LTL transportation strategies. Our algorithm helps identify which orders should be sent with each transportation mode. Our approach is also able to assess the impact of the carrier limit constraint and thus can help shippers agree with carriers on the right number of trucks to be chartered.

This work can be expanded in several ways.

Firstly, a natural expansion would be to consider multiple LTL carriers and integrate the LTL trucks into the carrier limit. Secondly, if we consider a multi-commodity case, the cost of delivering multiple commodities to the same client via LTL would impact the LTL rate which would make it impossible to determine the LTL cost of an order a priori. This cost could however be integrated as part of the already existing black box cost evaluation function. Another realistic expansion concerns a logistic network with multiple shippers and customers, with the possibility to deliver orders at transshipment facilities where the incoming flow coming from each shipper depot could be consolidated. This organization introduces the possibility to ship orders in two successive legs that can be either in FTL or LTL.

In a mixed FTL-LTL context, LTL shipment can be viewed as a way to outsource “complicated” deliveries. For example, deliveries to remote customer or with tight time windows. From a managerial point of view, it would be interesting to further analyze the typology of customer orders that are typically shipped in FTL or LTL mode.

Acknowledgement

The authors thank Annelieke Baller, Saïd Dabia, Wout Dullaert and Daniele Vigo for their precious help.

References

- C. Archetti, M. G. Speranza, and D. Vigo. Chapter 10: Vehicle Routing Problems with Profits. In P. Toth and D. Vigo, editors, *Vehicle Routing*, pages 273–297. Society for Industrial and Applied Mathematics, Philadelphia, PA, Nov. 2014.
- A. C. Baller, S. Dabia, W. E. H. Dullaert, and D. Vigo. The Vehicle Routing Problem with Partial Outsourcing. *Transportation Science*, 54(4):1034–1052, Aug. 2020.
- A. C. Baller, S. Dabia, W. E. H. Dullaert, and D. Vigo. Private communication, 2021.
- M.-C. Bolduc, J. Renaud, and F. Boctor. A heuristic for the routing and carrier selection problem. *European Journal of Operational Research*, 183(2):926–932, Dec. 2007.
- M.-C. Bolduc, J. Renaud, F. Boctor, and G. Laporte. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, 59(6):776–787, June 2008.

- T. Bulhões, M. H. Hà, R. Martinelli, and T. Vidal. The vehicle routing problem with service level constraints. *European Journal of Operational Research*, 265(2):544–558, Mar. 2018.
- S. Ceschia, L. Di Gaspero, and A. Schaerf. Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. *Journal of Scheduling*, 14(6):601–615, Dec. 2011.
- J. Christiaens and G. Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020.
- C.-W. Chu. A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research*, 165(3):657–667, Sept. 2005.
- G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.
- J.-F. Côté and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research*, 198(2), Oct. 2009.
- S. Dabia, D. Lai, and D. Vigo. An Exact Algorithm for a Rich Vehicle Routing Problem with Private Fleet and Common Carrier. *Transportation Science*, 53(4):986–1000, Aug. 2019.
- G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, 104(1):86–92, 1993.
- D. Dumez, F. Lehuédé, and O. Péton. A large neighborhood search approach to the vehicle routing problem with delivery options. *Transportation Research Part B: Methodological*, 144:103–132, 2021.
- J. Euchi. The vehicle routing problem with private fleet and multiple common carriers: Solution with hybrid metaheuristic algorithm. *Vehicular Communications*, 9:97–108, July 2017.
- R. F. Fachini and V. A. Armentano. Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering*, 148:106641, Oct. 2020.
- C. Gahm, C. Brabänder, and A. Tuma. Vehicle routing with private fleet, multiple common carriers offering volume discounts, and rental options. *Transportation Research Part E: Logistics and Transportation Review*, 97:192–216, Jan. 2017.
- D. Goeke, T. Gschwind, and M. Schneider. Upper and lower bounds for the vehicle-routing problem with private fleet and common carrier. *Discrete Applied Mathematics*, 264:43–61, July 2019.
- B. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11:49–66, Jan. 1984.
- P. Grangier, M. Gendreau, F. Lehuédé, and L.-M. Rousseau. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research*, 84:116–126, 2017.
- C. Groër, B. Golden, and E. Wasil. A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing*, 23(2):315–330, 2011.
- T. Gschwind and M. Drexler. Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. *Transportation Science*, 53(2):480–491, 2019.
- C. Koç, T. Bektaş, O. Jabali, and G. Laporte. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research*, 64:11–27, Dec. 2015.

- C. Koç, T. Bektaş, O. Jabali, and G. Laporte. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1–21, Feb. 2016.
- K. A. Lindsey, A. L. Erera, and M. W. P. Savelsbergh. A pickup and delivery problem using crossdocks and truckload lane rates. *EURO Journal on Transportation and Logistics*, 2(1-2):5–27, May 2013.
- F.-H. Liu and S.-Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society*, 50(7):721–732, 1999.
- R. Liu, Z. Jiang, X. Liu, and F. Chen. Task selection and routing problems in collaborative truckload transportation. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):1071–1085, Nov. 2010.
- M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.
- S. Mancini. A combined multistart random constructive heuristic and set partitioning based formulation for the vehicle routing problem with time dependent travel times. *Computers & Operations Research*, 88:290–296, 2017.
- S. Martello. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimization*, 1990.
- J. E. Mendoza and J. G. Villegas. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters*, 7(7):1503–1516, 2013.
- D. C. Paraskevopoulos, P. P. Repoussis, C. D. Tarantilis, G. Ioannou, and G. P. Prastacos. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14(5):425–455, Oct. 2008.
- S. N. Parragh and V. Schmid. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 40(1):490–497, 2013.
- P. H. V. Penna, A. Subramanian, L. S. Ochi, T. Vidal, and C. Prins. A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. *Annals of Operations Research*, 273(1): 5–74, 2019.
- D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.
- D. Pisinger and S. Ropke. Large neighborhood search. In *Handbook of metaheuristics*, pages 99–127. Springer, 2019.
- J.-Y. Potvin and M.-A. Naud. Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society*, 62(2):326–336, Feb. 2011.
- E. Prescott-Gagnon, G. Desaulniers, and L.-M. Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204, 2009.
- C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, Sept. 2009.
- S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006a.
- S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006b.

- J.-C. Régim. Generalized arc consistency for global cardinality constraint. *American Association for Artificial Intelligence (AAAI'96)*, pages 209–215, 1996.
- M. W. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations research*, 4(1):285–305, 1985.
- G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.
- P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming*, pages 417–431. Springer, 1998.
- M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.
- A. Subramanian, E. Uchoa, and L. S. Ochi. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531, 2013.
- E. D. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14, Jan. 1999.
- O. Tellez, S. Vercaene, F. Lehuédé, O. Péton, and T. Monteiro. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 91:99–123, 2018.
- T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.
- T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673, 2014.
- T. Vidal, N. Maculan, L. S. Ochi, and P. H. V. Penna. Large Neighborhoods with Implicit Customer Selection for Vehicle Routing Problems with Profits. *Transportation Science*, page 16, 2015.
- U. M. Yıldırım and B. Çatay. An ant colony-based matheuristic approach for solving a class of vehicle routing problems. In *International Conference on Computational Logistics (ICCL)*, pages 105–119. Springer, 2015.

Appendix A. Detailed results

		VRPPC					VRPFLC				
Instance	z^*	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
R101A	5053.9	5054.2	5060.2	$A^3B^3C^3D^3E^2$	2560	1390	5138.9	5153.0	$A^1B^3C^3D^3E^2$	2460	1695
R102A		4585.5	4608.8	$A^2B^3C^3D^3E^3$	3010	365	4683.1	4717.7	$A^1B^3C^3D^2E^3$	2710	1005
R103A		4353.1	4394.4	$A^3B^3C^3D^3E^3$	3060	190	4546.4	4571.6	$A^1B^3C^3D^2E^3$	2710	1005
R104A		4267.7	4296.3	$A^3B^3C^3D^3E^3$	3060	125	4519.0	4549.1	$A^1B^3C^3D^2E^3$	2710	990
R105A		4622.0	4657.6	$A^2B^3C^3D^3E^3$	3010	430	4753.4	4782.0	$A^1B^3C^3D^2E^3$	2710	1400
R106A		4403.4	4442.8	$A^2B^3C^3D^3E^3$	3010	270	4610.1	4635.0	$A^1B^3C^3D^2E^3$	2710	1000
R107A		4298.0	4333.2	$A^3B^3C^3D^3E^3$	3060	155	4532.0	4551.2	$A^1B^3C^3D^2E^3$	2710	1005
R108A		4242.7	4271.8	$A^3B^3C^3D^3E^3$	3060	115	4509.4	4529.6	$A^1B^3C^3D^2E^3$	2710	1000
R109A		4393.8	4427.9	$A^3B^3C^3D^3E^3$	3060	160	4562.1	4615.8	$A^1B^3C^3D^2E^3$	2710	1005
R110A		4327.5	4360.3	$A^3B^3C^3D^3E^3$	3060	155	4550.7	4577.6	$A^1B^3C^3D^2E^3$	2710	1005
R111A		4311.3	4336.8	$A^3B^3C^3D^3E^3$	3060	155	4531.0	4543.3	$A^1B^3C^3D^2E^3$	2710	990
R112A		4262.1	4282.4	$A^3B^3C^3D^3E^3$	3060	115	4502.0	4540.4	$A^1B^3C^3D^2E^3$	2710	1010
R201A		3668.6	3726.3	A^2B^2	2300	310	3716.2	3775.4	A^2B^2	2300	300
R202A		3526.5	3601.2	A^2B^2	2300	295	3507.3	3584.0	A^2B^2	2300	290
R203A		3436.7	3476.7	A^2B^2	2300	295	3412.4	3468.9	A^2B^2	2300	10
R204A		3285.6	3317.2	A^1B^3	2550	10	3300.7	3321.9	A^1B^3	2550	5
R205A		3520.4	3551.5	A^2B^2	2300	305	3471.1	3528.6	A^2B^2	2300	305
R206A		3409.8	3462.2	A^2B^2	2300	290	3438.3	3473.9	A^2B^2	2300	15
R207A		3375.3	3399.4	A^1B^3	2550	0	3363.5	3391.0	A^1B^3	2550	10
R208A		3264.9	3280.6	A^1B^3	2550	5	3270.5	3276.1	A^1B^3	2550	5
R209A		3438.1	3452.2	A^1B^3	2550	5	3404.6	3452.5	A^2B^2	2300	10
R210A		3436.4	3489.6	A^2B^2	2300	300	3436.1	3486.4	A^2B^2	2300	15
R211A		3314.5	3342.3	A^1B^3	2550	5	3309.9	3328.3	A^1B^3	2550	5
RC101A	5161.7	5161.7	5168.1	$A^3B^3C^3D^3$	2880	1144.5	5186.5	5190.3	$A^2B^3C^3D^2$	2370	1977.5
RC102A	5012.2	5016.5	5025.0	$A^3B^3C^3D^3$	2880	1123.5	5115.4	5126.2	$A^2B^3C^3D^2$	2370	1949.5
RC103A		4954.5	4967.2	$A^3B^3C^3D^3$	2880	1120	5073.4	5082.5	$A^2B^3C^3D^2$	2370	1949.5
RC104A		4908.0	4927.6	$A^3B^3C^3D^3$	2880	1113	5043.2	5050.1	$A^2B^3C^2D^3$	2520	1792
RC105A	5046.1	5046.1	5059.3	$A^3B^3C^3D^3$	2880	1144.5	5131.2	5137.4	$A^2B^3C^3D^2$	2370	1956.5
RC106A	4980.9	4980.9	4995.2	$A^3B^3C^3D^3$	2880	1113	5084.2	5093.4	$A^2B^3C^3D^2$	2370	1946
RC107A		4930.4	4944.1	$A^3B^3C^3D^3$	2880	1113	5050.1	5060.9	$A^2B^3C^2D^3$	2520	1771
RC108A		4904.3	4916.9	$A^3B^3C^3D^3$	2880	1113	5040.1	5048.3	$A^2B^3C^2D^3$	2520	1778
RC201A		4385.9	4423.9	$A^3B^2C^3$	2800	451.5	4484.9	4547.4	$A^2B^2C^3$	2650	451.5
RC202A		4239.4	4299.6	$A^3B^2C^3$	2800	458.5	4370.1	4427.2	$A^1B^3C^3$	2850	462
RC203A		4087.9	4142.7	$A^2B^3C^3$	3000	108.5	4173.9	4237.7	$A^3B^2C^2D^1$	3050	115.5
RC204A		3983.6	4033.4	$A^2B^3C^3$	3000	101.5	4046.0	4112.8	$A^3B^2C^2D^1$	3050	115.5
RC205A		4262.6	4332.2	$A^3B^2C^3$	2800	451.5	4365.3	4424.5	$A^1B^3C^3$	2850	455
RC206A		4210.3	4261.3	$A^2B^3C^3$	3000	101.5	4281.9	4363.4	$A^1B^3C^3$	2850	444.5
RC207A		4135.3	4186.2	$A^2B^3C^3$	3000	101.5	4210.9	4308.1	$A^1B^3C^3$	2850	444.5
RC208A		3980.7	4010.8	$A^2B^3C^3$	3000	91	4108.5	4156.8	$A^1B^3C^3$	2850	448
C101A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C102A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C103A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C104A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C105A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C106A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C107A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C108A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C109A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C201A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C202A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C203A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C204A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C205A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C206A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C207A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
C208A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620
R101B		2647.7	2660.6	$A^3B^3C^3D^3E^3$	612	840	3170.9	3176.6	$A^1B^3C^3D^3E^2$	492	1650
R102B		2129.7	2155.3	$A^3B^3C^3D^3E^3$	612	220	2515.6	2542.7	$A^1B^3C^3D^2E^3$	542	1025
R103B		1886.0	1926.8	$A^3B^3C^3D^3E^3$	612	180	2374.4	2408.0	$A^1B^3C^3D^2E^3$	542	1000
R104B		1794.1	1812.9	$A^3B^3C^3D^3E^3$	612	115	2332.5	2360.4	$A^1B^3C^3D^2E^3$	542	995
R105B		2163.7	2200.6	$A^3B^3C^3D^3E^3$	612	225	2598.7	2619.5	$A^1B^3C^3D^2E^3$	542	1060
R106B		1951.8	1994.2	$A^3B^3C^3D^3E^3$	612	185	2433.8	2456.8	$A^1B^3C^3D^2E^3$	542	995
R107B		1832.0	1874.2	$A^3B^3C^3D^3E^3$	612	140	2360.5	2387.2	$A^1B^3C^3D^2E^3$	542	1005
R108B		1779.6	1821.1	$A^3B^3C^3D^3E^3$	612	135	2332.2	2359.4	$A^1B^3C^3D^2E^3$	542	1005
R109B		1927.1	1957.8	$A^3B^3C^3D^3E^3$	612	165	2424.9	2443.3	$A^1B^3C^3D^2E^3$	542	1005

VRPPC						VRPFLC					
Instance	z^*	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
R110B		1862.3	1906.1	$A^3B^3C^3D^3E^3$	612	125	2394.0	2409.4	$A^1B^3C^3D^2E^3$	542	1005
R111B		1844.7	1870.0	$A^3B^3C^3D^3E^3$	612	135	2356.2	2370.7	$A^1B^3C^3D^2E^3$	542	995
R112B		1792.6	1841.8	$A^3B^3C^3D^3E^3$	612	130	2348.7	2366.7	$A^1B^3C^3D^2E^3$	542	1000
R201B		1725.1	1740.5	A^3B^2	550	30	1716.1	1732.5	A^3B^2	550	30
R202B		1562.0	1595.9	A^3B^2	550	30	1561.0	1587.6	A^1B^3	510	30
R203B		1412.8	1450.4	A^1B^3	510	0	1415.7	1451.4	A^1B^3	510	10
R204B		1266.5	1307.3	A^3C^1	510	5	1251.3	1290.6	A^1B^3	510	5
R205B		1490.5	1536.1	A^1B^3	510	15	1489.6	1517.8	A^1B^3	510	20
R206B		1399.9	1429.8	A^1B^3	510	15	1410.8	1435.9	A^1B^3	510	15
R207B		1337.0	1376.3	A^3C^1	510	0	1339.6	1375.7	A^3C^1	510	10
R208B		1250.4	1293.1	A^3C^1	510	5	1230.6	1279.6	A^1B^3	510	5
R209B		1384.5	1427.1	A^1B^3	510	15	1379.0	1405.9	A^1B^3	510	10
R210B		1440.0	1471.9	A^1B^3	510	15	1438.9	1468.9	A^1B^3	510	15
R211B		1276.8	1331.3	A^1B^3	510	5	1274.1	1305.8	A^1B^3	510	20
RC101B		2857.7	2865.4	$A^3B^3C^3D^3$	576	1144.5	3211.1	3213.6	$A^2B^3C^2D^3$	504	1802.5
RC102B		2710.7	2715.3	$A^3B^3C^3D^3$	576	1116.5	3109.5	3117.7	$A^2B^3C^2D^3$	504	1767.5
RC103B		2650.5	2658.6	$A^3B^3C^3D^3$	576	1127	3059.5	3070.5	$A^2B^3C^2D^3$	504	1764
RC104B		2607.4	2622.2	$A^3B^3C^3D^3$	576	1116.5	3027.7	3034.3	$A^2B^3C^2D^3$	504	1785
RC105B		2749.6	2760.0	$A^3B^3C^3D^3$	576	1116.5	3123.8	3134.7	$A^2B^3C^2D^3$	504	1785
RC106B		2678.0	2702.3	$A^3B^3C^3D^3$	576	1113	3076.5	3088.5	$A^2B^3C^2D^3$	504	1774.5
RC107B		2630.4	2636.5	$A^3B^3C^3D^3$	576	1120	3044.1	3049.9	$A^2B^3C^2D^3$	504	1785
RC108B		2598.1	2611.3	$A^3B^3C^3D^3$	576	1109.5	3023.1	3032.0	$A^2B^3C^2D^3$	504	1771
RC201B		1952.8	1991.3	$A^3B^3C^3$	630	0	1972.0	1993.1	$A^2B^3C^2D^1$	650	17.5
RC202B		1794.0	1819.2	$A^2B^3C^2D^1$	650	0	1787.2	1815.8	$A^1B^2C^3D^1$	660	0
RC203B		1635.2	1648.5	$A^1B^2C^3D^1$	660	0	1632.1	1656.8	C^2D^3	700	0
RC204B		1512.5	1522.5	C^2D^3	700	0	1512.5	1532.4	C^2D^3	700	0
RC205B		1845.5	1863.4	$A^3B^3C^3$	630	0	1839.3	1880.9	$A^2B^3C^2D^1$	650	73.5
RC206B		1748.0	1788.4	$A^1B^2C^3D^1$	660	7	1750.6	1789.8	$A^1C^3D^2$	680	0
RC207B		1667.3	1690.0	$A^1B^2C^3D^1$	660	0	1660.8	1695.4	$A^1B^2C^3D^1$	660	0
RC208B		1490.1	1508.4	C^2D^3	700	0	1491.9	1510.5	C^2D^3	700	0
C101B		2609.8	2609.8	$A^3B^3C^3$	1470	400	2695.0	2695.2	$A^2B^3C^3$	1410	640
C102B		2541.7	2550.0	$A^3B^3C^3$	1470	260	2609.7	2615.9	$A^2B^3C^3$	1410	480
C103B		2471.4	2484.2	$A^3B^3C^3$	1470	160	2533.8	2544.0	$A^2B^3C^3$	1410	360
C104B		2413.2	2429.5	$A^3B^3C^3$	1470	160	2424.1	2472.5	$A^2B^3C^3$	1410	280
C105B		2548.8	2548.8	$A^3B^3C^3$	1470	340	2630.3	2630.3	$A^2B^3C^3$	1410	540
C106B		2570.1	2570.9	$A^3B^3C^3$	1470	340	2651.6	2651.6	$A^2B^3C^3$	1410	560
C107B		2537.7	2551.0	$A^3B^3C^3$	1470	280	2616.0	2618.4	$A^2B^3C^3$	1410	500
C108B		2527.2	2531.7	$A^3B^3C^3$	1470	260	2575.5	2588.8	$A^2B^3C^3$	1410	460
C109B		2455.7	2482.0	$A^3B^3C^3$	1470	160	2512.7	2525.6	$A^2B^3C^3$	1410	340
C201B		1663.0	1675.8	A^2B^2	960	20	1660.8	1677.7	A^2B^2	960	20
C202B		1656.1	1661.0	A^2B^2	960	40	1655.4	1673.4	A^2B^2	960	40
C203B		1637.7	1646.6	A^2B^2	960	20	1638.4	1646.2	A^2B^2	960	20
C204B		1634.6	1657.6	A^2B^2	960	20	1632.2	1642.6	A^2B^2	960	40
C205B		1653.5	1663.6	A^2B^2	960	40	1654.9	1676.3	A^2B^2	960	20
C206B		1652.1	1679.9	A^2B^2	960	20	1652.8	1667.4	A^2B^2	960	20
C207B		1643.3	1685.5	A^2B^2	960	20	1643.3	1643.6	A^2B^2	960	20
C208B		1649.5	1679.3	A^2B^2	960	20	1649.5	1665.2	A^2B^2	960	20
R101C	2341.7	2341.7	2344.3	$A^3B^3C^3D^3E^3$	306	840	2901.7	2908.1	$A^1B^3C^3D^2E^3$	271	1645
R102C		1825.8	1855.1	$A^3B^3C^3D^3E^3$	306	255	2241.2	2269.6	$A^1B^3C^3D^2E^3$	271	1015
R103C		1585.6	1629.2	$A^3B^3C^3D^3E^3$	306	165	2103.9	2129.2	$A^1B^3C^3D^2E^3$	271	1005
R104C		1492.7	1518.0	$A^3B^3C^3D^3E^3$	306	130	2067.8	2092.3	$A^1B^3C^3D^2E^3$	271	990
R105C		1845.1	1884.3	$A^3B^3C^3D^3E^3$	306	230	2326.9	2362.2	$A^1B^3C^3D^2E^3$	271	1040
R106C		1643.8	1679.8	$A^3B^3C^3D^3E^3$	306	190	2151.4	2177.8	$A^1B^3C^3D^2E^3$	271	1005
R107C		1516.5	1551.6	$A^3B^3C^3D^3E^3$	306	110	2092.2	2125.8	$A^1B^3C^3D^2E^3$	271	1000
R108C		1478.9	1505.6	$A^3B^3C^3D^3E^3$	306	140	2067.2	2075.7	$A^1B^3C^3D^2E^3$	271	995
R109C		1637.5	1663.3	$A^3B^3C^3D^3E^3$	306	135	2128.6	2166.5	$A^1B^3C^3D^2E^3$	271	1005
R110C		1551.6	1593.6	$A^3B^3C^3D^3E^3$	306	150	2116.5	2134.2	$A^1B^3C^3D^2E^3$	271	1010
R111C		1528.0	1555.5	$A^3B^3C^3D^3E^3$	306	140	2087.9	2110.9	$A^1B^3C^3D^2E^3$	271	995
R112C		1491.2	1524.6	$A^3B^3C^3D^3E^3$	306	135	2069.7	2095.0	$A^1B^3C^3D^2E^3$	271	1005
R201C		1444.5	1460.3	A^3B^2	275	30	1439.1	1457.8	A^3B^2	275	15
R202C		1293.9	1327.8	A^3B^2	275	30	1296.4	1314.9	A^3B^2	275	30
R203C		1156.8	1187.4	A^1B^3	255	10	1177.7	1204.5	A^3B^2	275	10
R204C		1001.4	1041.7	A^1B^3	255	5	1010.8	1037.7	A^1B^3	255	5
R205C		1227.0	1264.6	A^1B^3	255	15	1230.2	1243.6	A^3B^2	275	15
R206C		1136.9	1160.2	A^3C^1	255	15	1153.8	1178.3	A^1B^3	255	15
R207C		1085.3	1126.4	A^1B^3	255	0	1078.1	1120.8	A^1B^3	255	0
R208C		1004.4	1022.2	A^1C^2	285	5	999.8	1024.5	A^3C^1	255	5
R209C		1127.4	1171.1	A^1B^3	255	20	1127.9	1149.1	A^1B^3	255	0

Instance	z^*	VRPPC					VRPFLC				
		z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
R210C		1190.2	1208.4	A^1B^3	255	15	1180.6	1191.1	A^1B^3	255	20
R211C		1035.2	1086.6	A^1B^3	255	15	1027.8	1051.7	A^1B^3	255	10
RC101C	2569.7	2569.7	2574.0	$A^3B^3C^3D^3$	288	1144.5	2959.1	2962.4	$A^2B^3C^2D^3$	252	1802.5
RC102C	2420.2	2422.7	2431.3	$A^3B^3C^3D^3$	288	1116.5	2861.4	2867.3	$A^2B^3C^2D^3$	252	1778
RC103C		2361.5	2378.4	$A^3B^3C^3D^3$	288	1113	2810.2	2817.2	$A^2B^3C^2D^3$	252	1774.5
RC104C		2321.2	2329.5	$A^3B^3C^3D^3$	288	1109.5	2773.0	2781.8	$A^2B^3C^2D^3$	252	1774.5
RC105C	2454.1	2461.8	2471.2	$A^3B^3C^3D^3$	288	1127	2871.8	2880.0	$A^2B^3C^2D^3$	252	1785
RC106C	2388.9	2404.2	2420.2	$A^3B^3C^3D^3$	288	1113	2822.7	2832.9	$A^2B^3C^2D^3$	252	1785
RC107C		2341.6	2348.9	$A^3B^3C^3D^3$	288	1120	2781.3	2796.1	$A^2B^3C^2D^3$	252	1778
RC108C		2319.2	2333.2	$A^3B^3C^3D^3$	288	1113	2770.9	2776.7	$A^2B^3C^2D^3$	252	1767.5
RC201C		1625.8	1645.8	$A^1B^2C^3D^1$	330	21	1637.8	1651.9	$A^2B^3C^2D^1$	325	10.5
RC202C		1444.1	1471.4	$A^1B^2C^3D^1$	330	17.5	1457.3	1480.9	$A^1B^2C^3D^1$	330	17.5
RC203C		1280.4	1300.2	C^2D^3	350	0	1280.4	1299.9	C^2D^3	350	0
RC204C		1163.6	1185.9	$C^3D^1E^1$	355	0	1163.6	1188.5	$C^3D^1E^1$	355	0
RC205C		1505.9	1523.5	$A^1B^2C^3D^1$	330	28	1506.3	1532.7	$A^1B^2C^3D^1$	330	28
RC206C		1418.2	1447.3	$A^1C^3D^2$	340	24.5	1410.6	1442.2	$A^1C^3D^2$	340	24.5
RC207C		1336.4	1351.5	$A^1C^3D^2$	340	0	1337.5	1357.5	$A^1C^3D^2$	340	7
RC208C		1146.2	1186.8	C^2D^3	350	0	1147.1	1186.6	C^2D^3	350	0
C101C	1874.8	1874.8	1874.8	$A^3B^3C^3$	735	400	1990.0	1990.9	$A^2B^3C^3$	705	640
C102C		1806.7	1820.4	$A^3B^3C^3$	735	260	1904.7	1910.7	$A^2B^3C^3$	705	460
C103C		1736.4	1754.9	$A^3B^3C^3$	735	160	1837.2	1854.2	$A^2B^3C^3$	705	380
C104C		1670.7	1703.4	$A^3B^3C^3$	735	140	1758.5	1776.5	$A^2B^3C^3$	705	300
C105C	1813.8	1813.8	1813.8	$A^3B^3C^3$	735	340	1925.3	1925.3	$A^2B^3C^3$	705	540
C106C	1835.1	1835.1	1835.1	$A^3B^3C^3$	735	340	1946.6	1946.6	$A^2B^3C^3$	705	560
C107C	1802.7	1802.7	1810.2	$A^3B^3C^3$	735	280	1911.0	1919.0	$A^2B^3C^3$	705	500
C108C	1792.6	1792.2	1794.1	$A^3B^3C^3$	735	260	1887.0	1890.7	$A^2B^3C^3$	705	460
C109C		1720.7	1746.0	$A^3B^3C^3$	735	160	1807.7	1821.9	$A^2B^3C^3$	705	340
C201C		1183.0	1183.6	A^2B^2	480	20	1182.1	1184.5	A^2B^2	480	40
C202C		1172.1	1193.3	A^3C^1	500	20	1172.1	1175.4	A^3C^1	500	40
C203C		1157.7	1176.3	A^2B^2	480	20	1158.4	1163.5	A^2B^2	480	20
C204C		1148.5	1159.4	A^2B^2	480	20	1151.1	1158.3	A^2B^2	480	20
C205C		1174.9	1179.7	A^2B^2	480	20	1174.9	1176.2	A^2B^2	480	20
C206C		1173.7	1194.7	A^2B^2	480	20	1172.1	1174.3	A^2B^2	480	20
C207C		1163.3	1202.2	A^2B^2	480	20	1163.3	1163.6	A^2B^2	480	20
C208C		1169.5	1184.0	A^2B^2	480	20	1169.5	1174.7	A^2B^2	480	20

Table A.8: Results on VRPPC and VRP-FLC instances, High Outsourcing Costs

Instance	VRPPC						VRPFLC				
	z^*	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
R101A	4376.8	4378.8	4384.3	$A^3B^3C^3D^3$	1560	2187.5	4409.5	4422.9	$A^1B^3C^3D^3$	1460.0	2387
R102A		4283.9	4291.5	$A^2B^3C^3D^3E^1$	2010	1585.5	4312.5	4326.0	$A^1B^3C^3D^3E^1$	1960.0	1008
R103A		4198.9	4218.4	$A^2B^3C^3D^3E^2$	2510	906.5	4221.9	4238.2	$A^1B^3C^3D^3E^2$	2460.0	987
R104A		4189.0	4200.4	$A^3B^3C^3D^3E^2$	2560	773.5	4194.3	4214.9	$A^1B^3C^3D^3E^2$	2460.0	980
R105A	4309.6	4317.1	4334.6	$A^3B^3C^3D^3E^1$	2060	1494.5	4345.8	4365.2	$A^1B^3C^3D^3E^1$	1960.0	1683.5
R106A		4227.7	4265.2	$A^2B^3C^3D^3E^2$	2510	882	4263.3	4278.9	$A^1B^3C^3D^3E^2$	2460.0	990.5
R107A		4201.5	4214.0	$A^3B^3C^3D^3E^2$	2560	787.5	4210.3	4233.7	$A^1B^3C^3D^3E^2$	2460.0	987
R108A		4167.9	4191.2	$A^3B^3C^3D^3E^2$	2560	773.5	4191.0	4217.3	$A^1B^3C^3D^3E^2$	2460.0	973
R109A	4206.6	4231.8	4262.1	$A^3B^3C^3D^3E^2$	2560	787.5	4246.9	4273.6	$A^1B^3C^3D^3E^2$	2460.0	980
R110A		4197.8	4225.9	$A^2B^3C^3D^3E^2$	2510	885.5	4229.2	4244.2	$A^1B^3C^3D^3E^2$	2460.0	987
R111A		4195.9	4213.5	$A^3B^3C^3D^3E^2$	2560	784	4212.7	4224.8	$A^1B^3C^3D^3E^2$	2460.0	990.5
R112A		4169.2	4194.8	$A^2B^3C^3D^3E^2$	2510	882	4192.5	4213.1	$A^1B^3C^3D^3E^2$	2460.0	980
R201A		3518.8	3605.5	A^3B^1	2050	560	3535.5	3631.7	A^3B^1	2050.0	567
R202A		3390.0	3463.0	A^3B^1	2050	560	3463.3	3512.8	A^3B^1	2050.0	556.5
R203A		3318.7	3358.1	A^2B^2	2300	210	3320.1	3422.3	A^3B^1	2050.0	556.5
R204A		3187.8	3250.9	A^2B^2	2300	203	3213.3	3251.3	A^2B^2	2300.0	206.5
R205A		3406.7	3500.9	A^3B^1	2050	560	3383.4	3423.9	A^3B^1	2050.0	217
R206A		3296.0	3368.2	A^3B^1	2050	556.5	3286.1	3359.1	A^2B^2	2300.0	553
R207A		3253.5	3276.2	A^3B^1	2050	556.5	3239.3	3295.0	A^2B^2	2300.0	220.5
R208A		3186.5	3220.6	A^2B^2	2300	210	3155.1	3193.6	A^2B^2	2300.0	203
R209A		3303.4	3351.0	A^2B^2	2300	206.5	3294.4	3361.4	A^2B^2	2300.0	210
R210A		3305.0	3363.2	A^2B^2	2300	206.5	3327.6	3395.3	A^2B^2	2300.0	553
R211A		3214.9	3245.5	A^2B^2	2300	203	3187.2	3220.6	A^2B^2	2300.0	206.5
RC101A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC102A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC103A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC104A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC105A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC106A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC107A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC108A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC201A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC202A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC203A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC204A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC205A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC206A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC207A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
RC208A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448
C101A		905.0	905.0		0	905	905.0	905.0		0.0	905
C102A		905.0	905.0		0	905	905.0	905.0		0.0	905
C103A		905.0	905.0		0	905	905.0	905.0		0.0	905
C104A		905.0	905.0		0	905	905.0	905.0		0.0	905
C105A		905.0	905.0		0	905	905.0	905.0		0.0	905
C106A		905.0	905.0		0	905	905.0	905.0		0.0	905
C107A		905.0	905.0		0	905	905.0	905.0		0.0	905
C108A		905.0	905.0		0	905	905.0	905.0		0.0	905
C109A		905.0	905.0		0	905	905.0	905.0		0.0	905
C201A		905.0	905.0		0	905	905.0	905.0		0.0	905
C202A		905.0	905.0		0	905	905.0	905.0		0.0	905
C203A		905.0	905.0		0	905	905.0	905.0		0.0	905
C204A		905.0	905.0		0	905	905.0	905.0		0.0	905
C205A		905.0	905.0		0	905	905.0	905.0		0.0	905
C206A		905.0	905.0		0	905	905.0	905.0		0.0	905
C207A		905.0	905.0		0	905	905.0	905.0		0.0	905
C208A		905.0	905.0		0	905	905.0	905.0		0.0	905
R101B		2393.3	2393.8	$A^3B^3C^3D^3E^3$	612	602	2655.0	2656.8	$A^1B^3C^3D^3E^2$	492.0	1274
R102B		2044.4	2060.9	$A^3B^3C^3D^3E^3$	612	220.5	2200.4	2243.5	$A^1B^3C^3D^2E^3$	542.0	749
R103B		1843.8	1869.9	$A^3B^3C^3D^3E^3$	612	154	2083.8	2103.7	$A^1B^3C^3D^2E^3$	542.0	703.5
R104B		1756.4	1777.5	$A^3B^3C^3D^3E^3$	612	122.5	2042.9	2061.8	$A^1B^3C^3D^2E^3$	542.0	696.5
R105B		2083.8	2102.9	$A^3B^3C^3D^3E^3$	612	241.5	2273.3	2295.9	$A^1B^3C^3D^2E^3$	542.0	801.5
R106B		1906.9	1934.6	$A^3B^3C^3D^3E^3$	612	140	2129.4	2144.9	$A^1B^3C^3D^2E^3$	542.0	717.5
R107B		1786.9	1822.5	$A^3B^3C^3D^3E^3$	612	101.5	2062.6	2088.7	$A^1B^3C^3D^2E^3$	542.0	700
R108B		1742.6	1754.9	$A^3B^3C^3D^3E^3$	612	91	2037.3	2053.7	$A^1B^3C^3D^2E^3$	542.0	696.5
R109B		1863.7	1886.3	$A^3B^3C^3D^3E^3$	612	168	2099.0	2129.3	$A^1B^3C^3D^2E^3$	542.0	717.5
R110B		1817.0	1845.7	$A^3B^3C^3D^3E^3$	612	143.5	2081.8	2096.3	$A^1B^3C^3D^2E^3$	542.0	703.5
R111B		1793.8	1817.5	$A^3B^3C^3D^3E^3$	612	119	2045.1	2079.0	$A^1B^3C^3D^2E^3$	542.0	703.5
R112B		1734.1	1766.6	$A^3B^3C^3D^3E^3$	612	108.5	2047.9	2068.5	$A^1B^3C^3D^2E^3$	542.0	707

VRPPC						VRPFLC					
Instance	z^*	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
R201B		1710.2	1757.0	A^3B^2	550	73.5	1718.0	1728.9	A^3B^2	550.0	63
R202B		1542.9	1601.1	A^1B^3	510	45.5	1544.6	1572.8	A^1B^3	510.0	73.5
R203B		1409.9	1443.0	A^3C^1	510	10.5	1414.1	1441.6	A^3C^1	510.0	49
R204B		1259.5	1278.2	A^3C^1	510	3.5	1267.8	1315.6	A^1B^3	510.0	3.5
R205B		1513.0	1556.5	A^3C^1	510	45.5	1477.8	1508.9	A^1B^3	510.0	24.5
R206B		1399.3	1453.0	A^3C^1	510	21	1405.3	1435.0	A^1B^3	510.0	31.5
R207B		1336.0	1400.0	A^1B^3	510	14	1327.1	1365.4	A^1B^3	510.0	0
R208B		1255.9	1296.9	A^3C^1	510	3.5	1258.7	1300.1	A^1B^3	510.0	3.5
R209B		1389.4	1426.5	A^1B^3	510	14	1378.6	1410.5	A^1B^3	510.0	14
R210B		1442.6	1466.0	A^1B^3	510	42	1433.8	1456.1	A^1B^3	510.0	38.5
R211B		1281.3	1340.6	A^3C^1	510	10.5	1273.1	1298.4	A^1B^3	510.0	21
RC101B		2344.1	2345.6	$A^3B^3C^3D^3$	576	744	2420.2	2424.4	$A^2B^3C^2D^3$	504.0	1086
RC102B		2221.1	2223.4	$A^3B^3C^3D^3$	576	680	2344.1	2350.0	$A^2B^3C^2D^3$	504.0	1034
RC103B		2164.9	2175.7	$A^3B^3C^3D^3$	576	670	2294.9	2301.8	$A^2B^3C^2D^3$	504.0	1016
RC104B		2122.0	2128.3	$A^3B^3C^3D^3$	576	650	2263.7	2270.9	$A^2B^3C^2D^3$	504.0	1026
RC105B		2241.5	2246.9	$A^3B^3C^3D^3$	576	678	2355.5	2357.9	$A^2B^3C^2D^3$	504.0	1036
RC106B		2198.2	2204.8	$A^3B^3C^3D^3$	576	674	2308.5	2318.9	$A^2B^3C^2D^3$	504.0	1042
RC107B		2144.3	2151.0	$A^3B^3C^3D^3$	576	664	2267.9	2276.3	$A^2B^3C^2D^3$	504.0	1024
RC108B		2117.6	2128.6	$A^3B^3C^3D^3$	576	658	2253.6	2264.7	$A^2B^3C^2D^3$	504.0	1022
RC201B		1918.7	1954.9	$A^1B^1C^2D^2$	640	178	1908.9	1935.4	$B^2C^3D^1$	630.0	120
RC202B		1756.5	1778.5	$A^1B^1C^3D^1$	590	268	1739.8	1779.7	$B^2C^3D^1$	630.0	162
RC203B		1604.4	1637.4	C^3D^2	650	58	1601.6	1618.1	C^3D^2	650.0	104
RC204B		1497.5	1533.3	C^3D^2	650	66	1476.4	1503.5	C^3D^2	650.0	64
RC205B		1792.9	1851.9	$A^1B^3C^3$	570	280	1788.0	1824.4	$A^2B^1C^3D^1$	620.0	140
RC206B		1768.0	1794.3	C^3D^2	650	60	1749.4	1769.2	$B^2C^3D^1$	630.0	74
RC207B		1641.1	1695.6	$B^2C^3D^1$	630	70	1628.0	1673.4	$B^2C^3D^1$	630.0	72
RC208B		1476.5	1529.9	C^3D^2	650	66	1477.5	1501.1	C^3D^2	650.0	52
C101B		905.0	905.0		0	905	905.0	905.0		0.0	905
C102B		905.0	905.0		0	905	905.0	905.0		0.0	905
C103B		905.0	905.0		0	905	905.0	905.0		0.0	905
C104B		905.0	905.0		0	905	905.0	905.0		0.0	905
C105B		905.0	905.0		0	905	905.0	905.0		0.0	905
C106B		905.0	905.0		0	905	905.0	905.0		0.0	905
C107B		905.0	905.0		0	905	905.0	905.0		0.0	905
C108B		905.0	905.0		0	905	905.0	905.0		0.0	905
C109B		905.0	905.0		0	905	905.0	905.0		0.0	905
C201B		905.0	905.0		0	905	905.0	905.0		0.0	905
C202B		905.0	905.0		0	905	905.0	905.0		0.0	905
C203B		905.0	905.0		0	905	905.0	905.0		0.0	905
C204B		905.0	905.0		0	905	905.0	905.0		0.0	905
C205B		905.0	905.0		0	905	905.0	905.0		0.0	905
C206B		905.0	905.0		0	905	905.0	905.0		0.0	905
C207B		905.0	905.0		0	905	905.0	905.0		0.0	905
C208B		905.0	905.0		0	905	905.0	905.0		0.0	905
R101C	2087.3	2087.3	2090.0	$A^3B^3C^3D^3E^3$	306	602	2397.2	2399.1	$A^1B^3C^3D^2E^3$	271.0	1214.5
R102C		1740.7	1753.1	$A^3B^3C^3D^3E^3$	306	238	1923.5	1942.8	$A^1B^3C^3D^2E^3$	271.0	724.5
R103C		1532.4	1559.6	$A^3B^3C^3D^3E^3$	306	126	1802.0	1819.8	$A^1B^3C^3D^2E^3$	271.0	714
R104C		1443.3	1462.2	$A^3B^3C^3D^3E^3$	306	98	1763.0	1785.0	$A^1B^3C^3D^2E^3$	271.0	696.5
R105C	1758.5	1776.0	1789.7	$A^3B^3C^3D^3E^3$	306	297.5	2009.9	2021.0	$A^1B^3C^3D^2E^3$	271.0	742
R106C		1580.6	1609.9	$A^3B^3C^3D^3E^3$	306	178.5	1850.3	1879.0	$A^1B^3C^3D^2E^3$	271.0	717.5
R107C		1480.9	1518.2	$A^3B^3C^3D^3E^3$	306	87.5	1783.7	1809.0	$A^1B^3C^3D^2E^3$	271.0	696.5
R108C		1447.1	1458.0	$A^3B^3C^3D^3E^3$	306	98	1763.8	1780.9	$A^1B^3C^3D^2E^3$	271.0	700
R109C		1577.8	1596.6	$A^3B^3C^3D^3E^3$	306	133	1824.6	1852.8	$A^1B^3C^3D^2E^3$	271.0	710.5
R110C		1513.2	1536.9	$A^3B^3C^3D^3E^3$	306	126	1805.2	1821.8	$A^1B^3C^3D^2E^3$	271.0	714
R111C		1482.4	1514.4	$A^3B^3C^3D^3E^3$	306	101.5	1769.8	1803.1	$A^1B^3C^3D^2E^3$	271.0	707
R112C		1448.2	1461.2	$A^3B^3C^3D^3E^3$	306	126	1758.3	1777.5	$A^1B^3C^3D^2E^3$	271.0	700
R201C		1432.9	1474.9	A^3B^2	275	108.5	1441.8	1451.8	A^3B^2	275.0	84
R202C		1277.3	1303.9	A^3B^2	275	45.5	1272.4	1303.2	A^3B^2	275.0	45.5
R203C		1159.5	1191.5	A^1B^3	255	28	1156.4	1170.2	A^3B^2	275.0	28
R204C		1003.2	1039.7	A^1B^3	255	10.5	995.9	1040.6	A^1B^3	255.0	3.5
R205C		1227.5	1259.0	A^3C^1	255	14	1219.7	1246.7	A^3B^2	275.0	21
R206C		1138.7	1195.0	A^1B^3	255	21	1143.2	1182.0	A^1B^3	255.0	10.5
R207C		1087.8	1120.9	A^1B^3	255	7	1084.9	1110.6	A^1B^3	255.0	28
R208C		1001.6	1022.2	A^1C^2	285	3.5	997.0	1019.2	A^1C^2	285.0	7
R209C		1134.9	1163.9	A^1B^3	255	7	1122.7	1136.0	A^1B^3	255.0	14
R210C		1177.6	1206.5	A^1B^3	255	10.5	1180.6	1198.2	A^1B^3	255.0	49
R211C		1046.8	1091.0	A^1B^3	255	14	1034.5	1062.6	A^1B^3	255.0	24.5
RC101C	2056.1	2056.1	2057.0	$A^3B^3C^3D^3$	288	744	2166.5	2171.5	$A^2B^3C^2D^3$	252.0	1086

VRPPC							VRPFLC				
Instance	z^*	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost	z_{min}	z_{avg}	Fleet	Fixed Cost	LTL Cost
RC102C	1933.1	1934.4	1938.1	$A^3B^3C^3D^3$	288	676	2091.9	2096.0	$A^2B^3C^2D^3$	252.0	1046
RC103C		1877.7	1884.3	$A^3B^3C^3D^3$	288	652	2042.9	2046.0	$A^2B^3C^2D^3$	252.0	1030
RC104C		1839.0	1846.5	$A^3B^3C^3D^3$	288	672	2008.3	2017.7	$A^2B^3C^2D^3$	252.0	1032
RC105C	1953.5	1953.5	1957.8	$A^3B^3C^3D^3$	288	678	2103.5	2105.6	$A^2B^3C^2D^3$	252.0	1036
RC106C	1909.0	1910.2	1921.5	$A^3B^3C^3D^3$	288	674	2056.5	2066.8	$A^2B^3C^2D^3$	252.0	1042
RC107C		1855.4	1859.2	$A^3B^3C^3D^3$	288	648	2019.6	2027.1	$A^2B^3C^2D^3$	252.0	1042
RC108C		1829.6	1840.0	$A^3B^3C^3D^3$	288	652	2008.4	2012.7	$A^2B^3C^2D^3$	252.0	1022
RC201C		1585.8	1607.1	$A^1B^3C^2D^1$	310	126	1598.4	1613.6	$B^2C^3D^1$	315.0	126
RC202C		1433.9	1460.9	C^3D^2	325	108	1417.7	1438.0	C^3D^2	325.0	144
RC203C		1260.1	1318.2	C^3D^2	325	58	1266.2	1291.6	C^2D^3	350.0	58
RC204C		1146.8	1163.6	C^3D^2	325	54	1159.0	1174.0	$C^1D^1E^2$	355.0	70
RC205C		1469.7	1488.1	$A^2B^1C^2D^2$	335	78	1476.1	1495.0	$A^2B^1C^2D^2$	335.0	58
RC206C		1392.9	1467.1	C^3D^2	325	88	1408.2	1439.9	$B^2C^2D^2$	340.0	12
RC207C		1310.4	1351.0	C^3D^2	325	78	1319.2	1340.0	$B^2C^2D^2$	340.0	98
RC208C		1141.5	1166.4	C^2D^3	350	6	1140.7	1165.3	C^2D^3	350.0	6
C101C		905.0	905.0		0	905	905.0	905.0		0.0	905
C102C		905.0	905.0		0	905	905.0	905.0		0.0	905
C103C		905.0	905.0		0	905	905.0	905.0		0.0	905
C104C		905.0	905.0		0	905	905.0	905.0		0.0	905
C105C		905.0	905.0		0	905	905.0	905.0		0.0	905
C106C		905.0	905.0		0	905	905.0	905.0		0.0	905
C107C		905.0	905.0		0	905	905.0	905.0		0.0	905
C108C		905.0	905.0		0	905	905.0	905.0		0.0	905
C109C		905.0	905.0		0	905	905.0	905.0		0.0	905
C201C		905.0	905.0		0	905	905.0	905.0		0.0	905
C202C		905.0	905.0		0	905	905.0	905.0		0.0	905
C203C		905.0	905.0		0	905	905.0	905.0		0.0	905
C204C		905.0	905.0		0	905	905.0	905.0		0.0	905
C205C		905.0	905.0		0	905	905.0	905.0		0.0	905
C206C		905.0	905.0		0	905	905.0	905.0		0.0	905
C207C		905.0	905.0		0	905	905.0	905.0		0.0	905
C208C		905.0	905.0		0	905	905.0	905.0		0.0	905

Table A.9: Results on VRPPC and VRP-FLC instances, Low Outsourcing Costs