Multi-depot Two-Echelon Fuel Minimizing Routing Problem with Heterogeneous Fleets: Model and Heuristic

Surendra Reddy Kancharla[a], Gitakrishnan Ramadurai[b,*]

[a] *Ph.D. Candidate, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai-600036, INDIA*
[b] *Associate Professor, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai-600036, INDIA*

Abstract

We formulate the two-echelon routing problem considering multiple depots and heterogeneous fleets. Our study (a) presents a Mixed Integer Linear Programming (MILP) formulation with load-dependent fuel minimization objective, (b) uses driving cycles to represent speed variations along a path, (c) allows the vehicles to return to any depot/satellite, and (d) conserves the total number of vehicles at each depot/satellite. We call the problem a Multi-Depot Two-Echelon Fuel Minimizing Routing Problem (MD2E-FMRP). Prior studies assumed there is a fixed number of vehicles available at each satellite/depot, whereas we allow different number of vehicles of each vehicle type at each satellite and depot. Our formulation relaxes several unrealistic assumptions in existing two-echelon formulations and hence has greater practical application. Despite the relaxation of constraints, the running time of our model is comparable to existing formulations. Gurobi optimizer is used to find a better upper bound for up to 56 node instances within a given time limit of 10,000s. We also propose an Adaptive Large Neighborhood Search (ALNS) based heuristic solution technique that outperformed Gurobi in all the tested instances of MD2E-FMRP. We observe an average saving of 13.11% in fuel consumption by minimizing fuel consumed instead of minimizing distance. In general, adapting heterogeneous fleets results in fuel savings and consequently lower emissions compared to using a homogeneous fleet.

Keywords: Multi-depot, Heterogeneous fleet, Adaptive large neighborhood search, Fuel consumption, Vehicle routing problem, Mixed integer linear programming.

1. Introduction

In recent years, many cities around the world are battling poor air quality (WHO, 2018). A recent report by World Health Organization (WHO) (WHO, 2016) certified that 98% of cities in developing countries and 56% of cities in developed countries with a population more than 100,000 fail air quality standards. Freight vehicles constitute only 5% of road traffic yet they produce over 30-40% of the air pollution in

*Corresponding author
Email address: gitakrishnan@iitm.ac.in (Gitakrishnan Ramadurai)

cities (Bathmanabhan and Madanayak, 2010). Even though freight transport plays a vital role in a cities' sustenance, their disproportionately higher negative effects have led city planners and managers to impose restrictions on their movement. These include full restrictions on large vehicles from entering parts of the city or limited restrictions during certain times of the day. Such restrictions may rise the cost of urban freight distribution. Innovative sustainable freight strategies can help develop solutions that balance the interests of all stakeholders.

Traditionally, goods are delivered to the customers from a central depot using large trucks. In this process, these large trucks may travel significantly longer distances carrying less than truckload or running empty leading to higher fuel consumption and emissions. Adopting a two-level distribution system and smaller vehicles (Jacobsen and Madsen, 1980) could reduce fuel consumption and emissions. Unwittingly, cities in developing economies have started adopting smaller delivery vehicles; most likely in response to vehicle restrictions. For example, in India, vehicle registrations of Light Commercial Vehicles (LCV) have crossed Heavy Commercial Vehicles (HCV) in 2008-2009, and by 2012-2013 they were twice the number of HCV registrations (MORTH, 2015). However, the use of small vehicles for longer distances may be inefficient; there are economies of scale benefits possible with the use of large vehicles.

In a two-level distribution system, large trucks deliver goods to satellites, and then smaller trucks deliver to customers; literature (Jacobsen and Madsen, 1980) refers to this set-up as the two-echelon distribution system. Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP) determines optimal vehicle routes in a two-echelon distribution system. These multi-level systems have applications in express delivery services, grocery and hypermarket-product distribution, spare-parts distribution, e-commerce and home delivery services, and newspaper and press distribution (Perboli et al., 2011). In 2E-CVRP, routes and other decision variables in both echelons are optimized. The routes and location of satellites in the second echelon are optimized first. The accessibility of each satellite and its demand are known from the solution of the second echelon, which are required to optimize the routes in the first echelon. If heterogeneous fleet is allowed, the optimal mix of the fleet at each satellite and depot need to be determined as well. Further, the need to improve air quality compels us to consider sustainable performance measures as objectives in the routing problem. Adopting sustainable performance measures may not be antithetical to profit-oriented freight operators. Fuel cost alone contributes to 40% of the overall cost for logistic operators (Sahin et al., 2009), and fuel consumption is directly related to emissions; minimizing fuel consumption rather than distance could be a win-win solution. A 2E-CVRP using a heterogeneous fleet, optimizing over sustainable performance measures, and allowing multiple satellites and depots requires a new optimal route planning paradigm such as the one we introduce in this paper. We (a) present a MILP formulation with load-dependent fuel minimization objective, (b) use driving cycle to represent speed variations along a path, (c) allow multiple depots, and a heterogeneous fleet that can start and end their routes at different depot/satellite, and (d) conserve the total number of vehicles at each depot/satellite. We call the problem a Multi-Depot Two-Echelon Fuel Minimizing Routing Problem (MD2E-FMRP).

Our study relaxes several assumptions in literature (Feliu et al., 2007; Perboli et al., 2011; Jepsen et al., 2013), such as ignoring the effect of acceleration in fuel consumption, fixing the number of vehicles at each depot/satellite, requiring every vehicle return to its starting depot/satellite after all deliveries, and operating a homogeneous fleet. The effects of relaxing these assumptions on fuel consumption and distance traveled are evaluated individually. Previous studies (Demir et al., 2012; Franceschetti et al., 2013; Demir et al., 2014) model the effect of load and speed on fuel consumption but ignore the effect of acceleration/deceleration. The effect of acceleration will be significant in urban areas where stop-and-go conditions are prevalent. To the best of our knowledge, existing literature do not consider the effects of acceleration/deceleration on fuel consumption. We use driving cycles (a series of data points representing the second-by-second speed of a vehicle) to estimate fuel consumption. Further, our formulation incorporates a heterogeneous fleet with varying fleet sizes at each depot/satellite and hence has greater practical appeal. We also allow vehicles to end at a different depot/satellite simultaneously conserving the total flow from each depot/satellite.

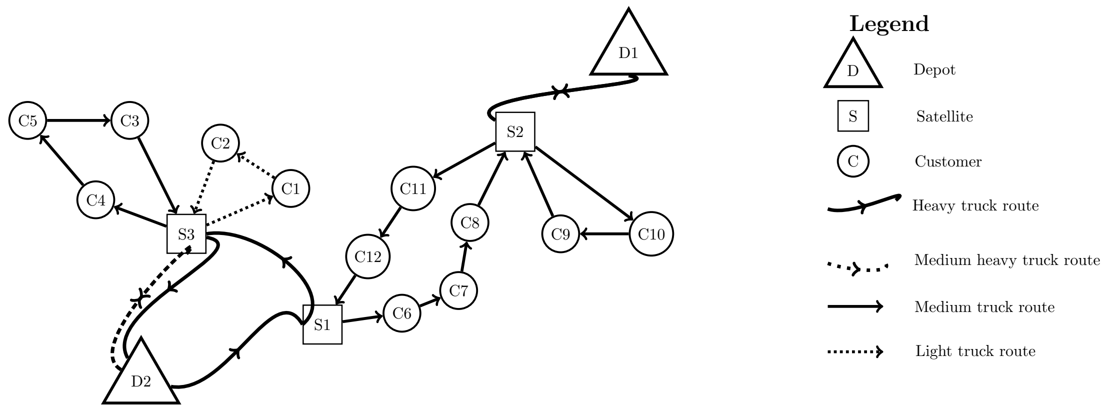


Figure 1: A solution to an MD2E-FMRP

The objective of the proposed MD2E-FMRP is to identify a set of routes that minimizes total fuel consumed in both the first and second echelons of the distribution system. Figure 1 is a typical solution to an MD2E-FMRP. In this figure, triangles represent depots, squares represent satellites, and circles represent customers. It highlights several features of the proposed MD2E-FMRP: (a) Heterogeneous Fleet: it uses four different vehicle classes. They are represented by different line types as shown in the legend. Two different vehicle classes are used to make deliveries from depot D2 and satellite S3; (b) Split delivery to satellite: multiple vehicles visit satellite S3; (c) End a route at any depot: a vehicle that started at S2 ends at S1, while another vehicle that started at S1 ends at S2; (d) Multi-depot: two depots are considered for deliveries; (e) No split in delivery to a customer: only one vehicle serves each customer.

The present study is the first, to the best of our knowledge, to consider multiple depots in the first echelon and heterogeneous fleet in both echelons. Existing 2E-CVRP formulations (Perboli et al., 2011;

Jepsen et al., 2013; Soysal et al., 2015) are incapable of handling multi-depots and heterogeneous fleets. In Perboli et al. (2011) constraints which identify the load on edge ensure sub-tour elimination; this approach can handle single-depot problems only. Similarly, the sub-tour elimination constraints in another study by Jepsen et al. (2013) cannot handle heterogeneous fleet. Also, Jepsen et al. (2013) had an additional index for each vehicle that increased the number of variables drastically. Another study by Soysal et al. (2015) did not consider split-deliveries in the first echelon, and their flow-based formulation does not require sub-tour elimination. Our study allows split deliveries from multiple depots and because our formulation is also flow-based, there is no need for specific sub-tour elimination constraints.

We perform extensive computational experiments with the proposed formulation. Even though the proposed MD2E-FMRP model has relaxed several assumptions, its computation time is comparable to existing formulations. The Gurobi Optimizer is used to find better upper bounds, and we propose a state-of-the-art Adaptive Large Neighborhood Search (ALNS) based heuristic solution.

The organization of the paper is as follows: In section 2 literature review on 2E-CVRP and related problems is presented. Section 3 introduces the MD2E-FMRP and provides a mathematical formulation. In section 4 the ALNS based heuristic is introduced for solving MD2E-FMRP. Section 5 introduces the test instances for MD2E-FMRP and discusses the computational results, followed by conclusions in section 6.

2. Literature Review

Over the past decade, unsurprisingly coinciding with the increase in urban freight traffic in many cities, the 2E-CVRP has received considerable attention (Feliu et al., 2007; Perboli et al., 2011; Jepsen et al., 2013). The two-echelon routing problem was first introduced by Jacobsen and Madsen (1980) though they use a different name for the problem. They had considered the newspaper distribution problem with few transfer points. They identify the number and location of such transfer points, and then routes from the printing office to transfer points to retailers. They developed a heuristic that assigns the retailers to the nearest transfer point and subsequently solves both first and second levels. The term 2E-CVRP was first coined by Feliu et al. (2007). They proposed a mathematical model that was later strengthened using a family of inequalities by Perboli et al. (2010). The strengthened model solved seven new instances to optimality and reduced the optimality gap in several other instances. However, the model proposed by Feliu et al. (2007) and Perboli et al. (2011) may not provide tight upper bounds for cases with more than two satellites. Since many real-world cases have more than two satellites, Jepsen et al. (2013) proposed an edge-flow based formulation, derived from the formulations of Capacitated Vehicle Routing Problem (CVRP) and Split-Depot Vehicle Routing Problem (SDVRP). Their formulation had fewer variables compared to the previous model by Perboli et al. (2011), but few of the constraint sets grew exponentially. To overcome this issue, the authors came up with a specialized branch rule for the branch-and-cut algorithm to find the optimal solution.

Most of the studies (Perboli et al., 2011; Crainic et al., 2011; Nguyen et al., 2012b) first decompose the 2E-CVRP to either CVRP or Multi-Depot CVRP and then solve using heuristic methods such as math-based heuristic (Perboli et al., 2011), multistart heuristic (Crainic et al., 2011), hybrid ant colony optimization (Meihua et al., 2011), Adaptive Large Neighborhood Search (ALNS) (Hemmelmayr et al., 2012), and Greedy Randomized Adaptive Search Procedure (GRASP) (Nguyen et al., 2012b). Among these ALNS, is the best performing heuristic (Cuda et al., 2015) for 2E-CVRP.

We briefly review the literature on three variants of 2E-CVRP. First, 2E-CVRP with the assumption that vehicles in the first echelon can also deliver to customers directly. Second, Two-Echelon Capacitated Location Routing Problem (2E-CLRP), which combines location and routing decisions on two levels. 2E-CLRP aims to minimize total cost (the sum of the fixed costs for opening the facilities, the usage cost of the vehicles, and the routing costs) by finding the optimal set of location sites for the depots and satellites as well as vehicle routes that satisfy the customer demands that do not violate the capacity constraints. Finally, the Truck Trailer Routing Problem (TTRP) that aims to identify the routes that can be visited by a truck alone, a truck with a trailer, and a combination of these two depending on the constraints.

There are only few studies on the 2E-CVRP variant with first level vehicles delivering to both satellites and customers. Escuín et al. (2012) is one of the first studies that allowed the first level vehicles to deliver to customers. Recently, a study by Abdulkader et al. (2018) followed a similar approach, and they called it vehicle routing in omni-channel distribution.

Boccia et al. (2010) formally introduced the 2E-CLRP. They proposed a Tabu Search (TS) based heuristic by extending the nested approach (Nagy and Salhi, 1996) and two-phase iterative approach (Tuzun and Burke, 1999) for LRPs. Later, Boccia et al. (2011) introduced three MILP formulations using one, two, and three index variables. Based on the computational experiments it was found that the three-index formulation outperformed the two-index formulation on medium-sized instances and provided better lower bounds. Similar to the 2E-CVRP, the 2E-LRP problem is divided into two LRPs, one at each echelon that enabled the application of algorithms developed for CLRP (Toyoglu et al., 2012). A branch-and-cut algorithm proposed by Contardo et al. (2012) solved 75 out of 147 instances to optimality. ALNS heuristic introduced by the same authors outperformed all the previous heuristics as it was able to improve 133 Best Known Solutions (BKS) out of 147. Schwengerer et al. (2012) presented a Variable Neighborhood Search (VNS) by extending the algorithm proposed for LRP. It was able to outperform heuristics proposed in Nguyen et al. (2012a,b), but on an average, it could not outperform ALNS (Boccia et al., 2011). Nguyen et al. (2012a,b) studied 2E-CLRP with a Single Depot, a special case of 2E-LRP, and proposed a two-index Integer Linear Programming (ILP) model and a GRASP complemented by a learning process with a path relinking procedure heuristic. The same authors proposed a MILP model and a multi-start Iterated Local Search (ILS) algorithm coupled with few simpler and less performing heuristics. However, based on computational experiments it is found that ALNS outperformed multi-start ILS (Nguyen et al., 2012a,b).

The MD2E-FMRP presented here does not include location decisions and related costs in the objective function for opening satellites/depots. However, it includes the decision whether or not to use a satellite for customer delivery. ALNS which has worked very well with 2E-CLRP problems and its variants are expected to perform well for the MD2E-FMRP.

Unlike the 2E-CVRP literature, heterogeneous fleets are well-studied in the VRP literature. The first paper which tackled TTRPs is by Semet and Taillard (1993). The term TTRP was first coined by Chao (2002) where a heuristic that requires an initial feasible solution that is improved using TS is proposed to handle Capacitated TTRP (CTTRP). Scheuerer (2006) proposed a heuristic comprising of two constructive heuristics and a TS algorithm to solve the CTTRP. The first constructive heuristic is called T-Cluster, which uses a cluster-based sequential insertion procedure where routes are constructed one-by-one until the vehicle is fully utilized. The second constructive heuristic is called T-Sweep, derived from the classical sweep algorithm followed by TS. This algorithm was able to outperform the heuristic introduced by Chao (2002). The Simulated Annealing based algorithm proposed by Lin et al. (2009) was able to outperform Chao (2002) and performed slightly better compared to Scheuerer (2006). This algorithm was later extended by Lin et al. (2010) to solve the Relaxed TTRP (where the limited fleet constraints are removed) and CTTRP with Time Windows (Lin et al., 2011). Villegas et al. (2011) proposed a hybrid meta-heuristic based on GRASP, VNS, and path relinking. Computational results showed that the GRASP/VNS with evolutionary path relinking has better performance on average compared to previous algorithms. Mirmohammadsadeghi and Ahmed (2015) extended this problem to include stochastic demands and proposed a memetic heuristic approach to solve the problem. Urban freight needs a wider range of vehicle types to cater to the demands efficiently. Our proposed MD2E-FMRP allows more than two classes of vehicles. Hence, we refer to our problem as MD2E-FMRP with heterogeneous fleets.

Most of the above studies considered distance minimization as their objective with a fixed cost. To the best of our knowledge, there are only two studies (Crainic et al., 2012; Soysal et al., 2015) that considered fuel minimization as an objective in two-echelon problems. Crainic et al. (2012) divided a day into eight time zones each with different speeds. They tested the model for all time zones and compared the fuel consumption for each time zone. However, their model did not consider variation in speed explicitly. To overcome this issue, Soysal et al. (2015) incorporated time-dependent speed in the second echelon for few links in specific time zones and assumed free-flow speeds for remaining links. Even in time-dependent routing problems, speed in a particular time zone is constant. In our study, driving cycle, which is a more realistic representation of speed and variations in driving speed, is used. The use of average speeds instead of driving cycle leads to underestimating fuel consumption (Kancharla and Ramadurai, 2018), and hence can alter optimal route plans.

3. Model Formulation

3.1. Problem statement

It is required to deliver goods from depots to customers through a set of satellites using heterogeneous fleets. In the first echelon, goods are delivered from the depot to the satellites by larger trucks. In the second echelon, goods are delivered from the satellites to the customers using smaller trucks. The demand of each customer is given and fixed, whereas the satellites' demands depend on the customers it serves. Each customer must be visited exactly once, but a satellite can be visited multiple times till the demand is satisfied. Additionally, not all satellites need to be used, and the decision whether or not to use a satellite is not known in advance. Each depot/satellite has a limited, but a different number of vehicles available for use, and, finally, we assume all vehicles use the same type of fuel. The objective is to minimize the total fuel cost.

3.2. Fuel consumption estimation

Fuel consumption depends on many parameters such as speed, acceleration, load, grade, and gravity. However, the traffic state restricts the ranges of speed and acceleration. We use the Comprehensive Modal Emission Model (CMEM) (Barth et al., 2004, 2005) for fuel consumption estimation. Earlier studies on VRP that have applied the CMEM include Bektas and Laporte (2011), Franceschetti et al. (2013), and Soysal et al. (2015). This model is selected because it takes into account all the parameters listed above. Moreover, this model can be applied for heavy-duty diesel vehicles as well as smaller pick-up trucks. CMEM consists of three modules, namely an engine power module, an engine speed module, and a fuel rate module (Barth et al., 2004, 2005).

Engine Power (P) requirement is calculated using:

$$P = \frac{(Wa + Wg\sin\theta + WgC_r\cos\theta + 0.5C_d\rho A v^2)v}{1000\epsilon} + P_{acc}, \tag{1}$$

where v is the speed (m/s), a is acceleration (m/s^2), W is the gross vehicle weight (kg), g is the gravitational constant (m/s^2), θ is the road grade angle in degrees, ρ is the air density $(kg/m^3,)$, A is the frontal surface area (m^2), C_d is the coefficient of aerodynamic drag, C_r the coefficient of rolling resistance, ϵ is the vehicle drive train efficiency, P is the second-by-second engine power output (kW), and P_{acc} is the engine power demand associated with running losses of the engine and the operation of vehicle accessories such as usage of air conditioning (we assumed it to be 0, following Demir et al. (2011)).

Engine speed (N) is interpolated between idle rpm and governing rpm using the speed of vehicle at that moment and then Fuel Rate (FR) is calculated as follows:

$$FR = \frac{\varphi(kNV + P/\eta)}{U} \tag{2}$$

where φ is fuel-to-air mass ratio, k is the engine friction factor, V is the engine displacement (in liters), η is a measure of indicated efficiency for diesel engines, and U is the lower heating value for the fuel. For ease in computing, equation (2) is simplified as follows.

Substituting (1) in (2)

$$FR = \frac{\varphi\left(kNV + \dfrac{(Wa + Wg\sin\theta + WgC_r\cos\theta + 0.5C_d\rho Av^2)v}{1000\epsilon\eta} + \dfrac{P_{acc}}{\eta}\right)}{U} \tag{3}$$

P_{acc} is assumed to be 0. Hence, (3) can be rewritten as follows.

$$FR = \frac{\varphi}{U}\left(kNV + \frac{0.5C_d\rho Av^3}{1000\epsilon\eta}\right) + W\frac{\varphi v}{U}\left(\frac{a + g\sin\theta + gC_r\cos\theta}{1000\epsilon\eta}\right) \tag{4}$$

$$FR = \alpha + \beta W, \tag{5}$$

where

$\alpha = \dfrac{\varphi}{U}\left(kNV + \dfrac{0.5C_d\rho Av^3}{1000\epsilon\eta}\right)$ is the weight independent part,

$\beta W = \dfrac{\varphi v}{U}\left(\dfrac{a + g\sin\theta + gC_r\cos\theta}{1000\epsilon\eta}\right)W$ is the weight dependent part, and

W is the total weight of vehicle including load carried and curb weight of vehicle.

Simplification of the above equation allows for preprocessing the matrices of α and β using the driving cycle data for each link, thus reducing the run-time. The only unknown data W (weight) is a variable in the problem formulation.

3.3. Why driving cycle?

Is average speed good enough to estimate fuel consumption or does use of a driving cycle improve the accuracy? To investigate this, the amount of fuel consumed by different fully loaded trucks for traversing a length of 7 km with zero gradient is computed using US EPA driving cycle (LA4 and heavy-duty urban dynamometer driving schedule) and average speed of driving cycle. When average speed is used instead of variable speed, the fuel consumption estimated for a small truck ($<$ 2 tonnes), medium truck (3 tonnes), and heavy truck ($>$5 tonnes) is less by approximately 70%, 130%, and 160%, respectively. The variation in fuel consumption may have little effect on the optimal solution for a homogeneous fleet problem. However, for a heterogeneous fleet problem with fuel-minimizing objective, the effect will be significant. Therefore, the use of average speed will lead to sub-optimal results (Kancharla and Ramadurai, 2018).

3.4. Description of model

This section presents the four-index mixed integer linear programming formulation of the MD2E-FMRP. The present model is a multi-commodity flow based formulation that accounts for multiple depots and heterogeneous fleets in both echelons. The formulation for the second echelon is a straightforward extension of an earlier formulation with an additional index for vehicle types. However, the first echelon is significantly different as the present problem allows the deliveries to be split from different depots using a heterogeneous fleet, whereas earlier formulations in literature had a single depot and homogeneous fleet.

We define this problem on a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V} = \{\mathbf{T}, \mathbf{S}, \mathbf{C}\}$ is the set of vertices containing depots ($\mathbf{T}$), satellites ($\mathbf{S}$), and customers ($\mathbf{C}$). $\mathbf{E} = \{\mathbf{E}_1, \mathbf{E}_2\}$ is the set of edges, the first subset ($\mathbf{E}_1 = \{(i,j) : i,j \in (\mathbf{T} \cup \mathbf{S}) \times (\mathbf{T} \cup \mathbf{S})$ and $i \neq j\}$) has edges belonging to the first echelon, while the second subset ($\mathbf{E}_2 = \{(k,l) : k,l \in (\mathbf{S} \cup \mathbf{C}) \times (\mathbf{S} \cup \mathbf{C})$ and $k \neq l\}$) belongs to the second echelon edges. $\mathbf{H}$ is the set of vehicle types available in the first echelon and $\mathbf{M}$ is the set of vehicles available in the second echelon. The fleet size of vehicle type $h (\in \mathbf{H})$ at each depot $t (\in \mathbf{T})$ is γ_t^h and vehicle type $m (\in \mathbf{M})$ at each satellite $s (\in \mathbf{S})$ is δ_s^m. Capacities and curb weight of a first echelon vehicle are ζ_h and $\psi_h, h \in \mathbf{H}$ respectively. Similarly, ζ_m and $\psi_m, m \in \mathbf{M}$ are for the second echelon. α_{ij}^h, is the constant and β_{ij}^h is the coefficient of weight in the fuel consumption function (Eq. 5) for edge $(i,j) \in \mathbf{E}_1$ using vehicle type h in the first echelon. Similarly, α_{kl}^m, β_{kl}^m are the co-efficients for edge $(k,l) \in \mathbf{E}_2$ using vehicle type m in the second echelon. Length of the first echelon edge $(i,j) \in \mathbf{E}_1$ is λ_{ij}, similarly the length of the second echelon edge $(k,l) \in \mathbf{E}_2$ is λ_{kl}. d_c is the demand at a customer c $\in \mathbf{C}$ and it is assumed that $d_c \geq 1$. ξ is the unit fuel cost.

The variables corresponding to the first echelon are as follows: $x_{ij}^{th} \in \mathbb{Z}^+$ is the number of vehicles of type $h \in \mathbf{H}$ starting from depot $t \in \mathbf{T}$ and traversing edge $(i,j) \in \mathbf{E}_1$; and $Q_{ij}^{th} \in \mathbb{R}^+$ is the quantity of freight carried on edge $(i,j) \in \mathbf{E}_1$ by a vehicle of type $h \in \mathbf{H}$ starting from depot $t \in \mathbf{T}$. For the second echelon, the variables are as follows: $y_{kl}^{sm} \in \{0,1\}$ is equal to 1 iff a vehicle of type $m \in \mathbf{M}$ starting at satellite $s \in \mathbf{S}$ and going on edge $(k,l) \in \mathbf{E}_2$ and $Q_{kl}^{sm} \in \mathbb{R}^+$ is the quantity of freight carried on edge $(k,l) \in \mathbf{E}_2$ by a vehicle of type $m \in \mathbf{M}$ starting from a satellite $s \in \mathbf{S}$. Let X, Y, and Q be the solution vectors of x_{ij}^{th}, y_{kl}^{sm}, and Q_{ij}^{th} and Q_{kl}^{sm} respectively.

MD2E-FMRP is formulated thus:

$$f(X^*, Y^*, Q^*) = min \sum_{t \in \mathbf{T}} \sum_{h \in \mathbf{H}} \sum_{(i,j) \in \mathbf{E}_1} ((\alpha_{ij}^h + \beta_{ij}^h \psi_h) x_{ij}^{th} + \beta_{ij}^h Q_{ij}^{th}) \xi \quad + \tag{6a}$$

$$\sum_{s \in \mathbf{S}} \sum_{m \in \mathbf{M}} \sum_{(k,l) \in \mathbf{E}_2} ((\alpha_{kl}^m + \beta_{kl}^m \psi_m) y_{kl}^{sm} + \beta_{kl}^m Q_{kl}^{sm}) \xi \tag{6b}$$

The objective function comprises two parts. Part 1 (6a) is fuel cost in first echelon where $(\alpha_{ij}^h + \beta_{ij}^h \psi_h) x_{ij}^{th}$ is the fuel consumed by all vehicles of type h from depot t passing through edge (i,j) if they were running empty and $\beta_{ij}^h Q_{ij}^{th}$ is the additional fuel consumed due to the load carried by the vehicle. Similarly, Part 2 (6b) is fuel cost for the second echelon. Objective is to minimize total fuel cost in both levels subject to

Notations	
$\mathbf{T} = \{1,2,\cdots,t,\cdots\}$	Set of depots
$\mathbf{S} = \{1,2,\cdots,s,\cdots\}$	Set of satellites
$\mathbf{C} = \{1,2,\cdots,c,\cdots\}$	Set of customers
$\mathbf{H} = \{1,2,\cdots,h,\cdots\}$	Set of vehicle types available in the first echelon
$\mathbf{M} = \{1,2,\cdots,m,\cdots\}$	Set of vehicle types available in the second echelon
$\mathbf{E}_1$	Set of edges in the first echelon $\{(i,j) : i,j \in (\mathbf{T} \cup \mathbf{S}) \times (\mathbf{T} \cup \mathbf{S})$ and $i \neq j\})$
$\mathbf{E}_2$	Set of edges in the second echelon $\{(k,l) : k,l \in (\mathbf{S} \cup \mathbf{C}) \times (\mathbf{S} \cup \mathbf{C})$ and $k \neq l\})$
γ_t^h	Number of the first echelon vehicles of type $h \in \mathbf{H}$ available at depot $t \in \mathbf{T}$
δ_s^m	Number of the second echelon vehicles of type $m \in \mathbf{M}$ available at satellite $s \in \mathbf{S}$
ζ_h	Capacity of vehicle type $h \in \mathbf{H}$
ζ_m	Capacity of vehicle type $m \in \mathbf{M}$
α_{ij}^h	Constant in fuel consumption equation for first echelon vehicle ($h \in \mathbf{H}$) when traveling on edge $(i,j) \in \mathbf{E}_1$
β_{ij}^h	Coefficient of weight in fuel consumption equation for first echelon vehicle ($h \in \mathbf{H}$) when traveling on edge $(i,j) \in \mathbf{E}_1$
α_{kl}^m	Constant in fuel consumption equation for second echelon vehicle ($m \in \mathbf{M}$) when traveling on edge $(k,l) \in \mathbf{E}_2$
β_{kl}^m	Coefficient of weight in fuel consumption equation for second echelon vehicle ($m \in \mathbf{M}$) when traveling on edge $(k,l) \in \mathbf{E}_2$
x_{ij}^{th}	Total number of vehicles of type $h \in \mathbf{H}$ starting from depot $t \in \mathbf{T}$ and traversing edge $(i,j) \in \mathbf{E}_1$
y_{kl}^{sm}	Binary variable equal to 1 if the second echelon edge $(k,l) \in \mathbf{E}_2$ is used by vehicle $m \in \mathbf{M}$ starting from satellite $s \in \mathbf{S}$
Q_{ij}^{th}	Quantity of freight through edge $(i,j) \in \mathbf{E}_1$ by vehicle type $h \in \mathbf{H}$ starting from depot $t \in \mathbf{T}$
Q_{kl}^{sm}	Quantity of freight through edge $(k,l) \in \mathbf{E}_2$ by vehicle type $m \in \mathbf{M}$ starting from satellite $s \in \mathbf{S}$
w_{ts}^h	Demand of satellite $s \in \mathbf{S}$ satisfied by depot $t \in \mathbf{T}$ using vehicle $h \in \mathbf{H}$
λ_{ij}	Length of the edge $(i,j) \in \mathbf{E}_1$
λ_{kl}	Length of the edge $(k,l) \in \mathbf{E}_2$
ξ	Fuel cost per liter
ψ_h	Curb weight of vehicle type $h \in \mathbf{H}$
ψ_m	Curb weight of vehicle type $m \cup \mathbf{M}$
d_c	Demand of freight at customer $c \in \mathbf{C}$ ($d_c > 1$)
Ω	Large positive integer

constraints (7)-(27).

$$\sum_{j \in \mathbf{S}} x_{tj}^{th} \leq \gamma_t^h \quad \forall\, t \in \mathbf{T}, h \in \mathbf{H} \tag{7}$$

$$\sum_{j \in \mathbf{S}} x_{pj}^{ph} = \sum_{t \in \mathbf{T}} \sum_{j \in \mathbf{S}} x_{jp}^{th} \quad \forall\, p \in \mathbf{T}, h \in \mathbf{H} \tag{8}$$

$$\sum_{\substack{i \in \mathbf{S} \cup \mathbf{T} \\ i \neq s}} x_{is}^{th} = \sum_{\substack{i \in \mathbf{S} \cup \mathbf{T} \\ i \neq s}} x_{si}^{th} \quad \forall\, s \in \mathbf{S}, t \in \mathbf{T}, h \in \mathbf{H} \tag{9}$$

$$\sum_{t \in \mathbf{T}} \sum_{\substack{i \in \mathbf{S} \cup \mathbf{T} \\ i \neq s}} Q_{is}^{th} - \sum_{t \in \mathbf{T}} \sum_{\substack{i \in \mathbf{S} \cup \mathbf{T} \\ i \neq s}} Q_{si}^{th} = \sum_{t \in \mathbf{T}} w_{ts}^h, \quad \forall\, s \in \mathbf{S}, h \in \mathbf{H} \tag{10}$$

$$\sum_{h\in\mathbf{H}}\sum_{t\in\mathbf{T}}\sum_{j\in\mathbf{S}} Q_{jp}^{th} = 0, \ \ \forall\, p \in \mathbf{T} \tag{11}$$

$$x_{ij}^{th} \leq \Omega Q_{ij}^{th} \ \ \forall\, t \in \mathbf{T}, h \in \mathbf{H}, (i,j) \in \mathbf{E}_1 \tag{12}$$

$$Q_{ij}^{th} \leq \zeta_h x_{ij}^{th} \ \ \forall\, t \in \mathbf{T}, h \in \mathbf{H}, (i,j) \in \mathbf{E}_1 \tag{13}$$

$$\sum_{s\in\mathbf{S}} Q_{ts}^{th} = \sum_{s\in\mathbf{S}} w_{ts}^{h}, \ \ \forall\, t \in \mathbf{T}, h \in \mathbf{H} \tag{14}$$

Constraints (7)-(9) are for the first echelon. Constraints (7) ensure that for each vehicle type h, the number of vehicles starting from a depot t, are less than the maximum available vehicles at that depot. Constraints (8) allow vehicles to end at any depot and at the same time ensure that total vehicles entering a depot is equal to the total vehicles leaving form the depot. Constraints (9) ensure the number of vehicles entering and leaving any satellite in the first echelon are equal. Constraints (10) conserve demand at satellites. Constraints (11) ensure vehicles returning to a depot are empty. Constraints (10) - (11) also determine the load on each link. Constraints (12) ensure that quantity of freight carried and number of vehicles from a depot are jointly positive or jointly zero. Constraints (13) impose vehicle capacity restrictions. Constraints (14) ensure the total load carried from a depot equals the load delivered from the depot and these constraints along with constraints (12) and (13) help in sub-tour elimination.

$$\sum_{t\in\mathbf{T}}\sum_{h\in\mathbf{H}} w_{ts}^{h} = \sum_{m\in\mathbf{M}}\sum_{k\in\mathbf{S}\cup\mathbf{C}}\sum_{c\in\mathbf{C}} y_{kc}^{sm} d_c, \ \ \forall\, s \in \mathbf{S} \tag{15}$$

Constraints (15) couple both echelons. It allows a first level route to connect to a satellite only if there are second level routes starting from that satellite simultaneously ensuring demand satisfaction at the satellite.

$$\sum_{l\in\mathbf{C}} y_{sl}^{sm} \leq \delta_s^m \ \ \forall\, s \in \mathbf{S}, m \in \mathbf{M} \tag{16}$$

$$\sum_{l\in\mathbf{C}} y_{pl}^{pm} = \sum_{s\in\mathbf{S}}\sum_{l\in\mathbf{C}} y_{lp}^{sm} \ \ \forall\, p \in \mathbf{S}, m \in \mathbf{M} \tag{17}$$

$$\sum_{\substack{k\in\mathbf{S}\cup\mathbf{C}\\ k\neq c}} y_{kc}^{sm} = \sum_{\substack{k\in\mathbf{S}\cup\mathbf{C}\\ k\neq c}} y_{ck}^{sm} \ \ \forall\, c \in \mathbf{C}, s \in \mathbf{S}, m \in \mathbf{M} \tag{18}$$

$$\sum_{s\in\mathbf{S}}\sum_{m\in\mathbf{M}}\sum_{\substack{k\in\mathbf{S}\cup\mathbf{C}\\ k\neq c}} Q_{kc}^{sm} - \sum_{s\in\mathbf{S}}\sum_{m\in\mathbf{M}}\sum_{\substack{k\in\mathbf{S}\cup\mathbf{C}\\ k\neq c}} Q_{ck}^{sm} = d_c, \ \ \forall\, c \in \mathbf{C} \tag{19}$$

$$\sum_{m\in\mathbf{M}}\sum_{s\in\mathbf{S}}\sum_{k\in\mathbf{C}} Q_{kp}^{sm} = 0, \ \ \forall\, p \in \mathbf{S} \tag{20}$$

$$Q_{sl}^{sm} \leq \zeta_m y_{sl}^{sm} \ \ \forall\, l \in \mathbf{C}, s \in \mathbf{S}, m \in \mathbf{M} \tag{21}$$

$$\sum_{m\in\mathbf{M}}\sum_{s\in\mathbf{S}}\sum_{k\in\mathbf{S}\cup\mathbf{C}} y_{kc}^{sm} = 1, \ \ \forall\, c \in \mathbf{C} \tag{22}$$

$$x_{ij}^{th} \in \mathbb{Z}^{+} \;\; \forall\, t \in \mathbf{T}, h \in \mathbf{H}, (i,j) \in \mathbf{E}_1 \tag{23}$$

$$y_{kl}^{sm} \in \{\, 0, 1\} \;\;\; \forall\, s \in \mathbf{S}, m \in \mathbf{M}, (k,l) \in \mathbf{E}_2 \tag{24}$$

$$w_{tj}^{h} \in \mathbb{R}^{+}, \;\; \forall\, t \in \mathbf{T}, h \in \mathbf{H}, j \in \mathbf{S} \tag{25}$$

$$Q_{ij}^{th} \in \mathbb{R}^{+}, \;\; \forall\, t \in \mathbf{T}, h \in \mathbf{H}, (i,j) \in \mathbf{E}_1 \tag{26}$$

$$Q_{kl}^{sm} \in \mathbb{R}^{+}, \;\; \forall\, s \in \mathbf{S}, m \in \mathbf{M}, (k,l) \in \mathbf{E}_2 \tag{27}$$

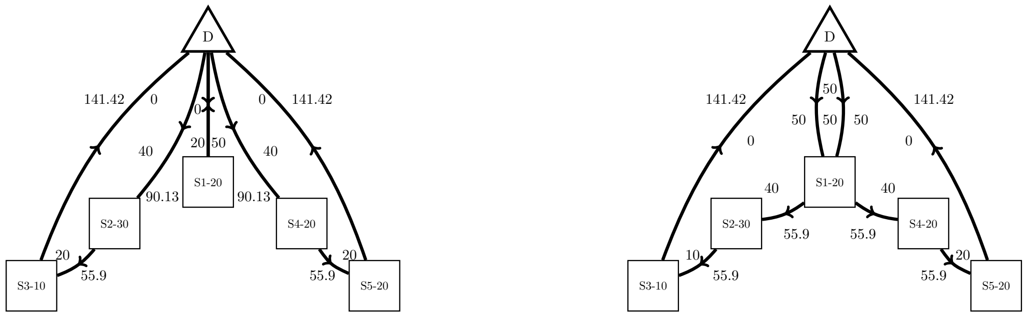
Constraints (16)-(18) are for the second echelon. Constraints (16) ensure that for each vehicle type m, the number of vehicles starting from a satellite are less than the maximum available vehicles at that satellite. Constraints (17) allow vehicles to end at any satellite and at the same time ensure that the total vehicles entering a satellite is equal to the total vehicles leaving from the satellite. Constraints (18) ensure flow conservation at each customer. Constraints (19) conserve demand at each customer and also help in sub-tour elimination. Constraints (20) ensure vehicles returning to a depot are empty. Constraints (19) and (20) also determine the load on each link. Constraints (21) impose vehicle capacity restrictions at second level routes. Constraints (22) allow a customer to connect to only one satellite. Constraints (23)-(27) specify the domain of the variables. If the objective is to minimize the total distance traveled, the objective function (equation 6a, 6b) is changed to equations (28a), (28b).

$$f'(X^*, Y^*) = min \sum_{t\in\mathbf{T}} \sum_{h\in\mathbf{H}} \sum_{(i,j)\in\mathbf{E}_1} \lambda_{ij} x_{ij}^{th} \;\; + \tag{28a}$$

$$\sum_{s\in\mathbf{S}} \sum_{m\in\mathbf{M}} \sum_{(k,l)\in\mathbf{E}_2} \lambda_{kl} y_{kl}^{sm} \tag{28b}$$

3.5. Split-deliveries and transshipment

Our formulation allows split-deliveries in the first echelon; multiple vehicles can visit a satellite. Allowing split-deliveries can lead to better solutions. For example, consider a case with one depot and five satellites as shown in Figure 2. The distance between the depots and satellites are shown above each arc. The satellites are numbered S1 through S5 and let us assume the demand at a satellite is known (shown beside the satellite number). The number below each arc is the load carried by the vehicle on that arc. Figure 2a is the optimal solution without split deliveries. The total distance traveled is 674.924 units. Allowing split deliveries (Figure 2b) reduces the distance traveled to 606.450 units. However, split-deliveries introduces additional complexities. When two vehicles visit the same location there is a possibility of freight exchange between them - also referred to as transshipment. Here, constraints (14) prevent transshipment. Considering transshipment requires altering the overall cost in the objective function.



(a) Without split-deliveries (b) With split-deliveries

Figure 2: Different cases for split-deliveries

4. Heuristic Solution Algorithm for MD2E-FMRP

MD2E-FMRP is solved using a hybrid heuristic that uses Adaptive Large Neighborhood Search (ALNS) algorithm for both echelons. ALNS, introduced by Ropke and Pisinger (2006), is one of the most effective heuristics that can handle large-scale vehicle routing problems. Below we describe the algorithm implementation in detail.

4.1. Initial solution

We obtain an initial solution by solving a bin-packing problem for the second echelon, and a greedy algorithm for the first echelon. The advantage of solving as a bin-packing problem over other algorithms such as Sweep algorithm (Renaud and Boctor, 2002) or Clarke and Wright savings algorithm (Clarke and Wright, 1964) is the number of vehicles required is minimized as well. A limitation of ALNS implementations in previous studies (Hemmelmayr et al., 2012) is it cannot implicitly minimize the number of vehicles required.

Best fit allocation based bin packing approach is used to find the customer allocation list with a minimum number of vehicles. Each of these vehicles is then randomly allotted such that there will be at least one vehicle used for each satellite ensuring no satellite is left out in the initial solution. We remove satellites that are "expensive" in the later stages of ALNS.

An initial solution for the first echelon is found using a greedy algorithm. For the cases with demand more than the capacity of the vehicle, demand is split among multiple vehicles till the demand is satisfied. The initial solution does not restrict the vehicle types in each echelon, and they are altered as the solution evolves. Algorithm 1 illustrates the initial solution construction procedure.

4.2. Overview of ALNS

While ALNS has been used in the past for solving vehicle routing problems, the present implementation has few new challenges. The number of open satellites and their demands are unknown. Further, multiple depots can serve these satellites. Allowing a heterogeneous fleet mix also adds another decision dimension to

Algorithm 1 Greedy algorithm for initial solution

Initialize V_c^x based on γ_t^h, δ_s^m, $\chi = C$ ▷ V_c^x - list of vehicles with capacity 'c' in echelon 'x'
$Cap = \max(\zeta_M)$
while $\chi \neq 0$ **do** ▷ χ - unassigned customers
    $RS \leftarrow$ Set of routes obtained with best fit bin packing with Cap as bin size
    for rou in RS **do**
        if $V_{Cap}^2 > 0$ **then**
            Assign rou to a satellite having vehicle with capacity Cap
            Remove customers in rou from χ
            Remove one vehicle from V_{Cap}^2
        end if
    end for
    if U $\neq 0$ **then**
        Remove Cap from ζ_M and $Cap = \max(\zeta_M)$
    end if
end while
Calculate satellite demands based on the second echelon solution generated in previous step (Sd)
$Cap = \max(\zeta_H)$
for sat in Sd **do**
    for veh in V_{Cap}^1 **do**
        $ac = Cap$ - veh_d ▷ ac- Available capacity; veh_d -demand catered by vehicle (veh)
        if ($sat > ac$) and ($ac \neq 0$) **then**
            Assign the satellite to vehicle (veh)
            Increase veh_d by ac and decrease sat by ac
        else if ($sat <= ac$) and ($ac \neq 0$) **then**
            Increase veh_d by sat
            Assign the satellite to the vehicle (veh)
        end if
        if $veh_d = Cap$ **then**
            Remove veh from V_{Cap}^1
        end if
        if $V_{Cap}^1 < 1$ **then**
            Remove Cap from ζ_H and $Cap = \max(\zeta_H)$
        end if
    end for
end for

solving the problem. To overcome these challenges, the second-echelon problem that determines the number of open satellites and the exact demands of these satellites is solved first.

The second echelon problem can be considered as equivalent to a Location Routing Problem with Heterogeneous vehicles (LRP-HV) that identifies the optimal location of satellites and the routes from these satellites simultaneously. The first echelon problem can be considered equivalent to a Multi-Depot Split Deliveries Capacitated Vehicle Routing Problem with Heterogeneous vehicles (MDSDCVRP-HV). Even though the number of satellites are less, the presence of multiple depots, split-deliveries, and heterogeneous vehicles motivated the use of ALNS in this case as well.

Algorithm 2 illustrates the implementation of the algorithm for the second echelon. The algorithm for first echelon is very similar and hence not repeated. We start with preprocessing of the inputs required for cost calculation, and then obtain an initial solution for both the echelons. We apply removal operators on the initial solution that are selected using the roulette wheel selection process. These operators remove **p** number of customers from the current solution and keep them in the customer pool, that are used later in the Insertion stage. We follow the hierarchical structure of removal operators developed by Hemmelmayr et al. (2012). The first level removal operators will have a small impact on the first echelon problem as the number of open/closed satellites remains the same. Whereas, the second level removal operators have a high impact on the first echelon solution because these operators can open or close any satellite. Second level operators will be called only if there is no improvement for **N** number of iterations. We use the record-to-record acceptance criteria, which accepts a new solution if the objective function value is no worse than κ % of the current solution or if the obtained solution is from a second level operator. We do not restrict the search space of the algorithm to the space of feasible solutions; the algorithm is allowed to violate the maximum number of available vehicles, vehicle capacity, and capacity at satellites constraints but imposes a hefty penalty to eliminate these violations.

4.3. Removal operators

We use four low impact removal operators (random removal, worst removal, related removal, and route removal) and three high impact removal operators (satellite closing, satellite opening, and satellite swap), which were introduced in Ropke and Pisinger (2006) and Hemmelmayr et al. (2012). These operators destroy the solution by removing **p** (min ($0.4n_c$, 60), where n_c is the total number of customers) customers from the solution and add them to the customer pool. We present a detailed description of these operators below.

Random Removal

This operator removes the **p** customers randomly for all the routes and adds them to the customer pool.

Worst Removal

This operator removes the **p** customers who have the highest removal gains. Removal gain is the difference in the cost with and without the customer in the solution. Subsequently, we normalize the removal gains by dividing it with the average cost of all the incoming edges to the corresponding node. The normalization is done to avoid repeatedly choosing far away customers. Further, to randomize the search, the removal gains are multiplied by a random factor $d \in [0.8, 1.2]$.

Related Removal

This operator first selects a seed customer randomly from the list of all the available customers and removes it from the present route and adds to the customer pool. Remaining **p-1** customers, which are closest to the seed customer are removed from their existing routes and stored in the customer pool.

Route Removal

This operator selects a random route from the available routes and all the customers in that route are

Algorithm 2 Adaptive Large Neighborhood Search

Read input data, initialize weights and scores
Generate an initial solution (X, Y, Q) using Algorithm 1
Make initial solution as best solution $(X^*, Y^*, Q^*) \leftarrow (X, Y, Q)$
Make initial objective as best objective $f(X^*, Y^*, Q^*) \leftarrow f(X, Y, Q)$
Sd - satellite demand configuration; PSD_i - unique satellite demand configurations found till iteration i
j = 0
for $i \leftarrow 0, MaxIterations$ **do**
    if j = N **then** ▷ N - Number of iterations allowed without improvement
        Call high impact Removal operator $((X, Y, Q)_{ip})$ ▷ $(X, Y, Q)_{ip}$ - Partial solution at iteration i
    else
        Call low impact Removal operator $((X, Y, Q)_{ip})$
    end if
    Call Insertion operator $((X, Y, Q)_i)$
    if Sd in PSD_i list **then**
        The first echelon solution is retrieved from memory
    else
        ALNS is called to get the solution for the first echelon
    end if
    if $f(X, Y, Q)_i \leq (1+\kappa) f(X, Y, Q)$ **then** ▷ κ - is a record-to-record acceptance factor
        $(X, Y, Q) \leftarrow (X, Y, Q)_i$
        j $\leftarrow$ 0 and update score
    else if j = N **then**
        $(X, Y, Q) \leftarrow (X, Y, Q)_i$
        j $\leftarrow$ 0 and update score
    else
        j += 1
    end if
    if $f(X, Y, Q) < f(X^*, Y^*, Q^*)$ **then**
        $(X^*, Y^*, Q^*) \leftarrow (X, Y, Q)$
    end if
    if i % Q = 0 **then** ▷ Q - Weights update interval
        Update weights based on scores
    end if
end for

removed and stored in the customer pool.

Satellite Opening

This operator randomly opens a satellite from the list of closed satellites. Then, **p** closest customers to the newly opened satellite are removed from their existing routes and stored in the customer pool.

Satellite Closing

This operator randomly closes a satellite from the list of open satellites. Then all the customers connected to the newly closed satellite are added to the customer pool. If the removed satellite is the only open satellite, a new satellite is opened to avoid closing all the satellites.

Satellite Swap

This operator is similar to the satellite closing operator. However, here a new satellite is opened every time based on a roulette wheel selection. The probability used in roulette wheel selection is inversely proportional to the distance from the satellite being closed. This helps in swapping the satellites, which are closer and does not change the solution drastically.

4.4. Insertion operators

These operators insert the customers from the customer pool into any of the existing routes or new routes.

Greedy Insertion

We calculate the insertion cost (increase in objective function by inserting the customer in a route) for all the customers in the customer pool. A random customer from the customer pool is selected and inserted at the lowest insertion cost position, and we recalculate insertion costs for the remaining customers. This operation can also insert customers by opening a new route. The new route cannot be the one that is removed by random route removal in the present iteration.

Greedy Insertion Perturbation

This operator is same as the Greedy Insertion operator with the only difference being the insertion costs are perturbed by a random factor $d \in [0.8, 1.2]$ to introduce randomness in the search.

Regret k-Insertion

Regret cost, the cost difference between the best insertion cost and the k^{th} best insertion cost, is calculated for all the customers in the customer pool. We treat the customers in the descending order of their regret cost. This operator uses look-ahead information (till k^{th} position), thus preventing situations where a customer has to be inserted in an inferior position due to unavailability of better positions. Upon inserting a customer at a position, we recompute the regret costs for the remaining unplaced customers.

4.5. Adaptive weights

We divide total iterations into m segments of 50 consecutive iterations. Weights of the operators are updated based on the scores obtained at the end of every segment. We set the scores to zero at the start of every segment, and then increment by a value in every iteration based on the performance of operators. Scores are incremented based on the following procedure: 50 is added to the score if the incumbent solution is a new global best solution; 20 if it is a new improving solution; 10 if it is an accepted deteriorating solution; otherwise, the score remains unchanged. The average score at the end of each segment is used to update the weight of the corresponding operator. We select the operators by a roulette wheel mechanism in each iteration. The selection probability of operator i is $p_i = w_i / \sum_{i \in OP} w_i$, where OP is the set of removal or insertion operators and w_i stands for the weight of the operator i.

5. Computational Tests

In this section, we evaluate the computational efficiency and solution quality of the model in different instances. The MILP model is coded in GAMS 23.9 and solved using Gurobi 6.9 solver hosted on NEOS server (Czyzyk et al., 1998; Dolan, 2001; Gropp and Moré, 1997). We ran all the instances on neo-server-3 or neo-server-5 that had a configuration of Intel Xeon E5-2430 @ 2.2GHz with 3 GB RAM (installed RAM is 64 GB, however there is a 3 GB limit for each job). The ALNS algorithm is coded in Python and tested on a PC with 2.2-GHz Xeon processor with 8 GB of RAM. Parameters used in the study are described in section 5.1 followed by driving cycle data in section 5.2, instances description in section 5.3, and finally, computational results in section 5.4.

5.1. Parameters

Table 1 (adopted from Demir et al. (2011)) shows description and values of the parameters mentioned in section 3.2, and Table 2 provides vehicle specific values of different trucks utilized in the test instance (light trucks - Demir et al. (2011), medium truck - Force (2018), medium heavy truck - Eicher (2018), heavy truck - TATA (2018)).

Table 1: Description of parameters used in CMEM

Parameter	Description	Value used
v	Speed in m/s	Driving cycle
a	Acceleration in m/s^2	Driving cycle
N	Engine speed	approx. from speed
W	Gross vehicle weight in kg	Twice the vehicle capacity
g	Gravitational constant m/s^2	9.81
θ	Road grade angle in degrees	0
ρ	Air density in kg/m^3	1.2041
ϵ	Vehicle drive train efficiency	0.4
φ	Fuel-to-air mass ratio	0.0667
k	Engine friction factor	0.2
η	Efficiency for diesel engines	0.45
U	Lower heating value for the fuel in MJ/kg	44 for diesel
A	Frontal surface area in m^2	See table 2
C_d	Coefficient of aerodynamic drag	See table 2
C_r	Coefficient of rolling resistance	See table 2
V	Engine displacement in liters	See table 2

5.2. Driving cycle data

The driving cycles used in the study are standard US EPA driving cycles (USEPA (2017), see Figures 3 and 4) and the Table 3 shows the driving cycle parameters. We repeat the same driving cycle and truncate the last driving cycle such that it exactly spans the length of the link. The average speed for each link is obtained by averaging the speeds from the driving cycle fitted to a link.

Table 2: Vehicle specific parameters

Vehicle type	**A**	**Cd**	**Cr**	**V**	**Idle rpm**	**Governing rpm**	**Capacity (ton)**
Light truck	2.25	0.57	0.01	0.7	800	3600	1.5
Medium truck	2.51	0.6	0.045	2.1	700	2800	2
Medium heavy truck	2.81	0.8	0.045	3.7	600	2600	3
Heavy truck	3.341	0.9	0.07	5.83	540	2400	6

To further increase the accuracy of fuel consumption estimation, specific driving cycles that depend on the link type and congestion level at different times of the day are required. Such driving cycles, if available, can be incorporated into the preprocessing stage of the proposed model.

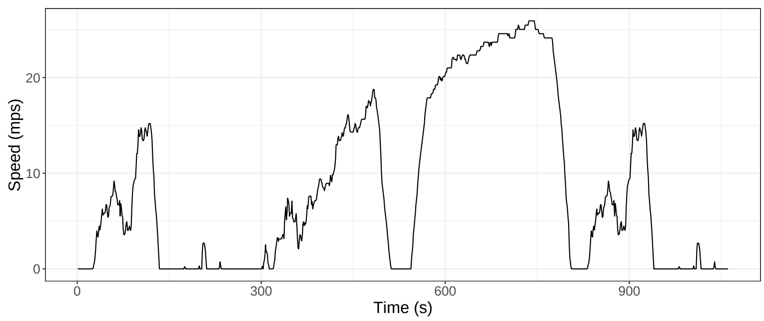


Figure 3: USEPA urban driving cycle for heavy duty vehicles

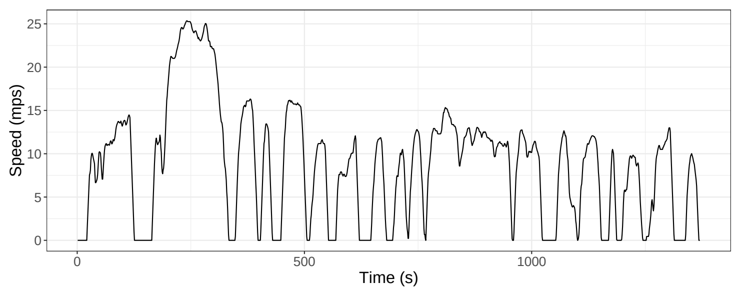


Figure 4: USEPA urban driving cycle for light duty vehicles

Table 3: Driving Cycle (DC) parameters

Parameter	**Light duty DC**	**Heavy duty DC**
Average speed (m/s)	8.75	8.42
Average running speed (m/s)	10.79	12.62
Average acceleration (m/s^2)	0.50	0.48
Average deceleration (m/s^2)	0.58	0.58
Root mean square acceleration (m/s^2)	0.63	0.47
Percentage of time in acceleration mode	39.71	27.05
Percentage of time in deceleration mode	34.67	22.62
Percentage of time in idling & creeping mode	25.62	50.33
Percentage of time in cruising mode	0.00	0.00

5.3. *Instance sets*

We test the model on 2E-CVRP instances introduced by Perboli et al. (2011) (set-2, set-3, and set-4), modified instances (37 to 54) from set-4 with an extra depot, and 8 new instances introduced for MD2E-FMRP. Both set-2 and set-3 instances have 22 or 33 customers with two satellites and a central depot or 51 customers with two or four satellites and a central depot, whereas set-4 is a more extensive network with a central depot, two to five satellites, and 50 customers. The main differences between set-2 and set-3 is the location of satellites and the central depot, and set-3 does not have instances with four satellites. The new instances mimic reality with depot location selected randomly along the outer boundary in a space enclosed by a 30 km $\times$ 30 km region excluding the central 25 km $\times$ 25 km region. Customers and satellites are selected randomly in the central 25 km $\times$ 25 km region. The first echelon has two vehicle types: a heavy vehicle (6 ton) and a medium-heavy vehicle (3 ton). In the second echelon, a light vehicle (1.5 ton) and a medium vehicle (2 ton) are available. The number of vehicles available at each depot and satellite can be different. Hence, there can a be a mix of heavy and medium-heavy vehicles at each depot, and light and medium vehicles at each satellite. We use the US EPA driving cycle for heavy-duty vehicles for the first echelon vehicles and light-duty vehicle driving cycle for the second echelon vehicles.

5.4. *Computational results of Gurobi*

New instances are solved based on the proposed formulation (6a)-(27) and the instances in literature are solved based on the constraints (7)-(27) with distance minimization objective (equation 28a, 28b). Distance minimization is considered to show the savings in distance because of the relaxation (constraints (8) and (17)) that vehicles can end at any depot/satellite.

All the instances are run using Gurobi with a time limit of 10,000s. Gurobi solved all instances with 22 customers to optimality, and the instances with 33 customers had an average gap of less than 1% compared to the best bound (obtained by Gurobi after linear relaxation). We test the present model with set-2 and set-3 instances with the restriction that every vehicle has to come back to the same satellite. Tables 4 and 5 show that Gurobi can find the optimal solution in 14 out of 21 instances and two more solutions are the same as the Best Known Solution (BKS) (Perboli et al., 2011; Hemmelmayr et al., 2012). Similarly, in set-3, Gurobi found the optimal solutions in 9 out of 18 instances; one solution is better than BKS, and three are the same as BKS. In the case with relaxation that vehicles can end at any depot (satellite) and yet conserve vehicles at each depot (satellite), there is an additional distance saving in 11 instances in set-2 and 14 instances in set-3. Distance reduction is up to 8.06% and 7.69% with an average of 1.13% and 1.3% for set-2 and set-3 respectively. Optimal values in the results are in bold font.

The new instances are tested with and without the relaxation to return to any depot (satellite) for both fuel and distance minimization separately and these results are summarized in Table 6. Here, Gurobi was able to find optimal solutions in the first two cases within the time limit of 10,000s and the remaining

Table 4: Results of set-2 instances

Instances	BKS (i) (Same depot)	Present model			
		Same depot (ii)	% dev between (i) and (ii)	Different depot (iii)	% dev between (ii) and (iii)
E22-K4-S06-17	**417.07**	**417.07**	0.00	**416.85**	-0.05
E22-K4-S08-14	**384.96**	**384.96**	0.00	**384.96**	0.00
E22-K4-S09-19	**470.60**	**470.60**	0.00	**445.01**	-5.44
E22-K4-S10-14	**371.50**	**371.50**	0.00	**366.90**	-1.24
E22-K4-S11-12	**427.22**	**427.22**	0.00	**418.41**	-2.06
E22-K4-S12-16	**392.78**	**392.78**	0.00	**377.23**	-3.96
E33-K4-S01-09	**730.16**	**730.16**	0.00	**727.64**	-0.35
E33-K4-S02-13	**714.64**	**714.63**	0.00	**712.12**	-0.35
E33-K4-S03-17	**707.49**	**707.48**	0.00	706.23	-0.18
E33-K4-S04-05	**778.74**	801.04	2.78	801.04	0.00
E33-K4-S07-25	**756.85**	**756.84**	0.00	**741.15**	-2.07
E33-K4-S14-22	**779.05**	**779.05**	0.00	**779.05**	0.00
E51-K5-S02-17	597.49	597.49	0.00	597.43	-0.01
E51-K5-S04-46	**530.76**	**530.76**	0.00	530.76	0.00
E51-K5-S06-12	554.80	627.09	11.53	576.58	-8.06
E51-K5-S11-19	581.64	581.64	0.00	581.64	0.00
E51-K5-S27-47	**538.22**	**538.22**	0.00	538.22	0.00
E51-K5-S32-37	**552.28**	**552.28**	0.00	552.28	0.00
E51-K5-S02-04-17-46	530.76	572.69	7.32	572.69	0.00
E51-K5-S06-12-32-37	531.92	622.28	14.52	622.28	0.00
E51-K5-S11-19-27-47	531.12	610.39	12.99	610.39	0.00
Average	565.72	580.29	2.34	574.23	-1.13

Table 5: Results of set-3 instances

Instances	BKS (i) (Same depot)	Present model			
		Same depot (ii)	% dev between (i) and (ii)	Different depot (iii)	% dev between (ii) and (iii)
E22-K4-S13-14	**526.15**	**526.15**	0.00	**519.20**	-1.32
E22-K4-S13-16	**521.09**	**521.09**	0.00	**515.11**	-1.15
E22-K4-S13-17	**496.38**	**496.38**	0.00	**495.72**	-0.13
E22-K4-S14-19	**498.81**	**498.81**	0.00	**484.56**	-2.86
E22-K4-S17-19	**512.80**	**512.80**	0.00	**501.48**	-2.21
E22-K4-S19-21	**520.41**	**520.41**	0.00	**513.87**	-1.26
E33-K4-S16-22	672.17	672.17	0.00	660.16	-1.79
E33-K4-S16-24	666.02	666.02	0.00	664.46	-0.23
E33-K4-S19-26	**680.36**	**680.36**	0.00	**668.97**	-1.67
E33-K4-S22-26	680.89	680.36	-0.08	678.82	-0.23
E33-K4-S24-28	**670.43**	**670.43**	0.00	**668.25**	-0.32
E33-K4-S25-28	**650.58**	**650.58**	0.00	643.95	-1.02
E51-K5-S12-18	692.37	705.32	1.84	705.32	0.00
E51-K5-S12-41	691.37	805.89	14.21	805.89	0.00
E51-K5-S12-43	712.48	712.48	0.00	712.48	0.00
E51-K5-S39-41	729.94	799.11	8.66	785.52	-1.70
E51-K5-S40-41	729.94	857.00	14.83	791.139	-7.69
E51-K5-S40-43	757.30	805.77	6.02	805.77	0.00
Average	630.27	650.73	2.52	641.97	-1.30

results reported are the upper bounds obtained at the end of the time limit. Results show that the relaxation resulted in savings of fuel up to 4.51% with an average savings of 1.97%. In the case of distance minimization, there is a reduction in the distance up to 11.66% with an average reduction of 3.05%.

Table 6: Results for new instances

Instances	Fuel minimization			Distance minimization		
	Different depot	**Same depot**	**% dev**	**Different depot**	**Same depot**	**% dev**
C16-S4-D2	**28.40**	**28.41**	-0.05	**136.513**	**137.542**	-0.75
C18-S4-D2	**37.95**	**39.00**	-2.68	**175.690**	**180.127**	-2.46
C24-S6-D2	51.56	54.00	-4.51	220.615	227.032	-2.83
C28-S5-D2	49.60	50.35	-1.50	226.285	256.149	-11.66
C28-S6-D3	47.36	48.73	-2.82	210.787	211.426	-0.30
C30-S3-D1	60.22	59.43	-0.83	275.610	276.190	-0.21
C30-S6-D3	54.88	56.03	-2.05	265.817	278.515	-4.56
C36-S4-D2	62.37	63.21	-1.32	277.699	282.215	-1.60
Average	49.04	49.91	-1.97	223.627	231.149	-3.05

Table 7 summarizes the results of solving the formulation using Gurobi for the modified set-4 instances. We solve these instances with the relaxation that vehicles can end at different depots. In these instances, Gurobi is not able to find optimal solutions within the time limit of 10,000s. Hence, we have made a comparison between the BKS and the upper bound obtained at the end of the time limit. We mark the cases where Gurobi is unable to find any feasible solution within the time limit or ran out of memory with a "*". These results are used to compare the performance of ALNS on medium-sized instances.

Table 7: Results for modified set-4 instances

Instances	Fuel minimization			Distance minimization		
	BKS	**Best bound**	**% dev**	**BKS**	**Best bound**	**% dev**
Instance-50-37	*			1634.79	794.95	51.37
Instance-50-38	21.62	15.16	29.88	1505.78	672.35	55.35
Instance-50-39	26.33	19.64	25.41	1423.04	871.91	38.73
Instance-50-40	20.20	13.7	32.18	1478.26	617.16	58.25
Instance-50-41	33.27	25.2	24.26	1727.80	956.68	44.63
Instance-50-42	24.74	20.74	16.17	1202.45	723.66	39.82
Instance-50-43	24.17	21.14	12.54	1291.52	719.11	44.32
Instance-50-44	19.94	16.14	19.06	1189.69	548.01	53.94
Instance-50-45	25.95	18.73	27.82	1301.60	724.43	44.34
Instance-50-46	18.79	14.27	24.06	942.37	535.58	43.17
Instance-50-47	30.28	25.4	16.12	1322.50	830.82	37.18
Instance-50-48	21.45	17.34	19.15	*		
Instance-50-49	28.42	19.89	30.01	1324.65	732.19	44.73
Instance-50-50	20.35	13.65	32.92	1002.58	579.21	42.23
Instance-50-51	24.94	18.39	26.26	1156.92	728.54	37.03
Instance-50-52	27.45	13.55	50.64	1096.80	567.25	48.28
Instance-50-53	37.90	19.91	47.47	1414.12	785.59	44.45
Instance-50-54	21.50	15.31	28.79	1008.57	619.24	38.60
Average	25.14	18.13	27.22	1295.50	706.28	45.08

Table 8 summarizes the results of solving the formulation using Gurobi with fuel minimization and distance minimization objectives separately. There is an average saving of 13.11% in fuel consumption even with an average increase of 15.11% in distance traveled while minimizing fuel consumed. The use of a heterogeneous fleet is a likely reason for the significant reduction in fuel consumption and an increase in distance. With the fuel minimization objective, the model utilizes all the available fuel-efficient trucks whereas in distance minimization case, the model uses large trucks. Hence, minimizing distance does not always reduce fuel consumption since the load on the vehicle and type of vehicle significantly impact fuel consumption.

Table 8: Comparison of distance minimization and fuel minimization objectives

Instances	**Fuel minimization**		**Distance minimization**		**Distance savings (%)(iv-ii)**	**Fuel savings (%)(iii-i)**
	Fuel cost (i)	**Distance cost (ii)**	**Fuel cost (iii)**	**Distance cost (iv)**		
C16-S4-D2	**28.40**	**160.03**	**40.71**	**136.51**	-14.70	30.21
C18-S4-D2	**37.95**	**229.50**	**51.54**	**175.69**	-23.45	24.33
C24-S6-D2	51.56	264.02	56.65	220.62	-16.44	4.68
C28-S5-D2	49.60	265.32	57.78	226.28	-14.71	12.60
C28-S6-D3	47.36	275.26	58.53	210.79	-23.42	16.74
C30-S3-D1	60.22	293.85	62.31	275.61	-6.21	4.62
C30-S6-D3	54.88	296.44	61.68	265.82	-10.33	9.16
C36-S4-D2	62.37	314.30	64.87	277.70	-11.64	2.56
Average	49.04	262.34	56.76	223.63	-15.11	13.11

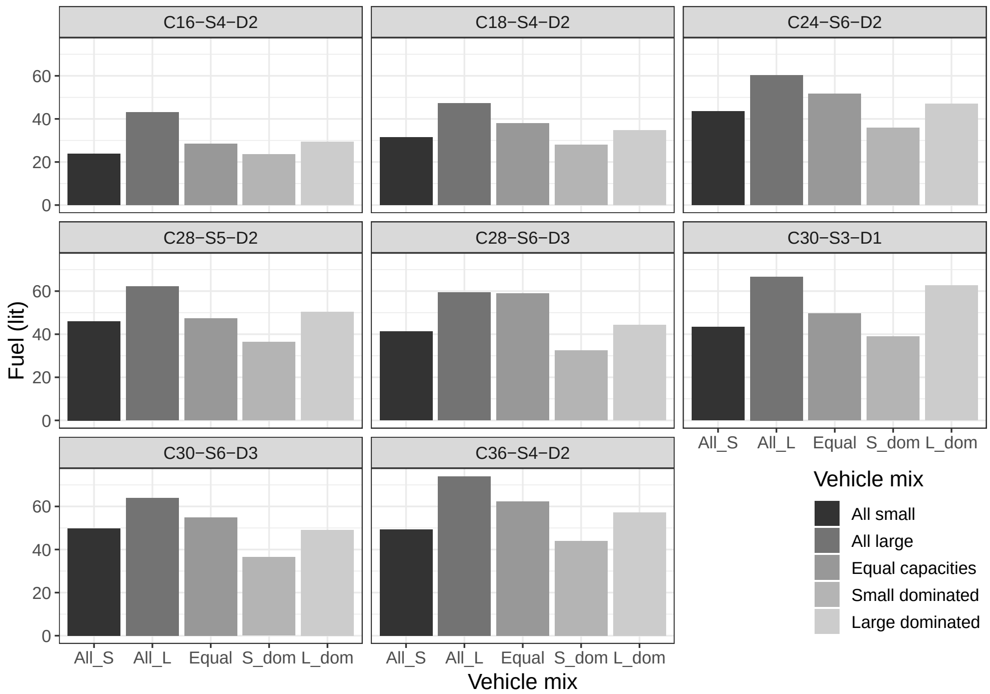


Figure 5: Effect of variation in the heterogeneous fleet on solution

Figure 5 shows the variation in fuel consumed with heterogeneous fleet composition. We compare five cases each with a different mix of trucks. The first two cases have homogeneous fleets of either small trucks (all small) or large trucks (all large) in each echelon. The next three cases have heterogeneous fleets in each echelon; equal capacities case has the mix of trucks such that the total capacity of all the available large trucks is equal to that of smaller trucks; 'small-dominated' ('large-dominated') case has more small (large) trucks (i.e., total capacities of all small trucks is more than that of large trucks). Having heterogeneous fleets results in lower fuel consumption than homogeneous fleets. Among the heterogeneous fleets cases, operating excess of smaller trucks results in the best fuel economy. We attribute these variations in homogeneous and heterogeneous cases to the reduction in empty and less than truckload trips.

Figure 6 shows the optimal routes for the instance C16-S4-D2 in all five cases. In these figures, we represent customers as circles, satellites as squares, and depots as triangles. We draw routes from depot and satellite with different line types based on the vehicle type used. The bottom two plots show the optimal routes obtained for the homogeneous fleet cases (all small vehicles and all large vehicles) and the cases with the heterogeneous fleet are shown in the top three figures. From these figures, we observe that small dominated case (refer the third plot in top row) uses a better mix of available vehicles.

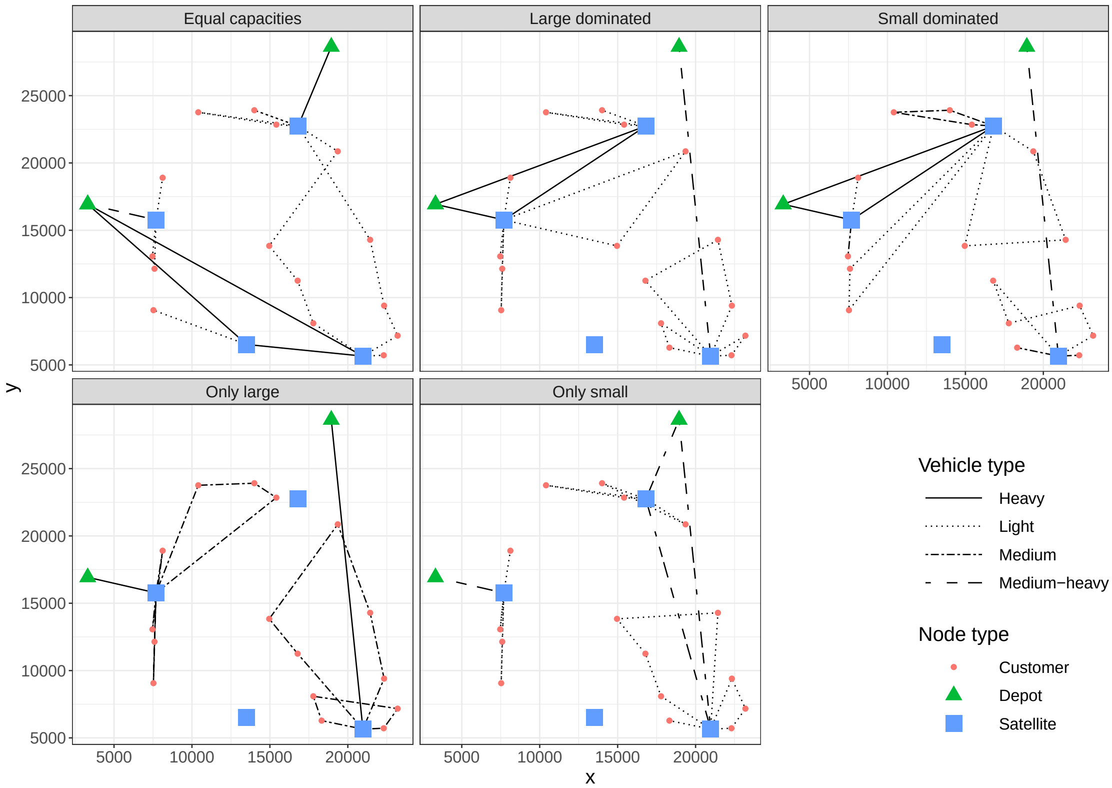


Figure 6: Optimal routes for Heterogeneous case

5.5. *Computational results of ALNS*

To validate the implemented ALNS, we compare with the results of Gurobi for the instance set-2 and set-3 with the relaxation that vehicles can return to any depot. ALNS algorithm found solutions that are the same if not better than the Gurobi in most cases and took an average time of 409 and 389 seconds for set-2 and set-3 respectively (Table 9). Gurobi is not able to find optimal solutions for set-4 instances within the time-limit of 10,000s. Hence, the set-4 instances results are used to compare the performance of ALNS. The results (Table 10) show that ALNS produced superior results compared to Gurobi in significantly less time and overall ALNS found solutions that are better than Gurobi in 57 cases, and in 23 cases the results are same as the one found by Gurobi. ALNS on set-4 instances (refer Table 11) is run imposing the constraint on vehicles to return to the same depot. It found new solutions in four cases and the same solutions as previous best solutions in eight other cases. The average gap for 54 instances is 0.73%, and runtime is 771s. The ALNS results reported here are after 200,000 iterations, whereas the best-known solutions reported in the literature were for 500,000 iterations. All the results reported for ALNS are the best out of three runs and the corresponding time.

Table 9: ALNS results compared to Gurobi on set-2 and set-3 Instances

Instances	**Gurobi**	**ALNS**	**% dev**	**ALNS Time (s)**	**Instances**	**Gurobi**	**ALNS**	**% dev**	**ALNS Time (s)**
Set-2					**Set-3**				
E22-K4-S06-17	416.85	416.85	0.00	103	E22-K4-S13-14	519.20	519.20	0.00	125
E22-K4-S08-14	384.96	384.96	0.00	130	E22-K4-S13-16	515.11	515.11	0.00	168
E22-K4-S09-19	445.01	445.01	0.00	109	E22-K4-S13-17	495.72	495.72	0.00	118
E22-K4-S10-14	366.90	366.90	0.00	190	E22-K4-S14-19	484.56	484.56	0.00	148
E22-K4-S11-12	418.41	418.41	0.00	130	E22-K4-S17-19	501.48	501.48	0.00	137
E22-K4-S12-16	377.23	377.23	0.00	104	E22-K4-S19-21	513.87	513.87	0.00	260
E33-K4-S01-09	727.64	727.64	0.00	167	E33-K4-S16-22	660.16	650.81	-1.42	155
E33-K4-S02-13	712.12	712.12	0.00	222	E33-K4-S16-24	664.46	654.09	-1.56	272
E33-K4-S03-17	706.23	695.81	-1.48	166	E33-K4-S19-26	668.97	668.98	0.00	172
E33-K4-S04-05	801.04	778.74	-2.78	228	E33-K4-S22-26	678.82	678.82	0.00	278
E33-K4-S07-25	741.15	741.15	0.00	167	E33-K4-S24-28	668.25	672.03	0.56	295
E33-K4-S14-22	779.05	779.05	0.00	166	E33-K4-S25-28	643.95	643.95	0.00	193
E51-K5-S02-17	597.43	597.43	0.00	656	E51-K5-S12-18	705.32	710.97	0.80	709
E51-K5-S04-46	530.76	531.77	0.19	737	E51-K5-S12-41	805.89	703.44	-12.71	894
E51-K5-S06-12	576.58	564.46	-2.10	727	E51-K5-S12-43	715.02	716.65	0.23	812
E51-K5-S11-19	581.64	585.83	0.72	665	E51-K5-S39-41	785.52	766.56	-2.41	689
E51-K5-S27-47	538.22	538.22	0.00	800	E51-K5-S40-41	791.14	791.14	0.00	716
E51-K5-S32-37	552.28	552.28	0.00	794	E51-K5-S40-43	805.77	850.26	5.23	863
E51-K5-S02-04-17-46	572.69	530.76	-7.32	864					
E51-K5-S06-12-32-37	622.28	564.99	-9.21	659					
E51-K5-S11-19-27-47	610.39	550.01	-9.89	806					
Average	574.23	564.74	-1.52	409		645.73	640.98	-0.63	389

Table 12 shows the performance of ALNS in both fuel minimization and distance minimization cases on the modified set-4 instances. ALNS performs exceptionally well in all these instances. In fuel minimization case it found nine new solutions and eight solutions same as Gurobi. In the distance minimization case, it found new solutions in all 18 instances. The average time taken is 893 and 817 seconds for fuel and distance minimization cases respectively.

Table 10: ALNS results compared to Gurobi on Set-4 instance

Instances	Gurobi	ALNS	% dev	Time (s)	Instances	Gurobi	ALNS	% dev	Time (s)
Instance_50-1	1617	1582	-2.18	476	Instance_50-28	1228	1216	-0.94	725
Instance_50-2	1499	1442	-3.81	597	Instance_50-29	1621	1569	-3.22	562
Instance_50-3	1625	1580	-2.75	966	Instance_50-30	1219	1218	-0.13	858
Instance_50-4	1461	1433	-1.89	1166	Instance_50-31	1677	1454	-13.29	758
Instance_50-5	2510	2194	-12.59	446	Instance_50-32	1194	1206	0.96	520
Instance_50-6	1314	1280	-2.57	564	Instance_50-33	1639	1517	-7.43	1149
Instance_50-7	1483	1438	-3.08	583	Instance_50-34	1245	1233	-0.93	654
Instance_50-8	1368	1366	-0.15	1141	Instance_50-35	1686	1598	-5.23	950
Instance_50-9	1451	1458	0.45	498	Instance_50-36	1226	1241	1.17	995
Instance_50-10	1361	1361	0.03	845	Instance_50-37	1844	1543	-16.32	710
Instance_50-11	2220	2070	-6.77	410	Instance_50-38	1303	1173	-9.99	891
Instance_50-12	1217	1209	-0.66	659	Instance_50-39	2997	1535	-48.77	995
Instance_50-13	1495	1478	-1.13	788	Instance_50-40	1224	1179	-3.65	440
Instance_50-14	1422	1411	-0.78	1029	Instance_50-41	2340	1694	-27.63	870
Instance_50-15	1422	1476	3.65	995	Instance_50-42	1260	1194	-5.24	863
Instance_50-16	1401	1391	-0.70	706	Instance_50-43	1576	1422	-9.76	520
Instance_50-17	2206	2117	-4.05	1039	Instance_50-44	1082	1039	-3.98	883
Instance_50-18	1314	1228	-6.56	1184	Instance_50-45	1711	1455	-14.95	1180
Instance_50-19	1900	1571	-17.30	536	Instance_50-46	1166	1068	-8.42	870
Instance_50-20	1412	1279	-9.42	512	Instance_50-47	1676	1590	-5.14	1120
Instance_50-21	1649	1581	-4.10	630	Instance_50-48	1109	1074	-3.19	1051
Instance_50-22	1324	1289	-2.67	438	Instance_50-49	1916	1442	-24.74	624
Instance_50-23	1986	1681	-15.35	1147	Instance_50-50	1234	1074	-12.98	499
Instance_50-24	1291	1285	-0.53	1051	Instance_50-51	1286	1405	8.48	800
Instance_50-25	1455	1436	-1.29	621	Instance_50-52	2563	1109	-56.74	447
Instance_50-26	1165	1167	0.18	700	Instance_50-53	*	1552		424
Instance_50-27	1722	1483	-13.87	1014	Instance_50-54	1160	1129	-2.69	516
Average						1556.13	1411.38	-7.26	771

Table 11: ALNS results compared to Gurobi on Set-4 instances with vehicle ending at same depot

Instances	BKS	ALNS	% dev	Time (s)
Instance_50-5	2194	2194	0.00	446
Instance_50-6	1280	1280	0.00	564
Instance_50-7	1459	1438	-1.44	583
Instance_50-10	1361	1361	0.00	845
Instance_50-12	1209	1209	0.00	659
Instance_50-13	1482	1478	-0.27	788
Instance_50-15	1490	1476	-0.94	995
Instance_50-18	1228	1228	0.00	1184
Instance_50-25	1440	1436	-0.30	621
Instance_50-26	1167	1167	0.00	700
Instance_50-34	1233	1233	0.00	654
Instance_50-48	1074	1074	0.00	1051

Table 12: ALNS on modified set-4 instances

Instances	Fuel Minimization				Distance Minimization			
	Gurobi	ALNS	% dev	Time (s)	Gurobi	ALNS	% dev	Time (s)
Instance_50-37-2	*	29.86	*	1248	1634.79	1063.22	-34.96	849
Instance_50-38-2	21.62	21.62	0.00	731	1505.78	1038.96	-31.00	1038
Instance_50-39-2	26.33	26.33	0.00	1163	1423.04	1106.84	-22.22	1241
Instance_50-40-2	20.20	20.20	0.00	675	1478.26	967.54	-34.55	1024
Instance_50-41-2	33.27	28.11	-15.52	924	1727.80	1224.89	-29.11	1014
Instance_50-42-2	24.74	24.56	-0.73	748	1202.45	1095.36	-8.91	735
Instance_50-43-2	24.17	24.17	0.00	1247	1291.52	983.96	-23.81	669
Instance_50-44-2	19.94	19.49	-2.27	1066	1189.69	983.75	-17.31	1070
Instance_50-45-2	25.95	25.95	0.00	704	1301.60	1018.41	-21.76	735
Instance_50-46-2	18.79	16.84	-10.38	734	942.37	899.91	-4.51	1037
Instance_50-47-2	30.28	26.63	-12.06	977	1322.50	1104.75	-16.47	478
Instance_50-48-2	21.45	21.56	0.51	771	*	952.81	*	744
Instance_50-49-2	28.42	28.42	0.00	619	1324.65	945.8	-28.60	623
Instance_50-50-2	20.35	20.35	0.00	1146	1002.58	890.03	-11.23	599
Instance_50-51-2	24.94	24.75	-0.74	871	1156.92	1036.41	-10.42	833
Instance_50-52-2	27.45	26.97	-1.75	747	1096.80	908.7	-17.15	667
Instance_50-53-2	37.90	32.94	-13.10	728	1414.12	979.03	-30.77	710
Instance_50-54-2	21.50	21.50	0.00	988	1008.57	961.91	-4.63	643
Average	25.14	24.46	-3.30	893.72	1295.50	1009.02	-20.43	817.17

In the case of newly introduced instances, Gurobi found optimal solutions only for cases C16-S4-D2 and C18-S4-D2. ALNS found solutions that are better in 7 out of 16 (8 each with fuel and distance minimization) instances and the same as Gurobi in 4 cases. The average improvement is 3.9% in fuel minimization case and -0.19% in distance minimization case (Table 13).

Table 13: ALNS on new instances

Instances	Fuel minimization				Distance minimization			
	Gurobi	ALNS	% dev	Time (s)	Gurobi	ALNS	% dev	Time (s)
C16-S4-D2	28.4	28.40	0.00	233	136.51	136.51	0.00	227
C18-S4-D2	37.95	37.95	0.00	254	175.69	180.13	2.46	278
C24-S6-D2	51.57	48.58	-6.15	342	220.62	225.34	2.09	385
C28-S5-D2	49.6	50.09	0.98	1161	226.29	226.29	0.00	963
C28-S6-D3	47.36	44.72	-5.91	729	210.79	218.62	3.58	542
C30-S3-D1	58.94	56.77	-3.83	502	275.61	278.20	0.93	429
C30-S6-D3	54.89	51.78	-6.01	1023	265.82	251.26	-5.80	989
C36-S4-D2	62.38	56.21	-10.98	604	277.70	272.82	-1.79	965
Average	48.88	46.81	-3.99	606	223.63	223.64	0.19	597

5.6. Computational results with vehicle minimization

A main limitation of existing implementations of ALNS is it does not minimize the number of vehicles. Bin packing gives the least number of vehicles required to satisfy all the demand. However, restricting to the minimum number of vehicles may result in increased fuel consumption or distance traveled. A tradeoff between number of vehicles used and fuel consumption is often desirable. From Table 14, it is clear that the ALNS algorithm is not restricted to use only the vehicles in the initial solution if using more vehicles reduces fuel consumption.

Table 14: Variation in number of vehicles between the initial and final solutions of ALNS

Instances	**Initial solution**						**Final solution**					
	Heavy Vehicles	**Medium Heavy vehicles**	**Medium vehicles**	**Light vehicles**	**Total Vehicles**	$f'(X,Y)$	**Heavy vehicles**	**Medium Heavy vehicles**	**Medium vehicles**	**Light vehicles**	**Total Vehicles**	$f'(X,Y)$
C16-S4-D2	1	1	3	3	8	289.28	1	1	1	7	10	136.51
C18-S4-D2	2	0	4	3	9	354.37	2	0	1	9	12	180.13
C24-S6-D2	2	1	6	2	11	444.45	2	1	2	10	15	225.34
C28-S5-D2	2	1	5	5	13	566.07	2	1	4	8	15	226.29
C28-S6-D3	2	1	6	4	13	490.69	2	1	1	13	17	218.62
C30-S3-D1	1	0	3	10	14	535.20	1	0	3	12	16	278.20
C30-S6-D3	2	2	6	6	16	595.62	2	2	2	12	18	251.26
C36-S4-D2	3	0	4	10	17	604.37	3	0	1	15	19	272.82

To minimize the number of vehicles along with the total cost, we have modified the objective function (equation 6a, 6b) to equation 29 by adding a secondary objective that minimizes the number of vehicles required. ω is the weight required to convert the vehicles to cost. We have used a value of 10000 for ω in this study.

$$f''(X^*, Y^*, Q^*) = min f(X,Y,Q) + \omega \left(\sum_{h \in \mathbf{H}} \sum_{t \in \mathbf{T}} \sum_{j \in \mathbf{S}} x_{tj}^{th} + \sum_{m \in \mathbf{M}} \sum_{s \in \mathbf{S}} \sum_{l \in \mathbf{C}} y_{sl}^{sm} \right) \tag{29}$$

We have solved the instances using ALNS with (P_{II}) and without (P_I) vehicle minimization, and a comparison between them is made in Table 15. Results show that ALNS for P_{II} uses lesser vehicles compared to ALNS for P_I, however the fuel consumption and distance traveled are marginally higher (2.71 % and 0.55% higher on average respectively).

Table 15: Comparison of two ALNS on new instances

Instances	**Fuel minimization**						**Distance minimization**					
	Objective value		**% dev**	**Vehicles used**		**% dev**	**Objective value**		**% dev**	**Vehicles used**		**% dev**
	$f(P_I)$	$f(P_{II})$		P_I	P_{II}		$f(P_I)$	$f(P_{II})$		P_I	P_{II}	
C16-S4-D2	28.4	28.4	0.00	10	9	10.00	136.51	136.51	0.00	10	9	10.00
C18-S4-D2	37.95	37.95	0.00	13	12	7.69	180.13	176.53	-2.04	12	12	0.00
C24-S6-D2	46.00	50.29	8.54	15	14	6.67	225.34	220.62	-2.14	15	13	13.33
C28-S5-D2	50.09	49.31	-1.58	15	15	0.00	226.29	229.45	1.38	15	15	0.00
C28-S6-D3	44.72	46.42	3.67	17	16	5.88	218.62	222.22	1.62	17	16	5.88
C30-S3-D1	56.77	59.23	4.16	16	16	0.00	278.20	272.45	-2.11	16	16	0.00
C30-S6-D3	51.78	54.17	4.42	18	18	0.00	251.26	267.43	6.05	18	18	0.00
C36-S4-D2	56.21	57.66	2.25	19	18	5.26	272.82	277.43	1.66	19	18	5.26
Average	46.49	47.93	2.71	15.37	14.75	4.44	223.65	225.33	0.55	15.25	14.62	4.31

6. Conclusion

We introduced a MILP formulation for Multi Depot Two-Echelon Fuel Minimizing Routing Problem (MD2E-FMRP) with heterogeneous fleet at each echelon. The model allows for speed variations in the comprehensive fuel estimation function, multiple depots, and heterogeneous fleets. We tested the model on small to medium-sized instances, and Gurobi was able to find optimal solutions in 49 instances and an upper bound in 87 instances within a time limit of 10,000s. Results showed that optimizing for fuel consumption instead of distance traveled saved 13.11% in fuel consumed even though distance traveled increased by 15.11%. We also showed that heterogeneous fleets always saves more fuel compared to homogeneous fleets. Allowing the vehicles to end at any depot (satellite) and simultaneously conserving vehicle flow at every depot (satellite) reduces both the fuel consumed and the distance traveled.

We proposed an Adaptive Large Neighborhood Search (ALNS) heuristic and tested on existing Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP) instances and newly introduced MD2E-FMRP instances. ALNS was able to find solutions that are equal to or better than existing best-known solutions for 86 out of 93 2E-CVRP instances and 29 out of 34 newly introduced MD2E-FMRP instances. The ALNS solutions, especially in the case of set-4 instances, were much better than the upper bound provided by Gurobi even though the runtime was less than one-tenth of Gurobi. We also presented a slightly modified version of ALNS with a secondary objective of minimizing the number of vehicles, and the results showed that both original and modified heuristics have comparable performances. However, the modified heuristic may be preferred if the operators are interested in minimizing the number of vehicles as well.

A possible extension of this paper is to formulate a problem that synchronizes both the first and second echelon vehicles at the satellites. This extension is essential in case of perishable goods distribution or when satellites do not have storage facility. The formulation can also be modified to incorporate electric vehicles.

Acknowledgment

The initial ideas for this paper took shape when the first author was visiting Prof. Russell Thompson at the University of Melbourne. The authors thank the Volvo Research and Education Foundations (VREF) and Center of Excellence for Sustainable Urban Freight Systems (CoE-SUFS) for supporting the visit.

References

Abdulkader, M. M., Gajpal, Y., ElMekkawy, T. Y., 2018. Vehicle routing problem in omni-channel retailing distribution systems. International Journal of Production Economics 196, 43–55.

Barth, M., Scora, G., Younglove, T., 2004. Modal Emissions Model for Heavy-duty Diesel Vehicles. Transportation Research Record: Journal of the Transportation Research Board 1880, 10–20.

Barth, M., Younglove, T., Scora, G., 2005. Development of a Heavy-duty Diesel Modal Emissions and Fuel Consumption Model. Tech. rep., University of California, Riverside.

Bathmanabhan, S., Madanayak, S. N. S., 2010. Analysis and Interpretation of Particulate Matter- PM10, PM2.5 and PM1 Emissions From The Heterogeneous Traffic Near an Urban Roadway. Atmospheric Pollution Research, 184–194.

Bektas, T., Laporte, G., 2011. The Pollution-Routing Problem. Transportation Research Part B: Methodological 45, 1232–1250.

Boccia, M., Crainic, T. G., Sforza, A., Sterle, C., 2010. A Metaheuristic for a Two Echelon Location-Routing Problem. In: Experimental Algorithms: 9th International Symposium, SEA 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings. Springer Berlin Heidelberg, pp. 288–301.

Boccia, M., Crainic, T. G., Sforza, A., Sterle, C., 2011. Location-Routing Models for Designing a Two-echelon Freight Distribution System. Tech. rep., Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport.

Chao, I. M., 2002. A Tabu Search Method for The Truck and Trailer Routing Problem. Computers & Operations Research 29, 33–51.

Clarke, G., Wright, J. W., 1964. Scheduling of Vehicles From a Central Depot to a Number of Delivery Points. Operations Research 12, 568–581.

Contardo, C., Hemmelmayr, V., Crainic, T. G., 2012. Lower and Upper Bounds for The Two-echelon Capacitated Location-Routing Problem. Computers and Operations Research 39, 3185–3199.

Crainic, T. G., Mancini, S., Perboli, G., Tadei, R., 2011. Multi-start Heuristics for The Two-echelon Vehicle Routing Problem. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6622 LNCS, 179–190.

Crainic, T. G., Mancini, S., Perboli, G., Tadei, R., 2012. Impact of Generalized Travel Costs on Satellite Location In The Two-echelon Vehicle Routing Problem. Procedia - Social and Behavioral Sciences 39, 195–204.

Cuda, R., Guastaroba, G., Speranza, M. G., 2015. A Survey on Two-echelon Routing Problems. Computers and Operations Research 55, 185–199.

Czyzyk, J., Mesnier, M. P., Moré, J. J., 1998. The neos server. IEEE Journal on Computational Science and Engineering 5, 68–75.

Demir, E., Bektaş, T., Laporte, G., 2011. A Comparative Analysis of Several Vehicle Emission Models for Road Freight Transportation. Transportation Research Part D: Transport and Environment 16, 347–357.

Demir, E., Bektaş, T., Laporte, G., 2012. An Adaptive Large Neighborhood Search Heuristic for The Pollution-Routing Problem. European Journal of Operational Research 223 (2), 346–359.

Demir, E., Bektaş, T., Laporte, G., 2014. The Bi-objective Pollution-Routing Problem. European Journal of Operational Research 232 (3), 464–478.

Dolan, E. D., 2001. The neos server 4.0 administrative guide. Technical memorandum, Mathematics and Computer Science Division, Argonne National Laboratory.

Eicher, 2018. Technical specifications of Eicher PRO 1049.
URL `http://www.eicher.in/etb/uploads/pro-1049.pdf`

Escuín, D., Millán, C., Larrodé, E., 2012. Modelization of Time-Dependent Urban Freight Problems by Using a Multiple Number of Distribution Centers. Networks and Spatial Economics 12, 321–336.

Feliu, J. G., Perboli, G., Tadei, R., Vigo, D., 2007. The Two-echelon Capacitated Vehicle Routing Problem. Proceedings of the 22nd European Conference on Operational Research, Prague, 1–40.

Force, 2018. Technical specifications of FORCE TRUMP 40.
URL `https://trucks.cardekho.com/en/trucks/force/trump-40/specifications`

Franceschetti, A., Honhon, D., Van Woensel, T., Bektas, T., Laporte, G., 2013. The Time-dependent Pollution-Routing Problem. Transportation Research Part B: Methodological 56, 265–293.

Gropp, W., Moré, J. J., 1997. Optimization environments and the neos server. In: Buhman, M. D., Iserles, A. (Eds.), Approximation Theory and Optimization. Cambridge University Press, pp. 167 – 182.

Hemmelmayr, V. C., Cordeau, J. F., Crainic, T. G., 2012. An Adaptive Large Neighborhood Search Heuristic for Two-echelon Vehicle Routing Problems Arising In City Logistics. Computers & Operations Research 39, 3215–3228.

Jacobsen, S., Madsen, O., 1980. A Comparative Study of Heuristics for a Two-level Routing-location Problem. European Journal of Operational Research 5 (6), 378–387.

Jepsen, M., Spoorendonk, S., Ropke, S., 2013. A Branch-and-cut Algorithm for The Symmetric Two-echelon Capacitated Vehicle Routing Problem. Transportation Science 47 (1), 23–37.

Kancharla, S. R., Ramadurai, G., 2018. Incorporating driving cycle based fuel consumption estimation in green vehicle routing problems. Sustainable Cities and Society 40, 214–221.

Lin, S. W., Yu, V. F., Chou, S. Y., 2009. Solving The Truck and Trailer Routing Problem Based on a Simulated Annealing Heuristic. Computers & Operations Research 36, 1683–1692.

Lin, S. W., Yu, V. F., Chou, S. Y., 2010. A Note on The Truck and Trailer Routing Problem. Expert Systems with Applications 37, 899–903.

Lin, S. W., Yu, V. F., Lu, C. C., 2011. A Simulated Annealing Heuristic for The Truck and Trailer Routing Problem With Time Windows. Expert Systems with Applications 38, 15244–15252.

Meihua, W., Xuhong, T., Shan, C., Shumin, W., 2011. Hybrid Ant Colony Optimization Algorithm for Two Echelon Vehicle Routing Problem. Procedia Engineering 15, 3361–3365.

Mirmohammadsadeghi, S., Ahmed, S., 2015. Memetic Heuristic Approach for Solving Truck and Trailer Routing Problems with Stochastic Demands and Time Windows. Networks and Spatial Economics 15, 1093–1115.

MORTH, 2015. Tech. rep., Ministry of Road Transport & Highways, Transport Research Wing, Government of India.

Nagy, G., Salhi, S., 1996. Nested heuristic methods for the location-routing problem. Journal of the Operational Research Society 47, 1166–1174.

Nguyen, V. P., Prins, C., Prodhon, C., 2012a. A Multi-start Iterated Local Search With Tabu List and Path Relinking for The Two-echelon Location-Routing Problem. Engineering Applications of Artificial Intelligence 25, 56–71.

Nguyen, V. P., Prins, C., Prodhon, C., 2012b. Solving The Two-echelon Location Routing Problem By a Grasp Reinforced By a Learning Process and Path Relinking. European Journal of Operational Research 216, 113–126.

Perboli, G., Tadei, R., Tadei, R., 2010. New Families of Valid Inequalities for The Two-echelon Vehicle Routing Problem. Electronic Notes in Discrete Mathematics 36, 639–646.

Perboli, G., Tadei, R., Vigo, D., 2011. The Two-echelon Capacitated Vehicle Routing Problem: Models and Math-based Heuristics. Transportation Science 45 (3), 364–380.

Renaud, J., Boctor, F. F., 2002. A Sweep-based Algorithm for The Fleet Size and Mix Vehicle Routing Problem. European Journal of Operational Research 140, 618–628.

Ropke, S., Pisinger, D., 2006. An Adaptive Large Neighborhood Search Heuristic for The Pickup and Delivery Problem With Time Windows. Transportation Science 40, 455–472.

Sahin, B., Yilmaz, H., Ust, Y., Guneri, A. F., Gulsun, B., 2009. An Approach for Analysing Transportation Costs and a Case Study. European Journal of Operational Research 193 (1), 1–11.

Scheuerer, S., 2006. A Tabu Search Heuristic for The Truck and Trailer Routing Problem. Computers & Operations Research 33, 894–909.

Schwengerer, M., Pirkwieser, S., Raidl, G. R., 2012. A Variable Neighborhood Search Approach for The Two-echelon Location-Routing Problem. In: Evolutionary Computation in Combinatorial Optimization: 12th European Conference, EvoCOP 2012, Malaga, Spain, April 11-13, 2012. Proceedings. Springer Berlin Heidelberg, pp. 13–24.

Semet, F., Taillard, E., 1993. Solving Real-life Vehicle Routing Problems Efficiently Using Tabu Search. Annals of Operations Research 41, 469–488.

Soysal, M., M.Bloemhof-Ruwaard, J., Bektaş, T., 2015. The Time-dependent Two-echelon Capacitated Vehicle Routing Problem With Environmental Considerations. International Journal of Production Economics 164, 366–378.

TATA, 2018. Technical specifications of TATA LPT 1613.
URL http://tatatrucks.tatamotors.com/downloads/pdf/lpt-16-2t-gvw-bs-4.pdf

Toyoglu, H., Karasan, O. E., Kara, B. Y., 2012. A New Formulation Approach for Location-Routing Problems. Networks and Spatial Economics 12, 635–659.

Tuzun, D., Burke, L. I., 1999. A Two-phase Tabu Search Approach to The Location Routing Problem. European Journal of Operational Research 116, 87–99.

USEPA, 2017. Dynamometer Drive Schedules.
URL https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., Velasco, N., 2011. A Grasp With Evolutionary Path Relinking for The Truck and Trailer Routing Problem. Computers & Operations Research 38, 1319–1334.

WHO, 2016. Air Pollution Levels Rising In Many of The World's Poorest Cities.
URL http://www.who.int/mediacentre/news/releases/2016/air-pollution-rising/en/

WHO, 2018. WHO ambient (outdoor) air quality database Summary results, update 2018. Tech. Rep. April.