Electric Vehicle Routing Problem with Non-Linear Charging and Load-Dependent Discharging

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

[a] *Senior Project Officer, 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*
[c] *Faculty Member, Robert Bosch Center for Data Science and Artificial Intelligence, Indian Institute of Technology Madras, Chennai-600036, India*

Abstract

We propose a three-index formulation for E-VRP with Non-Linear charging and Load-Dependent discharging (E-VRP-NL-LD), and an Adaptive Large Neighborhood Search (ALNS) algorithm to solve the E-VRP-NL-LD and E-VRP-NL-LD with Capacitated Charging Stations (E-VRP-NL-LD-CCS). Existing implementations of EVRP duplicate charging station nodes which enables the modelling of EVRP using extended VRP formulations. Two limitations of such an approach are: (i) the number of such duplications is not known a priori, and (ii) the size of the problem increases. In our formulation, we allow multiple visits to a charging station without duplicating nodes. We propose five new operators for ALNS which are tested on 120 instances each of E-VRP-NL and E-VRP-NL-LD, and 80 instances of E-VRP-NL-LD-CCS. Results show that our ALNS outperforms the existing algorithms improving the solution in 63% of the instances and matching the best known solution in 31% of the instances. Results also show that considering load-dependent discharge is critical to optimally solve E-VRP.

Keywords: Adaptive large neighborhood search, Electric vehicle routing problem, Non-linear charging, Partial charging, Load-dependent discharging.

1. Introduction

According to the Paris agreement (UNCCC, 2018), 55 countries have agreed to reduce the overall Greenhouse Gases (GHGs) emissions by at least 55%, and many countries who are not part of Paris agreement have set goals to reduce GHGs up to 40% by 2030. It is expected that government worldwide may impose restrictions on the road transport sector to achieve these goals since this sector contributes to 20% of GHGs worldwide (International Energy Agency, 2018). Near zero well-to-wheel emissions can be achieved with Electric Vehicles (EVs) using electricity generated from renewable sources. Hence, EVs running on renewable energy is the foremost sustainable alternative to fossil-fuel driven vehicles. However, limited driving range and long charging times are hindering the growth of EVs (Hidrue et al., 2011; Carley et al., 2013). These issues also pose challenges to logistics firms' vehicle scheduling and route planning.

Recently, many studies (Schneider et al., 2014; Bruglieri et al., 2015; Keskin and Çatay, 2016; Montoya et al., 2017) have focused on the Electric Vehicle Routing Problem (E-VRP). E-VRP extends the classic Vehicle Routing Problem (VRP) by considering a limited driving range and charging times of Electric Vehicles (EV). The concept of refueling in vehicle routing problems is not new. The study by Ichimori et al. (1981, 1983) was the first to introduce the concept of vehicles taking detours to refuel en-route. E-VRP differs from the vehicle refueling problem since it considers longer "refueling" (charging) times. Conrad

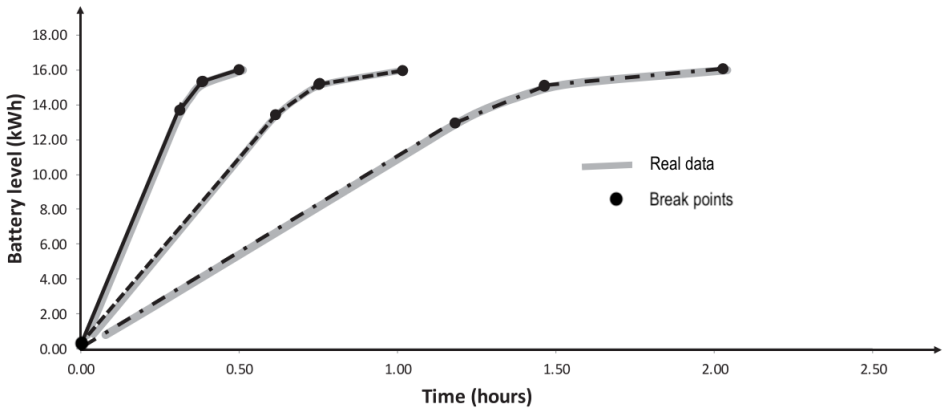
*Corresponding author
Email addresses: surendrareddy.kancharla@gmail.com (Surendra Reddy Kancharla), gitakrishnan@iitm.ac.in (Gitakrishnan Ramadurai)

and Figliozzi (2011) formulated a Recharging Vehicle Routing Problem (RVRP), where specific customer vertices also act as CSs. They assume that the charging of vehicles can happen while serving customers, and the quantum of recharge is a fixed proportion of battery capacity. As expected, their results showed that distance traveled and the number of vehicles increases with longer charging times and lower range. Schneider et al. (2014) is the first study to propose a formulation for the E-VRP with Time Windows (E-VRPTW). A limitation in their study is that vehicles always leave the CS fully charged. Instead of minimizing distance, the study by Bruglieri et al. (2015) considered battery level as a decision variable and minimized the weighted sum of total vehicles used, waiting time, charging time, and travel time. Unlike Schneider et al. (2014), they consider a partial charging policy. Their model improved the solution with respect to the total waiting time, charging time, and travel time for the same distance traveled and the number of vehicles used. Later, Keskin and Çatay (2016) proposed an Adaptive Large Neighborhood Search (ALNS) with Simulated Annealing (SA) acceptance criterion for the E-VRPTW considering partial charging. Based on their testing of ALNS on the E-VRPTW instances from Schneider et al. (2014), it is shown that allowing partial charging helps in achieving a better solution quality. Felipe et al. (2014) formulated a Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges (GVRP-MTPR) to minimize a weighted sum of charging costs at the depot, CSs, and fixed costs associated with battery cycles. They have considered a fixed stopping time at each CS, but a variable charging time based on the charging technology offered by the CS. This restricts the degree of partial recharges to a few preset levels. All the above studies used node-duplication and allowed unlimited simultaneous charging operations to use existing VRP models for solving E-VRP. Bruglieri et al. (2019) overcome the unlimited simultaneous charging operations issue by duplicating the chargers at CS in addition to CS itself.

Unlike the previous studies that considered a constant or linear charging rate, Montoya et al. (2017) realistically model the charging of vehicles via non-linear functions while accounting for various charging technologies. They call it the Electric Vehicle Routing Problem with Non-Linear charging (E-VRP-NL). E-VRP-NL assumes that all the CSs can handle an unlimited number of EVs simultaneously, EVs can carry an unlimited load, and EVs will always leave the depot with a full charge. Since battery capacity limits the range of EVs, they may have to take a detour en-route to visit a CS. These EVs can be recharged either fully or partially at any of the CSs. E-VRP-NL considers three types of CSs based on the charging technology: slow (11kW), moderate (22kW), and fast (44kW). Figure 1 shows the piecewise linear approximation of charging functions for a 16kWh battery. Instead of fixing the number of visits to a CS (β), Montoya et al. (2017) solved the Mixed-Integer Linear Programming (MILP) model iteratively by incrementing the values of β. They stop the iterations when there is no improvement between successive iterations or when the time runs out (100 hours). This methodology increases the complexity of the problem with an increase in β yet does not guarantee optimal solutions (Froger et al., 2019). Froger et al. (2019) introduced two formulations for E-VRP-NL. The first formulation uses arc-duplication similar to Koyuncu and Yavuz (2019), and as expected, it outperforms the traditional node-duplicating formulation. However, it still does not guarantee optimality. The second formulation is path-based, which allows multiple visits to a CS and guarantees optimality. But, it requires path enumeration, where the number of paths could grow exponentially and become intractable for larger instances. They report solutions only up to 20 customer instances. They also proposed a labeling algorithm for finding the optimal charging decisions for a given route.

All the studies mentioned above assume a linear discharge with distance irrespective of the load carried. This is an unrealistic assumption and does not hold good in a real-world scenario. Considering load in discharge calculations can significantly affect paths where heavier shipments are offloaded earlier to reduce the overall energy consumption (Suzuki, 2011). In this paper, we propose an arc-based MILP formulation that can be solved directly using standard optimizers and can find feasible solutions for up to 40 customer instances. Moreover, our model considers load in discharge calculations and allows multiple visits to a CS without node duplication. We also propose an ALNS heuristic that simultaneously determines the routes and charging decisions. We extend the proposed ALNS heuristic to consider capacity restrictions at charging stations.

The main contributions of this paper are: *i*) introduction of a new variant of E-VRP with load-dependent discharge and non-linear charging; *ii*) formulation of an arc-based three index Mixed-Integer Linear Pro-



Type of CSs	Slow	Moderate	Fast
Charging power (kWh/h)	11	22	44
Piecewise linear approximation	— · —	- - - - - -	———

Figure 1: Piecewise linear approximation for different types of CS charging an EV with a battery of 16 kWh (adopted from Montoya et al. (2017))

.

gramming (MILP) model for E-VRP-NL-LD that allows multiple visits to a CS; *iii*) demonstration of better performance of the proposed MILP for E-VRP-NL in comparison to existing models; *iv*) development of a modified ALNS algorithm delivering improved performance with new removal and insertion operators specific to E-VRP-NL-LD; and *v*) demonstration of the importance of considering load-dependent discharge through computational results.

The organization of the paper is as follows: section 3 introduces a MILP for E-VRP-NL-LD; section 4 introduces the ALNS heuristic for E-VRP-NL-LD. Section 5 presents the test instances and discusses the computational results. Finally, section 6 describes the conclusions.

2. Estimation of Power Required

For a vehicle to move from a state of rest, it has to overcome frictional resistance, drag resistance, and rolling resistance. The power required to overcome these resistances is estimated as follows:

$$P = \frac{(Ma + Mg\sin\theta + MgC_r\cos\theta + 0.5C_d\rho Av^2)v}{1000\epsilon}, \tag{1}$$

where v is the speed (m/s), a is acceleration (m/s^2), M 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 $(kg/m^3$, typically 1.2041), A is the frontal surface area (m^2), C_d is the coefficient of aerodynamic drag, C_r is the coefficient of rolling resistance, ϵ is the vehicle drive train efficiency, P is the second-by-second power required to overcome the resistances (kW). We consider that all vehicles travel at a fixed speed (40 kmph) and on level ground ($\theta = 0$). Upon substituting these values, equation (1) reduces to

$$P = \frac{(MgC_r + 0.5C_d\rho Av^2)v}{1000\epsilon}, \tag{2}$$

Equation (2) is rewritten as a linear function of load in equation (3):

$$P = \Gamma + \Xi M \tag{3}$$

Where, $\Gamma = \dfrac{0.5 C_d \rho A v^3}{1000\epsilon}$ is a constant and $\Xi = \dfrac{g C_r v}{1000\epsilon}$ is the coefficient of weight, M.

We use the values mentioned in table 1 for the parameters discussed above. These values were selected such that they are close to real-world data and at the same time energy consumption for a vehicle with half-load matches the consumption rate of 125 wh/km (used in Montoya et al. (2017) instances). The consumption rate of 125 wh/km is used in cases where the load is not considered.

Table 1: Vehicle parameters used in energy consumption estimation

ρ	C_d	A	v	ϵ	g	C_r
1.2041	0.48	2.3301	11.112	0.89	9.81	0.01

The values of Γ and Ξ can be obtained by using the data mentioned in the above table, and the only unknown gross vehicle weight (M) is a variable in the problem formulation.

3. Problem Description

The Electric Vehicle Routing Problem with Non-Linear charging and Load-Dependent discharging (E-VRP-NL-LD) is defined on a directed graph $G = (V, E)$, where $V = \{0\} \cup C \cup F$ is the set of vertices and $E = \{(i, j) : i, j \in V, i \neq j\}$ be the set of edges connecting vertices of V. The set of vertices has three subsets: C is the set of customers, F is the set of Charging Stations (CSs), and $\{0\}$ denotes the depot. The depot has an unlimited fleet of homogeneous EVs with a battery of capacity Q (expressed in kWh) and maximum tour duration of T_{max}. All the chargers in a CS are of the same type. Let $B = \{0, 1, 2, 3\}$ be the set of breakpoints associated with the piecewise linear approximation of the charging curve (as shown in Figure 1). Let g_{lb} and a_{lb} be the charging time and state of charge corresponding to breakpoint $b \in B$ at a CS l. Let s_i and Δ_i be the service time and demand at a customer vertex i respectively. Let τ_{ij} be the travel time between vertices i and j

Our MILP formulation uses the following decision variables: variable x_{ij}^l is a binary variable and is equal to 1 only if an EV travels from vertex i to j via $l \in F \cup \{D\}$. Note that variable x_{ij}^D represents a direct arc from i to j. Variables t_{ij}^l, u_{ij}^l and y_{ij}^l track the time, load, and state of charge when an EV arrives at vertex $j \in C^\dagger (C^\dagger = C \cup \{0\})$. Variables σ_{ij}^l and ς_{ij}^l specify the charge levels when an EV arrives at and departs from $l \in F$, and ϕ_{ij}^l and π_{ij}^l are the associated charging times. Variable $\lambda_{ij}^l = \pi_{ij}^l - \phi_{ij}^l$ gives the charging time of a vehicle traveling from i to j stopping en route at a CS $l \in F$. Variables ϖ_{ij}^{lb} and ρ_{ij}^{lb} are equal to 1 if the charge level is between $a_{l,b-1}$ and $a_{l,b}$, with $b \in B \backslash \{0\}$, when an EV arrives at and departs from CS $l \in F$, respectively. Finally, variables α_{ij}^{lb} and γ_{ij}^{lb} are fractional values which help in determining the part of the piecewise linear approximation in which the arrival and departure charges are located. The MILP formulation follows:

$$\min \sum_{i \in C^\dagger} \sum_{j \in C^\dagger} \left(x_{ij}^D \tau_{ij} + \sum_{l \in F} \left(x_{ij}^l (\tau_{il} + \tau_{lj}) + \lambda_{ij}^l \right) \right) \tag{4}$$

Subject to constraints:
for visiting customers exactly once:

$$\sum_{\substack{j \in C^\dagger \\ i \neq j}} \sum_{l \in F \cup \{D\}} x_{ij}^l = 1 \; \forall i \in C \tag{5}$$

for conserving flow:

$$\sum_{\substack{j\in C^\dagger\\ i\neq j}}\sum_{l\in F\cup\{D\}} x_{ij}^l = \sum_{\substack{j\in C^\dagger\\ i\neq j}}\sum_{l\in F\cup\{D\}} x_{ji}^l \,\forall i\in C \tag{6}$$

for tracking arrival time:

$$\sum_{\substack{j\in C^\dagger\\ j\neq i}}\left(\sum_{l\in F\cup\{D\}} t_{ij}^l - \tau_{ij}x_{ij}^D - \sum_{m\in F}((\tau_{im}+\tau_{mj})x_{ij}^m + \lambda_{ij}^m)\right) \geq \sum_{\substack{j\in C^\dagger\\ j\neq i}}\sum_{n\in F\cup\{D\}} t_{ji}^n + s_i \,\forall i\in C \tag{7}$$

$$\sum_{l\in F\cup\{D\}} t_{0j}^l - \tau_{0j}x_{0j}^D - \sum_{m\in F}((\tau_{0m}+\tau_{mj})x_{oj}^m + \lambda_{0j}^m) \geq 0 \,\forall j\in C \tag{8}$$

$$t_{ij}^l \leq T_{max}x_{ij}^l \;\forall i\in C^\dagger, j\in C^\dagger, l\in F\cup\{D\} \tag{9}$$

for tracking state of charge:

$$y_{ij}^l \leq Qx_{ij}^l \;\forall i\in C^\dagger, j\in C^\dagger, l\in F\cup\{D\} \tag{10}$$

$$y_{0j}^D = (Q-\Gamma)x_{0j}^D - \Xi u_{0j}^D \;\forall j\in C \tag{11}$$

$$y_{0j}^l = \varsigma_{0j}^l - (\Gamma x_{0j}^l + \Xi u_{0j}^l) \;\forall l\in F, j\in C \tag{12}$$

$$\sum_{l\in F}(y_{ij}^l + \Gamma x_{ij}^l + \Xi u_{ij}^l) \leq \sum_{l\in F}\varsigma_{ij}^l \;\forall i\in C, j\in C^\dagger \tag{13}$$

$$\sum_{\substack{j\in C^\dagger\\ i\neq j}}(y_{ij}^D + \Gamma x_{ij}^D + \Xi u_{ij}^D) \leq \sum_{\substack{j\in C^\dagger\\ i\neq j}}\sum_{m\in F} y_{ji}^m \;\forall i\in C \tag{14}$$

for determining charge on arrival:

$$\sum_{\substack{i\in C^\dagger\\ i\neq j}}(\sigma_{ji}^l + \Gamma x_{ji}^l + \Xi u_{ji}^l) \leq \sum_{\substack{i\in C^\dagger\\ i\neq j}}\sum_{m\in F\cup\{D\}} y_{ij}^m \;\forall j\in C, l\in F \tag{15}$$

$$\sigma_{ji}^l \leq (Q-\Gamma)x_{ji}^l - \Xi u_{ji}^l \;\forall i\in C^\dagger, j\in C^\dagger; i\neq j, l\in F \tag{16}$$

$$\sum_b \alpha_{ij}^{lb}a_{lb} = \sigma_{ij}^l \;\forall i\in C^\dagger, j\in C^\dagger l\in F \tag{17}$$

$$\phi_{ij}^l = \sum_b \alpha_{ij}^{lb}g_{lb} \;\forall i\in C^\dagger, j\in C^\dagger, l\in F \tag{18}$$

$$\alpha_{ij}^{lb} \leq x_{ij}^l \;\forall i\in C^\dagger, j\in C^\dagger, l\in F, b\in B \tag{19}$$

$$\varpi_{ij}^{lb} \leq \alpha_{ij}^{lb} \;\forall i\in C^\dagger, j\in C^\dagger, l\in F, b\in B/\{3\} \tag{20}$$

$$\alpha_{ij}^{lb} \leq \varpi_{ij}^{l(b-1)} \;\forall i\in C^\dagger, j\in C^\dagger, l\in F, b\in B/\{0,1\} \tag{21}$$

for determining charge at departure:

$$\varsigma_{ij}^{l} \leq Qx_{ij}^{l} \ \ \forall j \in C, l \in F, i \in C^{\dagger} \tag{22}$$

$$\varsigma_{ij}^{l} \geq \sigma_{ij}^{l} \ \ \forall j \in C, l \in F, i \in C^{\dagger} \tag{23}$$

$$\sum_{b} \gamma_{ij}^{lb} a_{lb} = \varsigma_{ij}^{l} \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F \tag{24}$$

$$\pi_{ij}^{l} = \sum_{b} \gamma_{ij}^{lb} g_{lb} \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F \tag{25}$$

$$\gamma_{ij}^{lb} \leq x_{ij}^{l} \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F, b \in B \tag{26}$$

$$\rho_{ij}^{lb} \leq \gamma_{ij}^{lb} \ \ \forall i \in C^{\dagger}, l \in F, b \in B/\{3\} \tag{27}$$

$$\gamma_{ij}^{lb} \leq \rho_{ij}^{l(b-1)} \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F, b \in B/\{0,1\} \tag{28}$$

for determining charging time:

$$\lambda_{ij}^{l} = \pi_{ij}^{l} - \phi_{ij}^{l} \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F \tag{29}$$

for tracking load on vehicle:

$$\sum_{\substack{j \in C^{\dagger} \\ i \neq j}} \sum_{l \in F \cup \{D\}} u_{ji}^{l} - \sum_{\substack{j \in C^{\dagger} \\ i \neq j}} \sum_{l \in F \cup \{D\}} u_{ij}^{l} = \Delta_{i} \ \ \forall i \in C \tag{30}$$

$$\sum_{\substack{i \in C^{\dagger} \\ i \neq j}} \sum_{l \in F \cup \{D\}} u_{i0}^{l} = 0 \tag{31}$$

on domain of variables:

$$x_{ij}^{l} \in \{0,1\}; \ \ y_{ij}^{l} \geq 0; \ \ t_{ij}^{l} \geq 0; \ \ u_{ij}^{l} \ \geq 0 \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F \cup \{D\} \tag{32}$$

$$\lambda_{ij}^{l} \geq 0; \ \ \sigma_{ij}^{l} \geq 0; \ \ \varsigma_{ij}^{l} \geq 0; \ \ \pi_{ij}^{l} \geq 0; \ \ \phi_{ij}^{l} \geq 0 \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F \tag{33}$$

$$\varpi_{ij}^{lb} \in \{0,1\}; \ \ \rho_{ij}^{lb} \in \{0,1\}; \ \ \alpha_{ij}^{lb} \in [0,1]; \ \ \gamma_{ij}^{lb} \in [0,1]; \ \ \forall i \in C^{\dagger}, j \in C^{\dagger}, l \in F, b \in B \tag{34}$$

The objective function (4) seeks to minimize the total time (travel times plus charging times). Constraints (5) ensure that each customer is visited exactly once. Constraints (6) ensure flow conservation at each customer. Constraints (7) ensures the time of departure at a vertex is no earlier than the sum of arrival time and the service time. Constraints (8) ensures the arrival time at the first vertex from the depot is greater than the travel time plus charging time if any. In the constraints (7) and (8) the term $\tau_{ij} x_{ij}^{D}$ will be non-zero only if the vehicle travels directly between i and j. Similarly, the term $\sum_{m \in F}((\tau_{im} + \tau_{mj}) x_{ij}^{m} + \lambda_{ij}^{m})$ will be non-zero only if the vehicle passes through a CS (m) while traveling between vertices i and j. Constraints (7) and (8) also help in sub-tour elimination. Constraints (9) ensures the maximum tour duration is not exceeded. Constraints (10) ensures that the maximum state of charge is not violated. Constraints (11) and (12) ensure that the charge on arrival at the first customer should be the difference of either full charge and charge required to travel from the depot to the first customer or departure charge from a CS and the charge required to travel from the CS to the first customer. Similarly, the constraints (13) and (14) ensure the arrival charge at a customer is always less than or equal to the difference of departure charge at previous node (either a CS or a customer) and charge required to travel from the previous node to the customer. Constraints (15) ensures the sum of arrival charge at a CS and the charge required to reach the CS from the previous node is always less than or equal to the departure charge at the previous node. Constraints (16) ensures the arrival charge at a CS is less than or equal to the difference of the maximum state of charge and the charge required to reach the CS from the previous node. In constraints (11)-(16), load-dependent discharging is included by splitting the charge required into two terms i) charge required by an empty vehicle

(Γx_{ij}^{l}) and ii) extra charge required for the load carried (Ξu_{ij}^{l}). Constraints (17) gives the fractional values of α_{ij}^{lb} based on the arrival charge, which is later used in constraints (18) to get the time required to charge from zero to the arrival charge of EV at the CS. Constraints (19)-(21) ensure the value of $\alpha_{ij}^{l(b+1)}$ is non-zero only if α_{ij}^{lb} is one. Constraints (22) and (23) ensure the departure charge of an EV from CS $l \in F$ can be at most equal to the maximum state of charge and at least equal to the arrival charge at the CS, respectively. Constraints (24) gives the fractional values of γ_{ij}^{lb} based on the departure charge at a CS that is later used in constraints(25) to get the time required to charge from zero to the departure charge of EV at the CS. Constraints (26)-(28) ensure the value of $\gamma_{ij}^{l(b+1)}$ is non-zero only if γ_{ij}^{lb} is one. Constraints (29) define the charging time spent at any CS. Constraints (30) ensures that the difference in the load carried by vehicle from arrival to departure at a vertex is equal to its demand. Constraints (31) ensure that no load is carried by a vehicle while returning to the depot. Finally, constraints (32) - (34) define the domain of the decision variables. The E-VRP-NL-LD directly reduces to E-VRP-NL. Assigning $\Delta_i = 0 \ \forall i \in C$, Γ = constant discharge rate, and $\Xi = 0$ results in the E-VRP-NL formulation that is used for comparing with models in the literature.

4. Solution method

Ropke and Pisinger (2006a) introduced the Adaptive Large Neighborhood Search (ALNS) as an extension of the Large Neighborhood Search (LNS) (Shaw, 1998). Unlike LNS, ALNS can adapt to the problem based on its performance in previous iterations and is one of the most effective heuristics for large-scale VRPs. Ropke and Pisinger (2006b) later proposed a unified ALNS approach for several variants of VRP. Since then, it has been widely used for various variants of VRP. For example, the pollution-routing problem (Demir et al., 2012), two-echelon VRP (Hemmelmayr et al., 2012), VRP with multiple routes (Azi et al., 2014), and electric VRP (Keskin and Çatay, 2016; Zhang et al., 2020) all use ALNS variants to solve the problem.

The earlier ALNS implementation for E-VRP (Keskin and Çatay, 2016) cannot handle partial charging, different charging technologies, and capacitated charging stations. When the CSs have the same charging technology, the share of charge at each CS does not affect the overall time. However, if the CSs have different technologies, the percentage of charge and the sequence of visits affect the total time. For example, if a route includes visits to a fast and a slow CS, visiting the fast CS earlier in the sequence may allow for a longer charge in less time and hence reduce the charging time at the slow CS. This reduction in charging time could be advantageous as opposed to visiting later in the sequence. When the algorithm processes the route from the starting depot, the minimum charge required to complete the route is unknown if there are two or more CS visits en-route. Hence, we process our solutions from the end depot and build the route back to the starting depot. In routes with two or more CS, we assume that vehicles always arrive at a CS with zero charge. Charging a vehicle with a quantum of charge that is more than what is required to reach the next CS will only worsen the objective value.

4.1. ALNS algorithm

We propose a modified ALNS algorithm for E-VRP-NL that can handle partial charging, different charging technologies, and capacity restrictions. We describe the steps of ALNS are demonstrated here. We provide an initial solution to the ALNS. Subsequently, we either remove q customers from the solution and re-insert them to create a new solution or remove r CSs from the solution and insert new CSs wherever required. The insertion and removal operators are selected based on their past success using the roulette wheel selection method. Since the weights of operators are updated based on their past performance, the operators with higher weights have a higher probability of being chosen again.

Simulated annealing (SA) acceptance criteria for a new solution is implemented as follows: we always accept when the total time of the new solution is lower than the current best solution; otherwise, we accept it with a probability $e^{-(f(S_{new})-f(S_{best}))/kT}$, where $f(S)$ is the objective value of the solution S. T is the temperature, and k is the Boltzmann constant. Temperature T is reduced by a temperature reduction factor (η) at the end of I_{temp} iterations at the current temperature. The ALNS stops either when the number

of non-improvement temperature decrements exceeds a specified number (150 in our study) or after having run maximum allowed iterations. Algorithm 1 presents the pseudo-code of the proposed ALNS algorithm.

Algorithm 1 Adaptive Large Neighborhood Search

Read input data, initialize weights and scores
Generate an initial solution using greedy heuristic (S)
Make initial solution as the best solution $S^* \leftarrow S$
$p = 0$
for $i \leftarrow 0, I_{max}$ **do**
    for $j \leftarrow 0, I_{temp}$ **do**
        if j % $N_s = 0$ **then**
            Use roulette wheel selection to select station removal and station insertion operators
            Solution after applying station removal operator on S is S_{ip}
            New solution after applying the station insertion operator on S_{ip} is S_i
        else
            if j % $N_{rr} = 0$ **then**
                Use roulette wheel selection to select customer removal operator
                Solution after applying customer removal operator on S is S_{ip}
            else
                Use roulette wheel selection to select route removal operator
                Solution after applying route removal operator on S is S_{ip}
            end if
            Use roulette wheel selection to select customer insertion operator
            New solution after applying the customer insertion operator on S_{ip} is S_i
        end if
        if $f(S_i) \leq f(S)$ **then**
            $S \leftarrow S_i$
            Update scores of operators
        else if $R \leq e^{-(f(S_i)-f(S^*))/kT}$ **then**
            $S \leftarrow S_i$
            Update scores of operators
        end if
    end for
    if $f(S) < f(S^*)$ **then**
        $S^* \leftarrow S$
        $p = 0$
    else
        $p = p+1$
    end if
    Update weights of operators based on scores
    Update Temperature ($T = \eta T$)
    if $p = \Theta$ **then**
        Stop
    end if
end for

Note: S - incumbent solution; S^* - best solution; p - counter for consecutive non improvement iterations; N_s- Station operation interval; N_{rr}- Route removal interval; R - random number $\in [0, 1]$; η - temperature reduction factor; Θ - maximum number of consecutive non improvement iterations

4.1.1. Initial solution

We construct the initial solution by assigning each customer to a vehicle. The cases where the charge in the vehicle is insufficient to make a return trip to the inserted customer, a CS is inserted using greedy station insertion, as explained in subsection 4.1.6.

4.1.2. Customer removal operators

We use the three customer removal operators introduced by Ropke and Pisinger (2006b) - random removal, worst distance removal, and related removal. All these operators remove q customers from their existing routes and add them to a customer pool. The value of q is randomly selected in the range $[min(30, 0.1n_c), min(60, 0.4n_c)]$, where n_c is the total number of customers in the network. The procedure of removing q customers from their current routes varies from operator to operator, as explained below.

In random removal, we remove q customers randomly from their current position and add them to the customer pool. In worst distance removal, we first calculate the removal gain, defined as the reduction in distance traveled in a particular route with and without the customer, for each customer and then remove the customer with the highest removal gain. We repeat this process until q customers are removed from their existing positions and added to the customer pool. In related removal, we remove a random customer along with $q - 1$ of its nearest customers.

4.1.3. Route removal operators

We use random route removal (Hemmelmayr et al., 2012) or greedy route removal (Keskin and Çatay, 2016) operator after every N_{rr} iterations based on roulette wheel selection. These operators remove p routes from the solution and then adds all the customers from those routes to the customer pool. We randomly choose the value of p in the range $[0.1n_r, 0.4n_r]$, where n_r is the total number of routes in the present solution. In random route removal, we remove p random routes from the solution. Whereas in greedy route removal, we first sort the routes in the increasing order of the number of customers visited and then remove the first p routes from the solution.

4.1.4. Station removal operators

We use four different station removal operators: two newly proposed by us and two from Keskin and Çatay (2016). All these operators remove r ($min(10, 0.4n_cs)$) CSs from their current route, where n_cs is the number of CS visits.

In the first operator from Keskin and Çatay (2016), we randomly remove r CSs from their existing routes. In the second operator from Keskin and Çatay (2016), as well as the newly proposed third and fourth operators we sort the CSs in decreasing order of the charge available on arrival, the charge required to reach the CSs, and charging time spent at the CSs in those routes respectively. Subsequently, we remove the first r CSs on the sorted list from the solution.

4.1.5. Customer insertion operators

We use three customer insertion operators, greedy insertion, greedy insertion perturbation, and regret k-insertion introduced in Ropke and Pisinger (2006b). Unlike in Ropke and Pisinger (2006b), here these operators need to be followed by a greedy station insertion (explained in section 4.1.6) when a customer insertion makes a route infeasible due to insufficient charge in battery. These operators insert all the customers from the customer pool into the solution.

In greedy insertion, we compute the insertion cost, defined as the increase in distance traveled with and without a random customer picked from the customer pool, and then insert that customer at a position with the lowest insertion cost. Greedy insertion perturbation works similar to greedy insertion, except that we introduce some noise to the insertion cost by multiplying with a uniformly distributed random variable $\delta \in [0.8, 1.2]$. In regret k-insertion, we compute the regret costs, defined as the difference between the lowest insertion cost and the k^{th} ($k = 2$ in our case) lowest insertion cost, for all the customers in the customer pool. We then insert the customer with the highest regret cost at a position that leads to the lowest increase in objective value. We repeat the operator selected through roulette wheel selection until the customer pool becomes empty.

In case of routes with CS, customer insertion can happen at three different positions: i) between the origin depot and a CS; ii) between a CS and the destination depot, and iii) between two CSs. In the first case, the insertion will only affect the state of charge at the CS. In the second case, we will increase the departure state of charge at the CS such that the vehicle can visit the customer and reach the depot with zero charge. In the last case, we will increase the departure state of charge at the previous CS such that the vehicle can visit the customer. Such a customer insertion in any given route (with or without CSs) or starting a new route with a customer can lead to infeasibility in terms of the state of charge. In such cases, we perform a greedy station insertion. If the problem becomes infeasible by exceeding maximum route duration after the greedy station insertion, then we will insert the customer at the next best position possible.

4.1.6. Station insertion operators

We have introduced three new operators named greedy insertion, best insertion, and compare-k insertion that can tackle the capacity constraints at a charging station. One of these operators selected using roulette wheel selection is used immediately after the station removal operators to make the infeasible routes feasible with respect to the state of charge.

In greedy insertion operator, we identify the customer where the vehicle arrives with a negative charge and then insert the nearest CS between that customer and the previous customer. If this insertion is infeasible due to lack of charge or number of simultaneous charging operations, we insert a CS at a feasible position in the earlier edges. In best station insertion, we insert the CS between the customer with negative arrival charge and a previously visited CS or depot such that the increase in objective value is least, and at the same time, the number of simultaneous charging operations is not violated. Compare-k insertion works similar to best station insertion except that we check for only k insertion positions starting from the customer with a negative charge to previous CS or depot instead of all insertion positions.

4.1.7. Adaptive weights

We use a roulette wheel selection to identify the insertion and removal operators. Initially, all the operators have equal weights, and the score of all operators is zero. We update operators' scores at the end of every iteration based on their performance in the current iteration. These scores are used to update the weights of the operators before performing the temperature reduction, and then scores are reset to zero. Weights are updated as $W_o^i = W_o^{i-1} + \pi_o^i/\omega_i$, where W_o^i is the weight of operator o after i^{th} temperature reduction and π_o^i is the score of the operator o during i^{th} temperature reduction, ω_i is the sum of scores of all the operators in the respective category (customer insertion, customer removal, station insertion, and station removal) during the i^{th} temperature reduction.

5. Computational Tests

The MILP model is coded in GAMS 23.9 and solved using Gurobi 7.5 solver hosted on NEOS server (Czyzyk et al., 1998; Dolan, 2001; Gropp and Moré, 1997). The server runs on Intel Xeon E5-2430 @ 2.2GHz with 3 GB RAM (available for each job submitted). The ALNS algorithm is coded in Python and tested on a PC running on 3.6 GHz Intel Core-i7-7700 processor with 16 GB of RAM. Instances are described in subsection 5.1; MILP results in subsection 5.2 followed by parameter tuning in subsection 5.3 and ALNS results 5.4.

5.1. Instance sets

Montoya et al. (2017) introduced 120 instances for the E-VRP-NL using real data from EV configuration and battery charging times. These 120 instances are presented as six subsets: each having 20 instances with 10, 20, 40, 80, 160, and 320 customers. Instances in each of these subsets vary by the number of CSs available (low and high availability) and types of charging technologies (slow, moderate, and fast charger). In all the instances, EV has a charge consumption rate of 0.125kWh/km, a battery capacity of 16kWh, and a maximum route duration of 10h. We also have created 120 new instances for E-VRP-NL-LD that include

customer freight demand by modifying the Montoya et al. (2017) instances. The demands (in kg) used are generated by drawing a random value in the range [50, 250], and the capacity of the EV is assumed to be 600. The same instances created for E-VRP-NL-LD are used for E-VRP-NL-LD-CCS with additional data on the number of simultaneous charging operations possible at each CS.

5.2. Gurobi results

Table 2 shows the results for ten, twenty, and forty customer instances for two E-VRP-NL models from the literature and the present model. In this table, the first column represents the number of customers ($|C|$). The second column indicates the model. The third column ($\#Unk$) lists the number of instances where the solver could not find a feasible solution within the time limit of three hours. The fourth column ($\#\overline{Opt}$) lists the number of instances where the solver found a feasible but not optimal solution within the time limit. The fifth column ($\#Opt$) lists the number of instances where the solver found an optimal solution. Finally, column six lists the average percentage improvement in the objective value over the Montoya et al. (2017) model.

The results reported by Montoya et al. (2017) were the incumbent solutions obtained by running Gurobi (version 5.6) for 100 hours. The longer run time is perhaps because they solve the problem repeatedly for different values of β (the maximum number of visits to a CS). Froger et al. (2019) obtained their solutions using Gurobi (version 7.5) at the end of three hours. Similarly, we obtained our using Gurobi (version 7.5) at the end of three hours. Montoya et al. (2017) found a feasible solution in 18 out of 20 cases for 20 customer instances and only 7 out of 20 cases for 40 customer instances whereas our model found a feasible solution for all the 20 cases in 20 customer instances and 12 out of 20 cases in 40 customer instances. Froger et al. (2019) had presented two arc-based formulations and one path based formulation. Among the two arc-based formulations, the one that uses node repetition performed worse than Montoya et al. (2017) mainly because it was run only for 3 hours, whereas Montoya et al. (2017) ran their model for 100 hours. The arc-based formulation performed relatively better, but still worse than Montoya et al. (2017). Our arc-based formulation outperformed the formulation of Montoya et al. (2017) and the two arc-based formulations of Froger et al. (2019). Compared to our results, the path-based formulation of Froger et al. (2019) reported better bounds for 20 customer instances. Froger et al. (2019) did not report results for 40 customer instances. However, their formulation requires path enumeration as a pre-processing step, whereas our formulation can be solved directly with any standard optimizer. Performing additional runs, we were able to match the results in Froger et al. (2019) by extending our run time to 8 hours. Appendix A presents the detailed results for each instance.

Table 2: Comparison of results of different models on 10, 20 and, 40 customer instances

$\|C\|$	**Model**	$\#Unk$	$\#\overline{Opt}$	$\#Opt$	**Improvement (%)**
10	Montoya et al. (2017)	0	0	20	-
10	Froger et al. (2019) (Node-rep)	0	11	9	-0.39
10	Froger et al. (2019) (Arc-rep)	0	4	16	0.00
10	Froger et al. (2019) (Path)	0	0	20	0.00
10	**Present model**	**0**	**0**	**20**	**0.00**
20	Montoya et al. (2017)	2	12	6	-
20	Froger et al. (2019) (Node-rep)	5	15	0	-0.6
20	Froger et al. (2019) (Arc-rep)	5	10	5	0.94
20	Froger et al. (2019) (Path)	0	15	5	1.11
20	**Present model**	**0**	**15**	**5**	**0.46**
40	Montoya et al. (2017)	13	7	0	-
40	**Present model**	**8**	**12**	**0**	**4.39**

Note: Node-rep is the formulation with node repetitions, Arc-rep is the formulation with arc repetitions, and Path is the path-based formulation

Table 3 shows the Gurobi results for the E-VRP-NL-LD instances obtained at the end of three-hour time limit. In this table, the first column represents the number of customers ($|C|$). The second column lists the number of instances where the solver could not find a feasible solution within the time limit. The third column lists the number of instances where the solver found a feasible but not optimal solution within the time limit, and the final column lists the number of instances where the solver found an optimal solution. Appendix B presents the detailed results for each instance. As expected, the performance of Gurobi with E-VRP-NL-LD is poorer compared to E-VRP-NL.

Table 3: Gurobi results for E-VRP-NL-LD instances

$\|C\|$	$\#Unk$	$\#\overline{Opt}$	$\#Opt$
10	1	2	17
20	1	17	2
40	12	8	0

5.3. Parameter tuning

The performance of heuristic algorithms largely depends on the parameters used. Our ALNS algorithm depends on the following parameters: maximum iterations (I_{max}), maximum iterations at a temperature (I_{temp}), temperature reduction factor (η), initial temperature (T_i), route removal operation interval (N_{rr}), station operations interval (N_s), score for the best update (σ_1), score for better solution (σ_2), and score for best solution (σ_3). We have tested the 20 customer instances with six values of I_{temp} (100, 200, 500, 800, 1000, 1200), five values of T_i (10, 20, 30, 50, 100), and four values of η (0.9, 0.95, 0.98, 0.99) for a total of 120 combinations. These combinations are ranked based on average solution obtained for the tested instances, and the combinations with the same average solution were ranked based on run times. The final combination that gave the best results is retained, and those were $I_{temp} = 800$; η =0.98; $T_i = 30$. The remaining parameters are taken from literature and are as follows: $N_{rr} = 50$; $N_s = 11$; $\sigma_1 = 45$; $\sigma_2 = 10$, $\sigma_3 = 30$, and $\Theta = 150$. We have fixed the value of I_{max} as 500000 (625 iterations in the outer loop each with I_{temp} iterations) similar to Hemmelmayr et al. (2012) and Li et al. (2015). However, in most cases algorithm will stop earlier than these iterations because of exceeding the limit for the number of consecutive non-improvement temperature decrements.

5.4. ALNS results for E-VRP-NL

We evaluate the performance of ALNS is evaluated on all the 120 E-VRP-NL instances introduced by Montoya et al. (2017). ALNS is run ten times for each of these instances, and we compare the results are compared with the Best Known Solution (BKS) from literature (Montoya et al., 2017; Froger et al., 2019). In table 4, "ALNS-Best" represents the best objective value obtained by ALNS during the ten runs and "ALNS-Average" represents the average of the objective values obtained during the ten runs. ALNS matched BKS in 38 instances, and of the remaining 81 instances, ALNS clearly outperformed the existing algorithms by improving the objective function value in 75 instances. Appendix C presents the detailed results of ALNS for each instance.

We have run ALNS with and without the new operators and evaluated the performance by analyzing the variation in the gap compared to BKS for the following: *i*) best solutions without new operators, *ii*) average solutions without new operators, *iii*) best solution with new operators, and *iv*) average solution with new operators. The run times for both cases are similar. ALNS with new operators always resulted in better solutions in more than 80% of the instances compared to cases without new operators. Average improvement in solutions is 0.13%, 0.95%, 1.13% and 0.69% for 40, 80, 160, and 320 customer instances respectively. For instances with customers of 10 and 20, there was no difference in with and without new operators. Figure 2 compares the gaps in all the cases mentioned above. The best solutions for ALNS with new operators is always better than ALNS without new operators.

Table 4: Performance of ALNS

Customers	BKS	ALNS-Average	ALNS-Best	Improved	Matched BKS	Optimal
10	14.25	14.25	14.25	0	20	20
20	19.44	19.46	19.4	9	10	6
40	31.48	31.5	31.3	13	4	0
80	38.01	37.42	37.10	19	1	0
160	70.24	69.00	68.16	20	0	0
320	132.47	133.90	131.62	14	3	0
Total				**75**	**38**	**26**

Performance of ALNS in 320 customer instances is less remarkable than instances with 160 customers. This could be because of the restriction on the removal operators to remove max(60,0.4n_c) vertices (includes both customer and CSs) from the solution. To test the effect of this restriction on the solution quality, we increased this limit to 100 vertices in 320 customer instances. The relaxation led to an average improvement of only 0.2% over the 60 vertex limit but resulted in an increase in run time by 43%. This is a trade-off between run time and solution quality, and we report the result with 60 vertices limit since the improvements with 100 vertices are not significant.

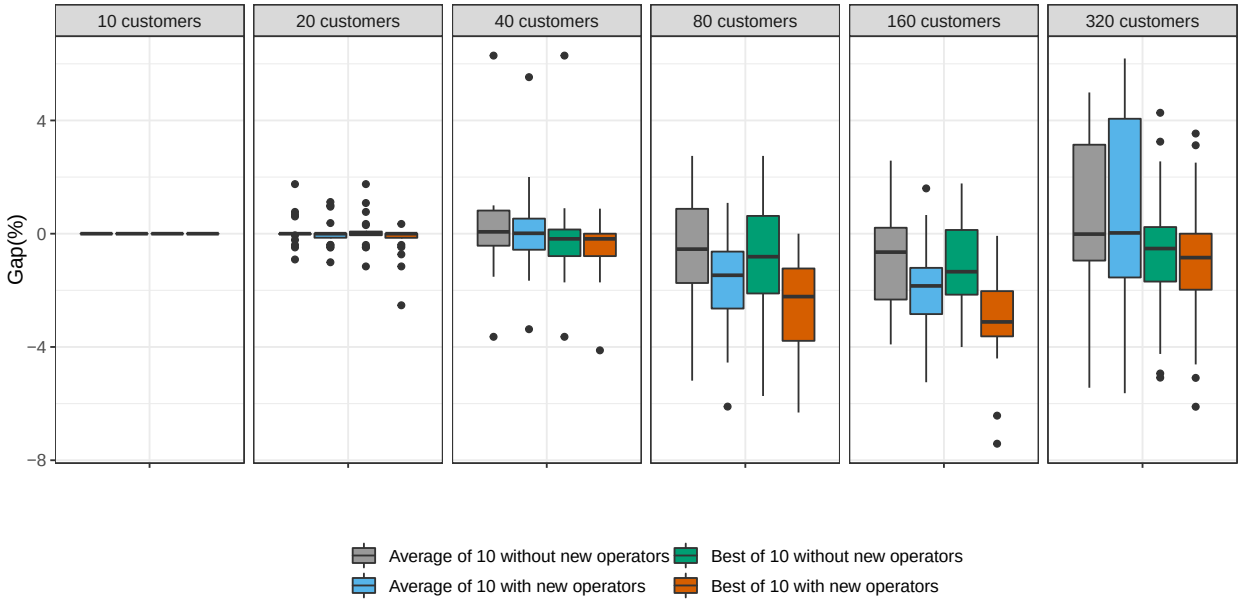


Figure 2: Variation in the gap for best solutions without new operators, best and average solutions with new operators compared to BKS.

5.5. ALNS results for E-VRP-NL-LD

We solve the newly introduced instances with and without considering the effect of load in energy estimation (refer to Table 5). The increase in the number of vehicles used and total time is insignificant for instances having up to 20 customers. However, beyond 20 customers, there is a considerable increase. Using a load-dependent discharge in the energy estimation results in higher discharge that leads to longer charging times. The increase in charging time results in an increase in the number of vehicles to avoid the violation of tour duration constraint. This increment leads to a rise in empty trips, thereby increasing the travel time and total time. See Appendix D for the detailed results of ALNS for each instance.

Table 5: Effect of considering the load on optimal routes

Instance size	With load		Without load		% dev in cost	% dev in vehicles
	Avg. Cost (i)	**Avg. Vehicles** (ii)	**Avg. Cost** (iii)	**Avg. Vehicles** (iv)	$\frac{(i-iii)}{i} * 100$	$\frac{(ii-iv)}{ii} * 100$
10	20.69	4.25	19.62	4.20	5.45	1.19
20	29.35	7.05	28.90	7.00	1.56	0.71
40	39.48	7.80	31.43	6.25	25.61	24.80
80	46.55	10.35	37.20	8.50	25.13	21.76
160	106.71	21.05	68.21	15.90	56.44	32.39
320	218.50	40.65	131.77	30.55	65.82	33.06

To evaluate the effect of ignoring load, we compare the number of CSs used, the maximum number of CSs per route, charging times, and travel time for both the cases - with and without considering load. First, we compare the average number of CS visits in figure 3, and it is clear that the number of visits increased significantly with instance size in cases considering load compared to without load. Even the maximum number of CS visits per route is higher in the cases considering load over cases without considering load (refer figure 4).

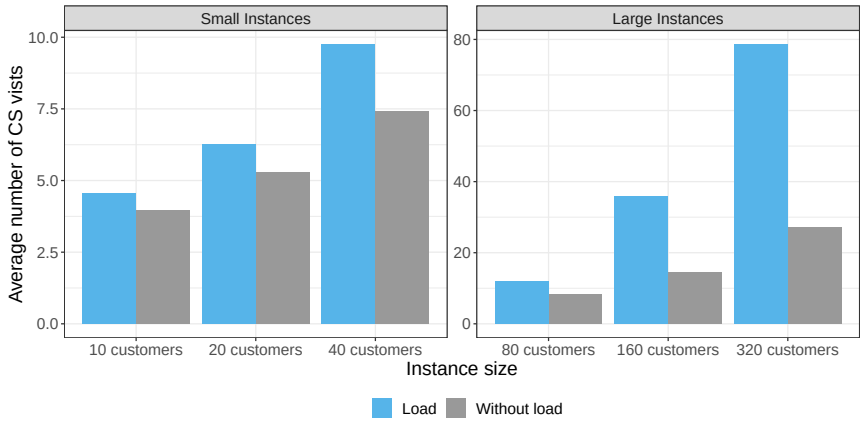


Figure 3: Average number of CS visits

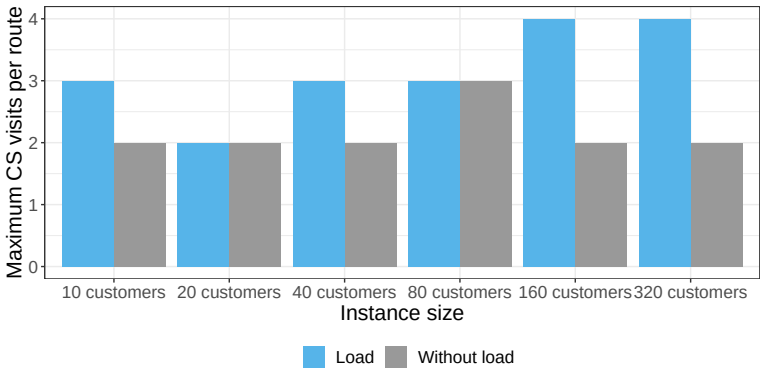


Figure 4: Maximum number of CS visits per route

We compare the charging times and travel times for both cases - with and without considering load (refer figures 5 and 6). The average increase in charging time for cases with load is around 15-20% for up

to 20 customer instances, whereas it is in the range of 50-80% for instances with 40 or more customers. We observe a similar trend in the case of travel time; however, the magnitude is lower. The average increase is around 2% for instances up to 20 customers and 14-30% for 40 or more customers.

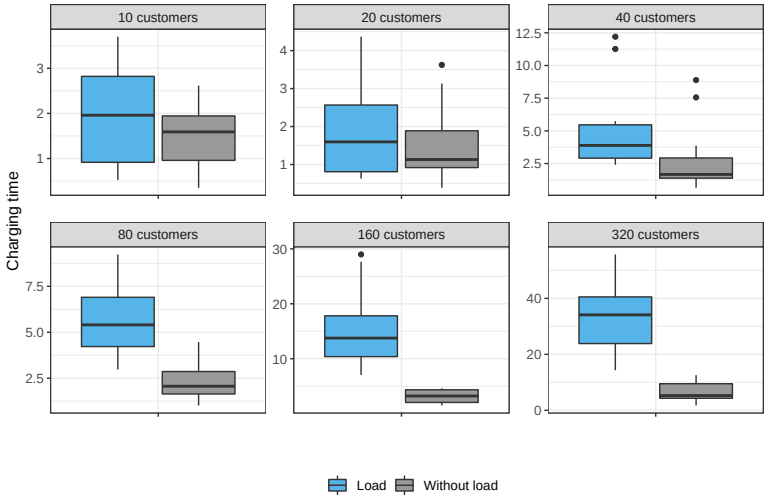


Figure 5: Comparison of variation in charging times

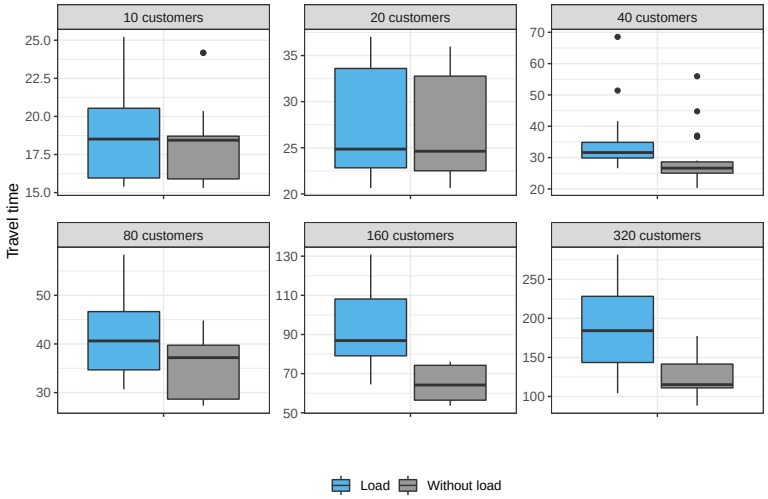


Figure 6: Comparison of variation in travel times

5.6. ALNS results for E-VRP-NL-LD-CCS

We solve the newly introduced E-VRP-NL-LD instances along with a restriction on the number of simultaneous charging operations at a CS. All the solutions without such a restriction had no more than five simultaneous visits to a CS. Most of the instances with customers equal to or more than 160 were infeasible when we restrict simultaneous visits to CS to less than four. Hence, we are using cases with less than 160 customers to evaluate the effect of this restriction.

Sensitivity analysis was performed on the number of simultaneous charging operations (see table 6) by restricting the number of simultaneous visits to two, three, and four. However, there is no change in results

when we set the limit to four. In table 6, the first column has the instance names. Columns two, three, and four has the results with the number of simultaneous visits restricted to two, three, unlimited (Un) or four, respectively. Columns five, six, and seven has percentage deviation in objective value for 2 vs 3 visits, 2 vs unlimited visits, 3 vs unlimited visits, respectively. The rows with gray color indicate those instances where the solutions in the unlimited case had at least one charging station with no more than three simultaneous visits, whereas the remaining rows had no more than two visits. A few of the instances had resulted in the same objective value even after the restriction, which shows that there are multiple solutions possible. We do not compare the number of vehicles used and the run time of the algorithm since they were not significantly different for cases with and without restriction. The objective function value was 2% higher on an average when the number of simultaneous charging operations was restricted to two compared to the unlimited case. The difference was only 0.5% when three simultaneous charging operations were allowed.

Table 6: Sensitivity analysis on number of simultaneous charging operations

Instance	**Simultaneous Charging operations**			**% dev 2 vs 3** $\frac{(i-ii)*100}{i}$	**% dev 2 vs Un** $\frac{(i-iii)*100}{i}$	**% dev 3 vs Un** $\frac{(ii-iii)*100}{ii}$
	2 (i)	**3 (ii)**	**Unlimited (iii)**			
tc1c10s2cf3	25.28	25.28	25.28	0.00	0.00	0.00
tc1c10s2ct3	22.93	22.06	22.06	3.80	3.80	0.00
tc0c20s3cf2	39.37	39.15	39.15	0.57	0.57	0.00
tc0c20s4cf2	40.72	39.08	39.08	4.03	4.03	0.00
tc1c20s3cf1	24.62	24.62	24.62	0.00	0.00	0.00
tc1c20s3ct1	30.63	29.17	28.87	4.77	5.76	1.04
tc1c20s4ct1	26.02	25.61	25.61	1.59	1.59	0.00
tc1c20s4ct3	25.72	25.58	25.58	0.57	0.57	0.00
tc2c20s3cf0	38.00	38.00	37.60	0.00	1.06	1.06
tc2c20s3ct0	40.35	40.30	40.30	0.13	0.13	0.00
tc2c20s4cf0	37.93	37.48	37.48	1.18	1.18	0.00
tc2c20s4ct0	42.76	42.69	41.39	0.15	3.20	3.05
tc0c40s5cf0	40.09	39.36	39.36	1.81	1.81	0.00
tc1c40s5ct1	64.89	62.66	62.66	3.43	3.43	0.00
tc2c40s5cf3	29.33	29.12	29.12	0.69	0.69	0.00
tc2c80s12cf4	57.74	56.75	55.23	1.71	4.35	2.68
tc2c80s8cf4	69.59	67.60	67.60	2.86	2.86	0.00
tc2c80s8ct4	62.85	61.72	61.72	1.80	1.80	0.00
tc2c80s12cf4	Infeasible	53.37	52.30	-	-	2.00
				1.616	2.046	0.517

6. Conclusion

In this paper, we have introduced a new MILP model for Electric Vehicle Routing Problem with Non-Linear charging and Load-Dependent discharging (E-VRP-NL-LD) and an Adaptive Large Neighborhood Search (ALNS) algorithm for solving the E-VRP-NL-LD. We then extend the ALNS to solve E-VRP-NL-LD-CCS where we impose capacity constraints on simultaneous charging of EVs at a Charging Station (CS). The restricted form of the new MILP model allows multiple visits to the CS and outperforms the model of Montoya et al. (2017). The proposed ALNS algorithm has five new operators, and it can handle multiple charging technologies and load-dependent discharge. We have tested the algorithm on 120 E-VRP-NL instances introduced by Montoya et al. (2017), and the results demonstrate that in most cases the ALNS outperforms the Iterated Local Search and a Heuristic Concentration (ILS+HC) (Montoya et al., 2017) and the labeling method used in Froger et al. (2019). The proposed new operators improved solutions by 0.71% on an average over the cases without the new operators. Overall, out of 120 instances tested, ALNS had

improved on the Best Known Solutions (BKS) in 75 instances (62.5% of instances) and matched BKS in 38 instances (31.6% instances), out of which 26 were optimal solutions.

We have also tested the MILP model on E-VRP-NL-LD instances that considers load to be delivered at each customer and a load-dependent discharge function. Gurobi found 19 optimal and 27 feasible solutions out of the 60 instances tested. We tested ALNS on 120 instances with and without considering load. When we compared the results of with and without load formulations, the cases without load always underestimated the number of charging station visits and vehicles required, which in turn resulted in an underestimate of charging time by 15-80% and travel time by 2-30%. These underestimates were significantly higher in instances with 40 or more customers. We also test ALNS on 80 instances of E-VRP-NL-LD with a restriction on simultaneous charging operations (E-VRP-NL-LD-CCS) that resulted in an increase in the objective value by up to 5.76% compared to E-VRP-NL-LD. Interestingly the run time for ALNS for solving E-VRP-NL-LD-CCS was comparable to solving E-VRP-NL-LD.

In this paper, we have assumed that: *i*) there are no delivery time window restrictions, except a maximum tour duration limit, *ii*) vehicles are from a homogeneous fleet, and *iii*) vehicles operate out of a single depot. However, these may not hold in the real world, and these assumptions will affect routing decisions. Future work will aim to relax the above assumptions.

References

Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An Adaptive Large Neighborhood Search for a Vehicle Routing Problem with Multiple Routes. *Computers & Operations Research*, 41(1):167–173.

Bruglieri, M., Mancini, S., and Pisacane, O. (2019). The green vehicle routing problem with capacitated alternative fuel stations. *Computers and Operations Research*, 112.

Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows. *Electronic Notes in Discrete Mathematics*, 47:221–228.

Carley, S., Krause, R. M., Lane, B. W., and Graham, J. D. (2013). Intent to Purchase a Plug-in Electric Vehicle: A Survey of Early Impressions in Large US Cites. *Transportation Research Part D: Transport and Environment*, 18(1):39–45.

Conrad, R. and Figliozzi, M. (2011). The Recharging Vehicle Routing Problem. In Doolen, T. and Van Aken, E., editors, *Proceedings of the 2011 Industrial Engineering Research Conference*.

Czyzyk, J., Mesnier, M. P., and Moré, J. J. (1998). The NEOS Server. *IEEE Journal on Computational Science and Engineering*, 5:68–75.

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

Dolan, E. D. (2001). The NEOS Server 4.0 Administrative Guide. Technical memorandum, Mathematics and Computer Science Division, Argonne National Laboratory.

Felipe, Á., Ortuño, M. T., Righini, G., and Tirado, G. (2014). A Heuristic Approach for the Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128.

Froger, A., Mendoza, J. E., Jabali, O., and Laporte, G. (2019). Improved Formulations and Algorithmic Components for the Electric Vehicle Routing Problem with Nonlinear Charging Functions. *Computers & Operations Research*, 104:256–294.

Gropp, W. and Moré, J. J. (1997). Optimization Environments and the NEOS Server. In Buhman, M. D. and Iserles, A., editors, *Approximation Theory and Optimization*, pages 167 – 182. Cambridge University Press.

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An Adaptive Large Neighborhood Search Heuristic for Two-Echelon Vehicle Routing Problems Arising in City Logistics. *Computers & Operations Research*, 39(12):3215–3228.

Hidrue, M. K., Parsons, G. R., Kempton, W., and Gardner, M. P. (2011). Willingness to Pay for Electric Vehicles and Their Attributes. *Resource and Energy Economics*, 33(3):686–705.

Ichimori, T., Ishii, H., and Nishida, T. (1981). Routing a Vehicle with the Limitation of Fuel. *Journal of the Operations Research Society of Japan*, 24(3):277–281.

Ichimori, T., Ishii, H., and Nishida, T. (1983). Two routing problems with the limitation of fuel. *Discrete Applied Mathematics*, 6(1):85–89.

International Energy Agency (2018). CO2 Emissions from Fuel Combustion 2018 Highlights. Technical report.

Keskin, M. and Çatay, B. (2016). Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127.

Koyuncu, I. and Yavuz, M. (2019). Duplicating nodes or arcs in green vehicle routing: A computational comparison of two formulations. *Transportation Research Part E: Logistics and Transportation Review*, 122:605–623.

Li, Y., Chen, H., and Prins, C. (2015). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252:27–38.

Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2017). The Electric Vehicle Routing Problem with Nonlinear Charging Function. *Transportation Research Part B: Methodological*, 103:87–110.

Ropke, S. and Pisinger, D. (2006a). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472.

Ropke, S. and Pisinger, D. (2006b). A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, 171(3):750–775.

Schneider, M., Stenger, A., and Goeke, D. (2014). The Electric Vehicle Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, 48(4):500–520.

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Principles and Practice of Constraint Programming - Cp98*, volume 1520, pages 417–431.

Suzuki, Y. (2011). A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, 16(1):73–77.

UNCCC (2018). The Paris Agreement.

Zhang, S., Chen, M., Zhang, W., and Zhuang, X. (2020). Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert Systems with Applications*, 145:113123.

Appendix A. Gurobi Results for E-VRP-NL

Table A.7: Detailed computational results on the 10-customer instances

Instance	Bound	LP Gap (%)	Time (s)	Instance	Bound	LP Gap (%)	Time (s)
tc0c10s2cf1	19.75	0.00	15	tc1c10s3cf2	9.03	0.00	35
tc0c10s2ct1	12.30	0.00	15	tc1c10s3cf3	16.37	0.00	181
tc0c10s3cf1	19.75	0.00	21	tc1c10s3cf4	14.90	0.00	15
tc0c10s3ct1	10.80	0.00	11	tc1c10s3ct2	9.20	0.00	330
tc1c10s2cf2	9.03	0.00	25	tc1c10s3ct3	13.02	0.00	71
tc1c10s2cf3	16.37	0.00	40	tc1c10s3ct4	13.21	0.00	30
tc1c10s2cf4	16.10	0.00	10	tc2c10s2cf0	21.77	0.00	81
tc1c10s2ct2	10.75	0.00	100	tc2c10s2ct0	12.44	0.00	40
tc1c10s2ct3	13.17	0.00	290	tc2c10s3cf0	21.77	0.00	90
tc1c10s2ct4	13.83	0.00	26	tc2c10s3ct0	11.51	0.00	175

Table A.8: Detailed computational results on the 20-customer instances

Instance	Bound	LP Gap (%)	Time (s)	Instance	Bound	LP Gap (%)	Time (s)
tc0c20s3cf2	27.46	47.19	10800	tc1c20s4cf1	16.38	24.86	10800
tc0c20s3ct2	17.44	21.32	10800	tc1c20s4cf3	16.79	28.97	10800
tc0c20s4cf2	27.79	50.59	10800	tc1c20s4cf4	16.99	0.00	1940
tc0c20s4ct2	17.05	21.77	10800	tc1c20s4ct1	18.02	29.27	10800
tc1c20s3cf1	17.48	11.88	10800	tc1c20s4ct3	14.43	0.00	1217
tc1c20s3cf3	16.80	26.91	10800	tc1c20s4ct4	16.99	15.53	10800
tc1c20s3cf4	16.99	0.00	6167	tc2c20s3cf0	24.67	38.94	10800
tc1c20s3ct1	19.38	30.48	10800	tc2c20s3ct0	25.78	44.03	10800
tc1c20s3ct3	12.60	0.00	5006	tc2c20s4cf0	24.67	40.16	10800
tc1c20s3ct4	16.21	0.00	1529	tc2c20s4ct0	26.02	46.84	10800

Table A.9: Detailed computational results on the 40-customer instances

Instance	Bound	LP Gap (%)	Time (s)	Instance	Bound	LP Gap (%)	Time (s)
tc0c40s5cf0	34.73	39.96	10800	tc1c40s8cf1	-	-	10800
tc0c40s5cf4	34.80	42.94	10800	tc1c40s8ct1	-	-	10800
tc0c40s5ct0	30.50	35.19	10800	tc2c40s5cf2	-	-	10800
tc0c40s5ct4	34.30	42.16	10800	tc2c40s5cf3	20.15	31.01	10800
tc0c40s8cf0	32.77	36.82	10800	tc2c40s5ct2	-	-	10800
tc0c40s8cf4	-	-	10800	tc2c40s5ct3	-	-	10800
tc0c40s8ct0	26.55	26.36	10800	tc2c40s8cf2	27.70	35.27	10800
tc0c40s8ct4	39.54	50.62	10800	tc2c40s8cf3	20.41	38.10	10800
tc1c40s5cf1	-	-	10800	tc2c40s8ct2	26.38	32.72	10800
tc1c40s5ct1	-	-	10800	tc2c40s8ct3	25.20	52.65	10800

Appendix B. Gurobi Results for E-VRP-NL-LD

Instance	Bound		Gap	Instance	Bound		Gap
	Lower	Upper			Lower	Upper	
tc0c10s2cf1	19.09	19.09	0.00	tc1c20s3cf3	23.03	25.47	09.55
tc0c10s2ct1	16.90	16.90	0.00	tc1c20s3cf4	21.46	21.46	0.00
tc0c10s3cf1	19.09	19.09	0.00	tc1c20s3ct1	20.33	32.70	37.83
tc0c10s3ct1	16.17	16.17	0.00	tc1c20s3ct3	23.36	25.48	08.35
tc1c10s2cf2	16.89	16.89	0.00	tc1c20s3ct4	20.95	21.65	03.23
tc1c10s2cf3	25.28	25.28	0.00	tc1c20s4cf1	20.49	26.67	23.16
tc1c10s2cf4	23.49	23.49	0.00	tc1c20s4cf3	23.30	25.47	08.50
tc1c10s2ct2	16.75	16.75	0.00	tc1c20s4cf4	21.46	21.46	0.00
tc1c10s2ct3	22.06	22.06	0.00	tc1c20s4ct1	20.34	28.50	28.64
tc1c10s2ct4	20.89	20.89	0.00	tc1c20s4ct3	22.90	25.50	10.19
tc1c10s3cf2	16.89	16.89	0.00	tc1c20s4ct4	21.06	21.46	01.88
tc1c10s3cf3	25.28	25.28	0.00	tc2c20s3cf0	32.27	36.62	11.88
tc1c10s3cf4	22.49	22.49	0.00	tc2c20s3ct0	31.95	40.45	21.01
tc1c10s3ct2	16.75	16.75	0.00	tc2c20s4cf0	31.82	36.77	13.45
tc1c10s3ct3	20.79	20.79	0.00	tc2c20s4ct0	31.88	40.56	21.40
tc1c10s3ct4	20.30	20.30	0.00	tc0c40s5cf4	21.06	53.38	60.54
tc2c10s2cf0	25.02	27.66	9.54	tc0c40s8ct0	19.66	33.34	41.04
tc2c10s2ct0	18.14	18.14	0.00	tc2c40s5cf3	17.19	30.81	44.23
tc2c10s3cf0	21.29	27.66	23.01	tc2c40s5ct2	17.78	32.92	45.97
tc0c20s3cf2	27.37	42.42	35.47	tc2c40s5ct3	15.16	41.78	63.70
tc0c20s3ct2	25.85	32.13	19.56	tc2c40s8cf3	15.84	31.43	49.61
tc0c20s4ct2	26.47	31.49	15.94	tc2c40s8ct2	17.47	33.44	47.75
tc1c20s3cf1	23.12	24.65	06.21	tc2c40s8ct3	14.35	36.47	60.66
Average					21.55	27.11	15.70

Appendix C. ALNS Results for Montoya et al. (2017) instances

Instances	BKS (i)	ALNS				
		Best (ii)	Gap (%) $\frac{(ii-i)*100}{(ii)}$	Average (iii)	Gap (%) $\frac{(iii-i)*100}{(iii)}$
tc0c10s2cf1	**19.75**	**19.75**	**0.00**	**19.75**	**0.00**
tc0c10s2ct1	**12.30**	**12.30**	**0.00**	**12.30**	**0.00**
tc0c10s3cf1	**19.75**	**19.75**	**0.00**	**19.75**	**0.00**
tc0c10s3ct1	**10.80**	**10.80**	**0.00**	**10.80**	**0.00**
tc1c10s2cf2	**9.03**	**9.03**	**0.00**	**9.03**	**0.00**
tc1c10s2cf3	**16.37**	**16.37**	**0.00**	**16.37**	**0.00**
tc1c10s2cf4	**16.10**	**16.10**	**0.00**	**16.10**	**0.00**
tc1c10s2ct2	**10.75**	**10.75**	**0.00**	**10.75**	**0.00**

Instances	**BKS** (i)	**ALNS**			
		Best (ii)	**Gap (%)** $\frac{(ii-i)*100}{(ii)}$	**Average** (iii)	**Gap (%)** $\frac{(iii-i)*100}{(iii)}$
tc1c10s2ct3	**13.17**	**13.17**	**0.00**	**13.17**	**0.00**
tc1c10s2ct4	**13.83**	**13.83**	**0.00**	**13.83**	**0.00**
tc1c10s3cf2	**9.03**	**9.03**	**0.00**	**9.03**	**0.00**
tc1c10s3cf3	**16.37**	**16.37**	**0.00**	**16.37**	**0.00**
tc1c10s3cf4	**14.90**	**14.90**	**0.00**	**14.90**	**0.00**
tc1c10s3ct2	**9.20**	**9.20**	**0.00**	**9.20**	**0.00**
tc1c10s3ct3	**13.02**	**13.02**	**0.00**	**13.02**	**0.00**
tc1c10s3ct4	**13.21**	**13.21**	**0.00**	**13.21**	**0.00**
tc2c10s2cf0	**21.77**	**21.77**	**0.00**	**21.77**	**0.00**
tc2c10s2ct0	**12.45**	**12.45**	**0.00**	**12.45**	**0.00**
tc2c10s3cf0	**21.77**	**21.77**	**0.00**	**21.77**	**0.00**
tc2c10s3ct0	**11.51**	**11.51**	**0.00**	**11.51**	**0.00**
tc0c20s3cf2	27.6	27.47	-0.47	27.49	-0.40
tc0c20s3ct2	17.08	17.08	0.00	17.08	0.00
tc0c20s4cf2	27.48	27.47	-0.04	27.48	0.00
tc0c20s4ct2	16.99	16.99	0.00	16.99	0.00
tc1c20s3cf1	17.5	17.49	-0.06	17.49	-0.06
tc1c20s3cf3	16.63	16.44	-1.16	16.46	-1.01
tc1c20s3cf4	**17.00**	**17.00**	**0.00**	**17.00**	**0.00**
tc1c20s3ct1	18.95	18.94	-0.05	18.94	-0.05
tc1c20s3ct3	12.65	12.60	-0.40	12.60	-0.39
tc1c20s3ct4	**16.21**	**16.21**	**0.00**	**16.21**	**0.00**
tc1c20s4cf1	16.39	16.38	-0.06	16.56	1.00
tc1c20s4cf3	16.56	16.44	-0.73	16.48	-0.49
tc1c20s4cf4	**17.00**	**17.00**	**0.00**	**17.00**	**0.00**
tc1c20s4ct1	18.25	17.80	-2.53	18.17	-0.47
tc1c20s4ct3	**14.43**	**14.43**	**0.00**	**14.43**	**0.00**
tc1c20s4ct4	**17.00**	**17.00**	**0.00**	**17.00**	**0.00**
tc2c20s3cf0	24.68	24.68	0.00	24.92	0.95
tc2c20s3ct0	**25.79**	**25.79**	**0.00**	25.89	0.38
tc2c20s4cf0	24.67	24.67	0.00	24.67	0.00
tc2c20s4ct0	26.02	26.11	0.34	26.31	1.12
tc0c40s5cf0	32.67	32.20	-1.46	32.49	-0.54
tc0c40s5cf4	30.77	30.25	-1.72	30.27	-1.66
tc0c40s5ct0	28.72	28.38	-1.20	28.46	-0.91
tc0c40s5ct4	28.63	28.63	0.00	28.66	0.12
tc0c40s8cf0	31.04[2]	31.00	-0.13	31.04	0.00
tc0c40s8cf4	29.32	28.16	-4.12	28.36	-3.37
tc0c40s8ct0	26.35	26.38	0.11	26.57	0.83
tc0c40s8ct4	29.20	29.08	-0.41	29.11	-0.32
tc1c40s5cf1	65.16	64.85	-0.48	65.03	-0.20

Instances	**BKS** (i)	**ALNS**			
		Best (ii)	**Gap (%)** $\frac{(ii-i)*100}{(ii)}$	**Average** (iii)	**Gap (%)** $\frac{(iii-i)*100}{(iii)}$
tc1c40s5ct1	52.68	52.33	-0.67	52.35	-0.63
tc1c40s8cf1	40.75	40.75	0.00	40.93	0.44
tc1c40s8ct1	40.56	40.49	-0.17	41.39	2.00
tc2c40s5cf2	27.54[2]	27.54	0.00	27.54	0.00
tc2c40s5cf3	19.74	19.65	-0.46	19.65	-0.46
tc2c40s5ct2	26.91	27.15	0.88	27.15	0.88
tc2c40s5ct3	23.54	23.54	0.00	24.92	5.53
tc2c40s8cf2	27.14[2]	27.15	0.04	27.26	0.46
tc2c40s8cf3	19.66	19.65	-0.05	19.75	0.45
tc2c40s8ct2	26.33	26.28	-0.19	26.53	0.76
tc2c40s8ct3	22.71	22.45	-1.16	22.45	-1.16
tc0c80s12cf0	34.64	34.38	-0.76	34.96	0.92
tc0c80s12cf1	42.90	41.75	-2.75	42.22	-1.61
tc0c80s12ct0	39.31	37.66	-4.38	38.45	-2.24
tc0c80s12ct1	41.94	40.30	-4.07	40.51	-3.53
tc0c80s8cf0	39.43	39.08	-0.90	39.11	-0.82
tc0c80s8cf1	45.22[2]	44.50	-1.62	44.75	-1.05
tc0c80s8ct0	41.90	40.58	-3.25	41.31	-1.43
tc0c80s8ct1	45.27	44.87	-0.89	45.00	-0.60
tc1c80s12cf2	29.53[2]	28.58	-3.32	28.64	-3.11
tc1c80s12ct2	29.52	28.47	-3.69	28.80	-2.5
tc1c80s8cf2	30.81	28.98	-6.31	29.04	-6.10
tc1c80s8ct2	31.74	29.88	-6.22	30.36	-4.55
tc2c80s12cf3	31.97	30.61	-4.44	30.62	-4.41
tc2c80s12cf4	43.89	43.31	-1.34	43.61	-0.64
tc2c80s12ct3	30.83	30.08	-2.49	30.11	-2.39
tc2c80s12ct4	42.40	42.34	-0.14	42.52	0.28
tc2c80s8cf3	32.44	31.82	-1.95	31.97	-1.47
tc2c80s8cf4	49.17[2]	48.29	-1.82	49.17	0.00
tc2c80s8ct3	32.31[2]	31.76	-1.73	31.84	-1.48
tc2c80s8ct4	44.83	44.83	0.00	45.32	1.08
tc0c160s16cf2	61.20	59.11	-3.54	59.31	-3.18
tc0c160s16cf4	82.86[2]	79.48	-4.25	80.92	-2.40
tc0c160s16ct2	59.90	58.14	-3.03	58.49	-2.42
tc0c160s16ct4	82.32[2]	79.83	-3.12	80.91	-1.74
tc0c160s24cf2	59.27	58.27	-1.72	58.76	-0.86
tc0c160s24cf4	81.38[2]	80.31	-1.33	81.79	0.51
tc0c160s24ct2	59.25	57.03	-3.89	57.62	-2.82
tc0c160s24ct4	80.80[2]	79.08	-2.18	79.66	-1.43
tc1c160s16cf0	79.80	79.20	-0.76	80.33	0.66
tc1c160s16cf3	71.51[2]	70.33	-1.68	70.54	-1.38

Instances	**BKS** (i)	**ALNS**			
		Best (ii)	**Gap (%)** $\frac{(ii-i)*100}{(ii)}$	**Average** (iii)	**Gap (%)** $\frac{(iii-i)*100}{(iii)}$
tc1c160s16ct0	79.04	77.39	-2.13	78.43	-0.78
tc1c160s16ct3	73.29	68.23	-7.42	69.64	-5.25
tc1c160s24cf0	78.60	78.54	-0.08	79.88	1.60
tc1c160s24cf3	68.51[2]	66.29	-3.35	67.20	-1.94
tc1c160s24ct0	78.21	75.85	-3.11	76.91	-1.68
tc1c160s24ct3	68.72	65.82	-4.41	67.08	-2.44
tc2c160s16cf1	60.34	58.54	-3.07	58.65	-2.88
tc2c160s16ct1	60.27	58.37	-3.26	58.47	-3.08
tc2c160s24cf1	59.82	57.91	-3.30	59.04	-1.32
tc2c160s24ct1	59.13	55.56	-6.43	56.35	-4.93
tc1c320s24cf2	152.06[2]	149.64	-1.62	154.48	1.56
tc1c320s24cf3	117.46[2]	117.37	-0.08	117.50	0.04
tc1c320s24ct2	148.77	152.60	2.51	155.35	4.24
tc1c320s24ct3	116.64	115.34	-1.13	116.48	-0.14
tc1c320s38cf2	141.62[2]	141.62	0.00	147.74	4.14
tc1c320s38cf3	116.22	115.90	-0.28	116.24	0.02
tc1c320s38ct2	140.96	140.96	0.00	150.26	6.19
tc1c320s38ct3	116.06[2]	115.55	-0.44	116.34	0.24
tc2c320s24cf0	182.45[2]	189.14	3.54	192.41	5.18
tc2c320s24cf1	95.51	90.01	-6.11	90.42	-5.63
tc2c320s24cf4	122.74	122.06	-0.56	124.12	1.11
tc2c320s24ct0	181.45	187.30	3.12	189.51	4.25
tc2c320s24ct1	94.73	90.55	-4.62	91.22	-3.85
tc2c320s24ct4	121.82[2]	120.29	-1.27	121.43	-0.32
tc2c320s38cf0	176.92	176.92	0.00	184.35	4.03
tc2c320s38cf1	94.29	90.33	-4.39	90.43	-4.27
tc2c320s38cf4	122.32[2]	119.75	-2.15	119.77	-2.13
tc2c320s38ct0	190.96[2]	187.36	-1.92	188.42	-1.35
tc2c320s38ct1	94.53[2]	89.95	-5.09	90.68	-4.25
tc2c320s38ct4	121.66[2]	119.74	-1.60	120.82	-0.69
Average	**50.97**	**50.30**	**-1.24**	**50.92**	**-0.52**

[2] - BKS from Froger et al. (2019)

Appendix D. ALNS Results for cases with and without considering load

File name	With load		Without load	
	Total cost	No.of Vehicles	Total cost	No.of Vehicles
tc0c10s2cf1	19.09	5	21.41	4
tc0c10s2ct1	16.90	4	16.09	4
tc0c10s3cf1	19.09	5	21.41	4
tc0c10s3ct1	16.17	4	15.90	4
tc1c10s2cf2	16.89	4	16.56	4
tc1c10s2cf3	25.28	4	20.57	4
tc1c10s2cf4	23.49	5	20.79	5
tc1c10s2ct2	16.75	4	16.62	4
tc1c10s2ct3	22.06	4	20.62	4
tc1c10s2ct4	20.89	4	20.24	4
tc1c10s3cf2	16.89	4	16.56	4
tc1c10s3cf3	25.28	4	20.57	4
tc1c10s3cf4	22.49	4	20.30	5
tc1c10s3ct2	16.75	4	16.77	4
tc1c10s3ct3	20.79	4	19.67	4
tc1c10s3ct4	20.30	4	19.97	4
tc2c10s2cf0	27.66	5	26.79	5
tc2c10s2ct0	18.14	4	17.75	4
tc2c10s3cf0	27.66	5	26.79	5
tc2c10s3ct0	17.52	4	17.06	4
tc0c20s3cf2	39.15	7	36.45	7
tc0c20s3ct2	30.46	7	29.45	7
tc0c20s4cf2	39.08	8	36.45	7
tc0c20s4ct2	29.77	7	29.45	7
tc1c20s3cf1	24.62	7	23.99	7
tc1c20s3cf3	25.62	7	25.06	7
tc1c20s3cf4	21.46	7	21.64	7
tc1c20s3ct1	28.87	7	24.36	7
tc1c20s3ct3	25.24	7	24.66	7
tc1c20s3ct4	21.65	7	21.39	7
tc1c20s4cf1	24.60	7	23.80	7
tc1c20s4cf3	25.60	7	25.06	7
tc1c20s4cf4	21.46	7	21.64	7
tc1c20s4ct1	25.61	7	24.16	7
tc1c20s4ct3	25.58	7	25.16	7
tc1c20s4ct4	21.46	7	21.64	7
tc2c20s3cf0	37.60	7	35.85	7
tc2c20s3ct0	40.30	7	39.10	7
tc2c20s4cf0	37.48	7	35.51	7
tc2c20s4ct0	41.39	7	39.01	7

File name	With load		Without load	
	Total cost	**No.of Vehicles**	**Total cost**	**No.of Vehicles**
tc0c40s5cf0	39.36	9	32.2	7
tc0c40s5cf4	38.10	7	30.25	6
tc0c40s5ct0	35.17	8	28.38	7
tc0c40s5ct4	36.16	7	28.63	6
tc0c40s8cf0	37.55	8	31	6
tc0c40s8cf4	34.14	7	28.16	5
tc0c40s8ct0	31.96	7	26.38	6
tc0c40s8ct4	34.32	6	29.08	5
tc1c40s5cf1	80.74	12	64.85	10
tc1c40s5ct1	62.66	10	52.33	8
tc1c40s8cf1	47.37	8	40.75	7
tc1c40s8ct1	46.24	8	40.49	7
tc2c40s5cf2	35.54	7	27.54	6
tc2c40s5cf3	29.12	7	20.98	5
tc2c40s5ct2	32.18	7	27.15	6
tc2c40s5ct3	38.61	9	23.54	6
tc2c40s8cf2	34.98	7	27.15	6
tc2c40s8cf3	30.05	7	21.07	5
tc2c40s8ct2	32.19	7	26.28	6
tc2c40s8ct3	33.24	8	22.47	5
tc0c80s12cf0	40.57	9	34.48	8
tc0c80s12cf1	51.16	11	41.75	9
tc0c80s12ct0	43.41	10	38.47	9
tc0c80s12ct1	48.25	10	40.3	9
tc0c80s8cf0	46.23	11	39.08	9
tc0c80s8cf1	55.29	11	44.5	10
tc0c80s8ct0	48.00	11	40.58	9
tc0c80s8ct1	56.63	12	44.87	9
tc1c80s12cf2	35.24	9	28.58	7
tc1c80s12ct2	34.26	8	28.47	7
tc1c80s8cf2	35.72	9	28.98	7
tc1c80s8ct2	36.85	9	29.88	8
tc2c80s12cf3	41.98	10	30.61	8
tc2c80s12cf4	55.23	11	43.31	9
tc2c80s12ct3	37.63	9	30.08	8
tc2c80s12ct4	53.37	11	42.34	9
tc2c80s8cf3	41.54	11	31.82	8
tc2c80s8cf4	67.60	13	49.29	10
tc2c80s8ct3	40.32	10	31.76	8
tc2c80s8ct4	61.72	12	44.83	9
tc0c160s16cf2	103.72	22	59.11	15
tc0c160s16cf4	159.78	26	79.74	17

File name	With load		Without load	
	Total cost	**No.of Vehicles**	**Total cost**	**No.of Vehicles**
tc0c160s16ct2	72.67	18	58.14	15
tc0c160s16ct4	104.52	21	79.83	17
tc0c160s24cf2	100.69	21	58.6	15
tc0c160s24cf4	155.14	26	80.4	17
tc0c160s24ct2	72.89	17	57.03	15
tc0c160s24ct4	103.40	20	79.08	17
tc1c160s16cf0	142.64	25	79.2	17
tc1c160s16cf3	135.98	25	70.33	16
tc1c160s16ct0	95.02	19	77.39	17
tc1c160s16ct3	92.01	19	68.23	16
tc1c160s24cf0	141.29	25	78.54	17
tc1c160s24cf3	126.16	24	66.4	16
tc1c160s24ct0	93.30	19	75.85	16
tc1c160s24ct3	85.23	19	65.82	16
tc2c160s16cf1	101.20	21	58.54	15
tc2c160s16ct1	100.24	20	58.45	15
tc2c160s24cf1	74.26	17	57.91	15
tc2c160s24ct1	74.06	17	55.56	14
tc1c320s24cf2	313.09	50	149.64	33
tc1c320s24cf3	220.07	42	117.37	29
tc1c320s24ct2	229.08	41	152.6	33
tc1c320s24ct3	164.68	36	115.34	29
tc1c320s38cf2	297.15	49	141.63	32
tc1c320s38cf3	209.14	40	115.9	28
tc1c320s38ct2	221.65	40	140.96	32
tc1c320s38ct3	155.72	34	115.71	29
tc2c320s24cf0	329.89	52	189.14	36
tc2c320s24cf1	175.29	36	90.59	26
tc2c320s24cf4	169.86	36	124.03	30
tc2c320s24ct0	271.89	46	187.3	36
tc2c320s24ct1	170.24	37	90.55	26
tc2c320s24ct4	226.18	42	120.46	30
tc2c320s38cf0	266.17	44	176.92	36
tc2c320s38cf1	118.44	30	90.47	26
tc2c320s38cf4	219.52	42	119.75	29
tc2c320s38ct0	333.99	52	187.36	36
tc2c320s38ct1	119.55	30	89.95	26
tc2c320s38ct4	158.36	34	119.74	29