Transportation in Developing Economies manuscript No.
(will be inserted by the editor)

An Adaptive Large Neighborhood Search Approach for Electric Vehicle Routing with Load-dependent Energy Consumption

Surendra Reddy Kancharla · Gitakrishnan Ramadurai

the date of receipt and acceptance should be inserted later

Abstract Electric vehicles are gaining popularity day-by-day aided by growing pollution concerns with fossil fuel vehicles. Many logistics companies have already started testing electric vehicles for deliveries in cities. However, electric vehicles have issues such as range anxiety and long recharge times. These issues have to be considered in routing electric vehicles to avoid inefficient routes. One of the important factors that affects the amount of battery consumed is load carried by the vehicle. Considering loads will significantly affect the routes determined in the Electric Vehicle Routing Problem (EVRP). Most previous studies solved EVRP with distance minimization as the objective. We have considered load of vehicle in the power estimation function to calculate the energy requirement. An Adaptive Large Neighborhood Search (ALNS) with special operators particular to this problem structure is presented. ALNS was tested on 56 benchmark instances and it found better solutions for 14 instances and for 15 instances the solutions matched the best-known solution.

Keywords Electric vehicle routing, Energy minimization, Adaptive large neighborhood search

1 Introduction and Background

In recent times, many cities are facing severe air pollution which has been directly linked to the increase in vehicular traffic. Urban freight transport has a disproportionately higher share in the overall pollution from road transport in cities. To reduce the pollution caused by freight transport, many city logistics companies have started testing and using electric vehicles for last-mile deliveries. However, electric vehicles have issues such as range anxiety and long recharge times. These issues of electric vehicles pose challenges in solving routing of electric vehicles when compared to fossil fuel based vehicles. Unlike in traditional routing problems, the vehicles in Electric Vehicle Routing Problem (EVRP) need to visit recharge stations a number of times in a day. Recharge times at the stations and location of

Department of Civil Engineering, Indian Institute of Technology Madras, Chennai 600036, India. E-mail: surendrareddy.kancharla@gmail.com; gitakrishnan@iitm.ac.in

recharge station affect final routes in the solution, especially in cases with time window restrictions for delivery.

The structure of EVRP is similar to distance constrained vehicle routing problem [6] and variable-route vehicle-refueling problem [11], but the solution methods developed for the later problems cannot be directly applied to EVRP. One of the first attempts to use electric vehicles in routing was by [2]. They presented a problem having a fleet of electric vehicles with vehicle capacity and time window constraints and considered that recharging can be done at particular customers' location with a constant recharge time. [4] proposed a Green VRP that considers routing of alternative fuel vehicles with limited fuel capacity and possibility to refuel along the routes. Their model aims at minimizing distance traveled with the assumptions that fuel consumed is a factor of distance traveled and refuel time is constant. Their model does not include vehicle capacity and time window constraints. [4] introduced two heuristics: a modified Clarke and Wright savings heuristic and density-based clustering heuristic which were tested on small to large size instances. Later, [5] extended this problem to electric vehicles with partial recharges, recharging stations with a different rate of recharge, and cost. However, they did not consider time windows and vehicle capacities. They formulated a math programming model and developed a simulated annealing based algorithm to solve EVRP. [9] introduced a variable neighborhood search algorithm combined with Tabu search to solve the EVRP with Time Window (EVRPTW). In this, they generated a new set of instances by modifying the well-known Solomon instances to have feasible time window after introducing the recharge stations. The assumptions in their study are: a constant rate of recharge at the recharge stations and fuel consumed is proportional to the distance traveled. [3] extended the EVRPTW considering four different recharging strategies. They introduced a branch-price-and-cut based exact algorithm to solve the problem. All the above studies except [5] considered that vehicle gets fully recharged at the station. [7] allowed partial recharge of vehicles with a restriction that vehicles leave depot with full charge and return with empty charge. They have proposed a mixed integer linear programming model along with Adaptive Large Neighborhood Search (ALNS) to solve the problem.

All the above studies minimized the distance traveled by electric vehicles instead of minimizing the energy consumed and restrict the number of visits to a recharge station. It is particularly important to consider energy minimization in case of electric vehicles since it ensures that battery is used effectively. The main factors which influence the battery are load, speed, and grade. Earlier studies considered that the amount of battery discharged is either fixed or proportional to the distance traveled, ignoring these important factors. Our study overcomes this issue considering a comprehensive power estimation function that includes all three factors to estimate the power discharged. Also, we introduce an Adaptive Large Neighborhood Search (ALNS) algorithm with new operators that are effective in solving the EVRPTW.

2 Methodology

2.1 Estimation of power required

In the present study, power module from the Comprehensive Modal Emission Model (CMEM) [1] is used to estimate the required power. This model is selected particularly because it takes into account all the important parameters such as speed, acceleration, load, and grade. Engine Power (Pe) requirement is calculated using:

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

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 the coefficient of rolling resistance, ϵ is the vehicle drive train efficiency (typically 0.8), P_e is the second-by-second engine power output (kW). The present study considers that all vehicles travel at a fixed speed and on a level ground. Upon substituting these values, equation (1) reduces to equation (2):

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

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

$$P_e = \alpha + \beta M \tag{3}$$

Where,
$\alpha = \dfrac{0.5C_d\rho Av^3}{1000\epsilon}$ is a constant and
$\beta = \dfrac{gC_rv}{1000\epsilon}$ is the coefficient of weight, M.

2.2 Problem Description

In EVRPTW, the vehicles have to deliver goods to the customers within the specified time window. In case a vehicle arrives before the start time of time window, then the vehicle waits until the start time and delivers the goods. No vehicle is allowed to visit a customer after the due time. These vehicles can visit a recharge station en route and replenish their battery. The recharge time at the station is a linear function of charge depleted. Each customer can be visited only once by any vehicle. The objective is to minimize the total energy required.

2.3 Mathematical formulation

The present EVRPTW formulation is adopted from [9] and the objective function is changed to minimize energy. EVRPTW can be defined on a complete directed graph $\mathbf{G} = (V'_{0,n+1}, A)$, where $V'_{0,n+1} = \mathbf{V} \cup \mathbf{F'} \cup \{0, \text{n+1}\}$ is the set of vertices,

V = {1, . . . , n} denotes the set of customers, and **F'** a set of dummy recharging stations generated that allows several visits to each recharging station in the set **F**. Vertices 0 and n+1 denote the same depot, and all routes start at 0 and end at n+1. To indicate that a set contains 0 and/or n +1, the set is subscripted with the respective instance of depot. $\mathbf{A} = \{(i, j) \mid i \text{ and } j \in V'_{0,n+1}\ i \neq j\}$ is the set of arcs. Each arc (i, j) is associated with distance d_{ij} and travel time t_{ij}. Total energy consumed for traveling on each arc is $(\alpha x_{ij} + \beta u_j)t_{ij}$, where α and β are constant and coefficient of weight respectively in energy consumption estimation function. u_j is the load carried by vehicle till customer j. Finally, x_{ij} is the binary decision variable that is 1 if and only if a route exits between vertex i and j. τ_i, y_i, s_i, q_i, e_i and l_i are time of arrival, charge left, service time, demand, earliest arrival and latest arrival at vertex i respectively. Q is the total battery capacity and C is the load capacity of the vehicle.
The mixed-integer program formulation of EVRPTW is as follows:

$$min \sum_{i \in V'_0, j \in V'_{n+1}, i \neq j} (\alpha x_{ij} + \beta u_j)t_{ij} \tag{4}$$

Subject to:

$$\sum_{j \in V'_{n+1} i \neq j} x_{ij} = 1 \ \forall i \in V, \tag{5}$$

$$\sum_{j \in V'_{n+1} i \neq j} x_{ij} \leq 1 \ \forall i \in F', \tag{6}$$

$$\sum_{i \in V'_0 i \neq j} x_{ij} = \sum_{i \in V'_{n+1} i \neq j} x_{ji} \ \forall j \in V', \tag{7}$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - l_{n+1}(1 - x_{ij}) \leq \tau_j \ \forall i \in V_0, j \in V'_{n+1}, i \neq j, \tag{8}$$

$$\tau_i + t_{ij}x_{ij} + g(Q - y_i) - (l_{n+1} + gQ)(1 - x_{ij}) \leq \tau_j \ \forall i \in F', j \in V'_{n+1}, i \neq j, \tag{9}$$

$$e_j \leq \tau_j \leq l_j \ \forall j \in V'_{0,n+1}, \tag{10}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}) \ \forall i \in V'_0, j \in V'_{n+1}, i \neq j, \tag{11}$$

$$0 \leq u_0 \leq C, \tag{12}$$

$$0 \leq y_j \leq y_i - (\alpha x_{ij} + \beta u_j)d_{ij} + Q(1 - x_{ij}) \ \forall i \in V, j \in V'_{n+1}, i \neq j, \tag{13}$$

$$0 \leq y_j \leq Q - (\alpha x_{ij} + \beta u_j)d_{ij} \ \forall i \in F'_0, j \in V'_{n+1}, i \neq j, \tag{14}$$

$$x_{ij} \in \{0, 1\} \ \forall i \in V'_0, j \in V'_{n+1}, i \neq j \tag{15}$$

The objective (4) minimizes the total energy required. Constraints (5) enforce that each customer is visited exactly once and constraints (6) ensure that a copy of recharge station is visited at most once. Constraints (7) establish flow conservation at each vertex, the number of incoming vehicles is equal to the number of outgoing vehicles. Constraints (8) and (9) ensures time feasibility for vehicles leaving customers and the depot, and vehicles leaving recharging stations respectively. Recharge times are for a complete recharge from the charge level y_i to Q with a recharging rate g. Constraints (10) enforce that deliveries are done within the time window at each vertex. Further, the formation of sub-tours is also eliminated by constraints (8)-(10). Constraints (11) and (12) ensure demand satisfaction at all customers by assuring a non-negative load on arrival at all vertex including the

depot. Finally, battery charge is tracked and constraints (13) and (14) ensure the charge never runs out.

2.4 Solution Algorithm

Adaptive Large Neighborhood Search (ALNS) introduced by [8] is one of the most effective heuristics for solving large-scale vehicle routing problems. The present implementation of ALNS has few challenges such as battery constraint and multiple recharge station visits. Few new operators are introduced and few operators are adopted from the literature [7, 8] to overcome these challenges.

2.4.1 Initial solution

A greedy algorithm is used to find the initial solution. First, a node that is closest to the depot is selected and added to the solution. Next, a list of all the possible customers (who satisfy the time window and capacity constraints) that can be inserted are found and sorted in terms of closeness to the present customer. The first customer that satisfies all the constraints ((8) to (14)) is removed from the sorted list and is added to the solution. In case, if any delivery to a customer fails due to battery constraint, a recharge station would be inserted before the present customer. Even after inserting recharge station, if any customer in the solution fails to satisfy the constraints, a new route would be created. Algorithm 1 presents the pseudo code of initial solution algorithm.

Algorithm 1 Greedy Algorithm for Initial Solution

```
Start a new route with a customer closest to the depot
repeat
    Generate the list of all feasible customers from the list of unserved customers
    if no feasible customer exists then
        Start a new route with an unserved customer closest to the depot
    else
        if feasible after station insertion then
            Insert the station that is nearest to previous customer and the customer that
  increases power required by least
        else
            Insert the customer such that the increase in total power required is least
        end if
    end if
until all the customers are served
```

2.4.2 Overview of algorithm

The initial solution generated by the greedy algorithm is used as input for the ALNS. ALNS improves the solution iteratively by destroying and recreating the solution. It uses two types of operators: (i) removal operators and (ii) insertion operators for both customers and recharge stations. These operators destroy the solution by removing a certain number of customers/recharge stations from the solution based on the operator type. In next step, all these removed customers are

added to a customer pool, which is later used in insertion stage. Total iterations are divided into m segments and the scores of all the operators used are set to zero at the start of each segment. The average score obtained at the end of each segment is used to update the weights. The probability of selecting an operator using a roulette wheel selection is p_o=$W_o/(\sum_i W_i)$, where p_o is the probability of selecting operator o and W_i is the weight of operator i. Record-to-record acceptance is used as acceptance criteria for new solutions. This criterion accepts any solution that is no worse than α% of the current solution. Algorithm 2 presents the pseudo code of ALNS algorithm.

Algorithm 2 Adaptive Large Neighborhood Search

```
Read input data, initialize weights and scores
Generate an initial solution using bin packing approach (S)
Make initial solution as best solution S* ← S
j = 0                                   ▷ j - Number of iterations allowed without improvement
for i ← 0, MaxIterations do
    if j = N then                       ▷ N - Maximum iterations allowed without improvement
        Call station removal operator (S_ip)
        Call station insertion operator (S_i)
    else
        if j % K = 0 then                                   ▷ K - Predefined iteration interval
            Call route removal operator (S_ip)
            Call customer insertion operator (S_i)
        else
            Call customer removal operator (S_ip)
            Call customer insertion operator (S_i)
        end if
    end if
    if S_i ≤ (1+α)S then
        S ← S_i
        j ← 0 and update score
    else if j = N then
        S ← S_i
        j ← 0 and update score
    else
        j += 1
    end if
    if S < S* then
        S* ← S
    end if
    if i % Z = 0 then                                       ▷ Z - Weights update interval
        Update weights based on scores
    end if
end for
```

2.4.3 Customer removal operators

Random removal, related removal, worst distance removal, and least time window removal are the operators used. The first three are introduced by [8] and we have introduced a new removal operator namely, least time window removal. All these operators remove P (= min ([0.4Tc, 60]), where Tc is total customers) customers from their existing position and add them to the customer pool. The method of

removal of customers from a solution varies from operator to operator. In random removal, P customers are randomly selected and then removed; in related removal, first, a seed customer is randomly picked from the total customers. Then, P-1 nearest customers to the random seed are removed; in worst distance removal, removal gain (defined as the cost difference with and without the customer in the solution) is calculated for all the customers. Then, first P customers with highest removal gain are removed; in the least time window removal, first P customers with least time windows gap are removed.

2.4.4 Recharge station removal operators

Recharge station is a vital component in electric vehicle routes. Hence, removal and insertion of these stations may significantly affect the solution and can lead to better solutions. These operators (random station removal, worst station removal, and worst charge usage station removal) are used after every N consecutive non-improvement iterations. These will work similar to their counterparts in customer removal operators. First, Q (=min ([0.4Ts, 10]), where Ts is total recharge stations) recharge stations are removed from the solution and the solution is updated. Random station removal removes a total of Q recharge stations from all the routes. In worst station removal, power required to visit all the recharge stations from its preceding customers is calculated. Then, Q recharge stations that require the highest power are removed and the solution is updated. Finally, in worst charge usage station removal, all recharge station visited are arranged in decreasing order of charge remaining before visiting the recharge station. Then, first Q recharge stations in the ordered list are removed and the solution is updated.

2.4.5 Route removal operators

Route removal operators used in the study are random route removal [8] and greedy route removal [7]. These operators are performed once after every K iterations. A total of W ([0.1Tr, 0.4Tr], where Tr is total routes) routes are removed from the solution along with corresponding recharge stations. Customers in those routes are added to customer pool and the solution is updated. In random route removal, W routes are randomly removed from the solution, whereas in case of greedy route removal first W routes with the lowest number of customers visited are removed.

2.4.6 Customer insertion operators

Insertion operators reconstruct the solution with the customers in customer pool. Insertion operators used in this study are taken from literature [8]. These operators include greedy insertion, greedy insertion perturbation, and regret-k insertion. Greedy insertion randomly selects a customer from the customer pool and is inserted at a position that increases the power required by the least and satisfies the constraints for all the customers on that route. The main difference between greedy insertion and greedy insertion perturbation is the latter multiplies the cost of insertion by a factor d ([0.8, 1.2]) in order to further randomize the search. Regret-k insertion overcomes the myopic nature of greedy insertion by selecting the customer for insertion based on the regret value (cost difference between the

best insertion and the k^{th} best insertion). A new insertion operator based on time window is introduced called as regret-k time window insertion. It is similar to the regret-k insertion, but the former considers the difference in possible arrival time to due time for the best insertion to k^{th} best insertion. Last two operators i.e., regret-k and regret-k time window operators avoid cases of inserting customers at poor positions.

All these insertion operators will confirm that the new insertion will not violate any of the battery, time window, and capacity constraints before inserting the new customer into the solution. A new recharge station can be inserted if it helps in avoiding violation of battery constraint.

2.4.7 Station insertion operators

Removal of recharge stations from the solution will make the solution infeasible with respect to the battery constraint. Recharge station insertion operators help in making the solution feasible by inserting new recharge stations where ever required. The operators used here are best insertion operator [7] and a new operator called compare-k insertion that is proposed in the present study. Compare-k insertion compares the increase in cost by inserting a recharge station for k positions before the customer, where battery charge required to travel to the next customer is more than the available charge. Recharge station insertion is done before the customer that leads to least increase in cost upon satisfying all the constraints. The best insertion operator identifies and inserts the recharge station at the best possible position between previous recharge station and customer where battery level drops to negative. If no previous recharge station exists in solution, then the insertion is done between depot and customer. Both operators check for feasibility of solution after the station insertion. If no station can be inserted feasibly, the algorithm returns to the previous feasible solution.

3 Computational Results

Performance of the proposed ALNS was validated using the benchmark instances introduced in [9]. Benchmark instances consist of 56 large size instances in three sets, and these are modified versions of the well-known VRPTW instances of [10]. All of these instances have 100 customers and 21 recharging stations distributed over a 100 x 100 grid. The main difference among these three sets is how the customers are distributed in the 100 x 100 grid: (i) first set has customers in clusters (C), (ii) second set has customers randomly distributed (R), and (iii) third set has customers both clustered and randomly distributed (RC). Each of these sets has two subsets, type 1 and type 2, which differ by length of the time windows, vehicle load, and battery capacities. A comparison of results obtained is made with the benchmark reported in the literature [9].

The algorithm is coded in the Python programming language and all the instances were run on a system with a configuration of 3.6 GHz Intel Processor, 64 GB RAM running Windows 10. The algorithm was run for 25000 iterations and best of three solutions is reported here. The solutions that are an improvement over or same as Best Known Solution (BKS) are shown in bold.

Benchmark instances are available only for the distance minimization case. Hence, in order to compare the performance of the proposed ALNS, we modified the objective of ALNS to distance minimization and ran for those instances. In order to check the impact of the new operators introduced, we have tested ALNS with and without the new operators and the results are shown in table 1. Results show that the use of new operators has led to an average improvement in solution by 9.97% for type-1 and 3.12% in type-2 instances.

Table 1 Comparison of ALNS with and without new operators

Instances	**ALNS**	**ALNS with new operators**	**%dev**	**Instances**	**ALNS**	**ALNS with new operators**	**%dev**
C101	1189	1053	11.42	C201	787	645	18.06
C102	1096	1069	2.44	C202	744	645	13.33
C103	1060	1066	-0.57	C203	739	683	7.61
C104	931	908	2.42	C204	654	661	-1.07
C105	1173	1095	6.66	C205	759	641	15.56
C106	1180	1057	10.42	C206	761	672	11.70
C107	1107	1092	1.37	C207	728	741	-1.81
C108	1012	1041	-2.87	C208	714	714	0.00
C109	981	981	0.00				
R101	2108	1671	20.72	R201	1146	1145	0.09
R102	1967	1785	9.25	R202	1053	1029	2.24
R103	1717	1299	24.33	R203	929	917	1.30
R104	1406	1088	22.60	R204	752	756	-0.59
R105	1886	1461	22.55	R205	1003	1007	-0.41
R106	1650	1516	8.11	R206	978	950	2.87
R107	1517	1155	23.86	R207	847	827	2.42
R108	1282	1050	18.09	R208	750	754	-0.59
R109	1667	1539	7.69	R209	912	896	1.75
R110	1415	1342	5.16	R210	883	881	0.28
R111	1438	1322	8.08	R211	778	775	0.40
R112	1330	1050	21.05				
RC101	2180	1731	20.58	RC201	1349	1326	1.68
RC102	1928	1801	6.58	RC202	1226	1189	3.03
RC103	1561	1553	0.51	RC203	929	989	2.67
RC104	1385	1239	10.56	RC204	1117	911	1.91
RC105	1874	1732	7.55	RC205	1016	1146	-0.30
RC106	1810	1757	2.92	RC206	1143	1108	0.76
RC107	1500	1448	3.49	RC207	960	941	1.97
RC108	1408	1210	14.03	RC208	845	850	-0.64
Average			**9.97**				**3.12**

Table 2 shows the results of ALNS in comparison with BKS. Instances in column one are type-1 instances that have shorter time windows, electric vehicles with lower load limit and shorter range for each charge. Instances in column five are type-2 instances that have larger time windows at each customer and electric vehicles with higher load limit having longer range for each charge. Column four and eight show the deviation in ALNS result compared to BKS. From Table 2, it is clear that the ALNS has a comparable performance with the previous algorithms.

Out of 56 instances tested, ALNS found 14 new solutions and 15 same as best known solutions.

Table 2 ALNS vs Best Known Solutions (BKS)

Instances	**BKS**	**ALNS**	**%dev**	**Instances**	**BKS**	**ALNS**	**%dev**
C101	1053	**1053**	0.00	C201	645	**645**	0.00
C102	1056	1069	-1.23	C202	645	**645**	0.00
C103	1041	1066	-2.40	C203	644	683	-6.06
C104	979	**908**	7.25	C204	636	661	-3.93
C105	1075	1095	-1.86	C205	641	**641**	0.00
C106	1057	**1057**	0.00	C206	638	672	-5.33
C107	1031	1092	-5.92	C207	638	741	-16.14
C108	1100	**1041**	5.36	C208	638	714	-11.91
C109	1036	**981**	5.31				
R101	1671	**1671**	0.00	R201	1265	**1145**	9.49
R102	1495	1785	-19.40	R202	1052	**1029**	2.19
R103	1299	**1299**	0.00	R203	896	917	-2.34
R104	1088	**1088**	0.00	R204	791	**756**	4.42
R105	1461	**1461**	0.00	R205	989	1007	-1.82
R106	1345	1516	-12.71	R206	925	950	-2.70
R107	1155	**1155**	0.00	R207	849	**827**	2.59
R108	1050	**1050**	0.00	R208	737	754	-2.31
R109	1294	1539	-18.93	R209	872	912	-4.59
R110	1127	1342	-19.08	R210	847	881	-4.01
R111	1106	1322	-19.53	R211	847	**775**	8.50
R112	1050	**1050**	0.00				
RC101	1731	**1731**	0.00	RC201	1445	**1326**	8.24
RC102	1555	1801	-15.82	RC202	1413	**1189**	15.85
RC103	1351	1553	-14.95	RC203	1074	**989**	7.91
RC104	1239	**1239**	0.00	RC204	885	911	-2.94
RC105	1475	1732	-17.42	RC205	1322	**1146**	13.31
RC106	1438	1757	-22.18	RC206	1191	**1108**	6.97
RC107	1276	1448	-13.48	RC207	996	**941**	5.52
RC108	1210	**1210**	0.00	RC208	838	850	-1.43
Average			**-5.76**				**0.72**

A comparison between the energy minimizing and distance minimizing objectives in terms of energy required and vehicles used are shown in Table 3. Results clearly show that minimizing energy instead of distance leads to less energy consumption but at the cost of increase in the number of vehicles used. There is an average saving of 2.06% in energy with an average increase in the number of vehicles by 10.68% in the tested instances. However, there are 30 cases with either reduced or same number of vehicles along with the reduction in energy consumption. The variation in the number of vehicles and the energy consumed depends greatly on how the customers are distributed and the time window in which they accept the deliveries. The problem considered has no limitations on the number of vehicles that can be used and hence the algorithm used more vehicles given that it decreases the total energy required. Many vehicles with smaller loads can result in lower energy consumption than few vehicles with heavy loads. The former is used by energy minimization case, whereas the latter is used by distance minimization since it leads to lesser empty return trips and hence total distance.

Table 3 Energy and vehicles required in distance and energy minimization

Instances	Distance min.		Energy min.		%dev energy	%dev vehicles
	Energy	Vehicles	Energy	Vehicles		
C101	97.21	12	89.10	13	8.34	-8.33
C102	89.40	12	87.45	12	2.18	0.00
C103	88.01	12	87.65	12	0.41	0.00
C104	76.20	10	75.43	10	1.01	0.00
C105	94.89	12	89.57	12	5.60	0.00
C106	94.62	11	88.68	12	6.27	-9.09
C107	89.45	12	89.17	12	0.31	0.00
C108	95.24	11	89.82	12	5.69	-9.09
C109	81.38	12	80.85	11	0.65	-9.09
C201	68.58	4	68.21	7	0.54	-75.00
C202	66.43	4	66.25	6	0.27	-50.00
C203	64.30	5	62.95	6	2.09	-20.00
C204	65.03	4	62.98	5	3.15	-25.00
C205	67.32	4	65.55	6	2.63	-50.00
C206	64.90	5	63.27	6	2.51	-20.00
C207	66.51	6	66.15	7	0.55	-16.67
C208	67.52	6	65.90	5	2.39	16.67
R101	165.51	18	164.65	25	0.52	-39.89
R102	141.26	22	140.59	22	0.48	0.00
R103	127.87	13	127.18	20	0.54	-53.85
R104	108.78	11	106.97	14	1.66	-27.27
R105	135.09	14	133.53	21	1.16	-50.00
R106	129.75	18	120.26	19	7.31	-5.56
R107	114.85	12	109.48	16	4.68	-33.33
R108	107.31	11	104.33	14	2.78	-27.27
R109	123.35	17	122.14	18	0.98	-5.88
R110	109.87	15	108.01	15	1.69	0.00
R111	110.23	15	106.12	14	3.72	6.67
R112	101.65	11	101.62	14	0.03	-27.27
R201	137.31	9	134.19	10	2.27	-11.11
R202	126.99	8	126.32	8	0.52	0.00
R203	114.18	7	112.16	7	1.77	0.00
R204	102.77	5	102.70	5	0.07	0.00
R205	126.60	8	125.90	7	0.55	12.50
R206	119.89	7	118.12	7	1.48	0.00
R207	112.12	6	109.13	6	2.66	0.00
R208	105.48	5	103.40	5	1.97	0.00
R209	114.82	7	114.65	7	0.14	0.00
R210	112.87	6	110.61	7	2.00	-16.67
R211	108.24	5	105.39	5	2.64	0.00
RC101	161.36	16	151.52	20	6.10	-25.00
RC102	147.72	17	143.88	18	2.60	-5.88
RC103	125.80	15	125.01	16	0.62	-6.67
RC104	111.70	11	110.50	14	1.07	-27.27
RC105	140.83	18	139.04	17	1.27	5.56
RC106	140.85	18	140.07	18	0.55	0.00
RC107	125.27	15	117.06	15	6.55	0.00
RC108	114.88	11	110.66	14	3.68	-27.27
RC201	166.14	10	162.59	9	2.14	10.00
RC202	155.06	9	154.37	9	0.44	0.00
RC203	133.69	7	133.34	7	0.26	0.00
RC204	123.41	7	121.60	7	1.46	0.00
RC205	146.82	8	146.02	8	0.55	0.00
RC206	150.17	7	147.97	7	1.47	0.00
RC207	128.51	7	128.01	6	0.40	14.29
RC208	118.85	6	118.62	6	0.19	0.00
Average					**2.06**	**-10.68**

4 Conclusion

The present study introduced an Adaptive Large Neighborhood Search (ALNS) algorithm with three new operators (least time window removal, regret-k time window removal, and compare-k insertion) to solve the Electric Vehicle Routing Problem with Time Windows (EVRPTW). The new operators had led to an improvement in solution by 9.97% and 3.12% for type-1 and type-2 instances respectively. ALNS was tested on 56 benchmark instances and it found a better solution for 14 instances and for 15 instances the solutions matched the best-known solutions. Overall, the proposed algorithm showed comparable performance with the existing algorithms in distance minimization case. In energy minimization objective, the algorithm resulted in an average energy savings of 2.06% albeit at the cost of increase in the number of vehicles being used (10.68%). Since the formulation does not limit the number of vehicles that can be used, the present algorithm realized solutions that utilize more vehicles. The increase in the number of vehicles compared to distance minimization can also be attributed to the use of load in power requirement calculations.

One of the main limitations of ALNS is it does not minimize the number of vehicles used. Few possible ways of extending this work are to combine ALNS with a low-performance heuristic to help in minimizing the vehicles required. Another extension is the inclusion of acceleration/deceleration patterns that may lead to better estimation of power required that can alter the final routes.

References

1. 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
2. Conrad RG, Figliozzi MA (2011) The recharging vehicle routing problem. In: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), p 1
3. Desaulniers G, Errico F, Irnich S, Schneider M (2016) Exact algorithms for electric vehicle-routing problems with time windows. Operations Research 64(6):1388–1405
4. Erdoğan S, Miller-Hooks E (2012) A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review 48(1):100–114
5. Felipe Á, Ortuño MT, Righini G, 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
6. Juan AA, Goentzel J, Bektaş T (2014) Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations? Applied Soft Computing 21:84–94
7. Keskin M, Çatay B (2016) Partial recharge strategies for the electric vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies 65:111–127

8. Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40(4):455–472
9. Schneider M, Stenger A, Goeke D (2014) The electric vehicle-routing problem with time windows and recharging stations. Transportation Science 48(4):500–520
10. Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35(2):254–265
11. Suzuki Y (2012) A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands. Decision Support Systems 54(1):758–767