Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence
Canadian AI 2026
25–29 May 2026, Vancouver, BC

A Hybrid ALNS-Based Approach for the Electric Vehicle Routing Problem with Time Windows

Chiamaka Anicho-Okoro[†], Mariem Belhor[‡], Omar Alam[†,*]
[†] Trent University, Peterborough, Ontario, Canada
[‡] University of Picardie Jules Verne, LTI UR 3899, Amiens, France.

Abstract

Facing the need for low-carbon vehicle routing solutions, this paper addresses the Electric Vehicle Routing Problem with Time Windows (EVRPTW), which involves battery limitations and recharging constraints. In particular, we propose a hybrid metaheuristic approach that combines Adaptive Large Neighborhood Search (ALNS), a Greedy Time-Oriented Nearest Neighborhood Heuristic (GTONNH), and Tabu Search (TS). Experiments on standard benchmark instances show that the proposed GTONNH-ALNS-TS variant outperforms baseline approaches, achieving the best solutions on more than 60% of large-scale instances and reaching the minimum fleet size in nearly 95% of the cases. On average, the proposed approach reduces the required number of vehicles by more than one compared to classical ALNS, while maintaining competitive travel distances. High-quality solutions are obtained within a few seconds on small and medium-sized instances, highlighting the efficiency of the proposed framework.

Keywords: Electric Vehicle Routing Problem, Metaheuristics, Hybrid Approaches, Adaptive Large Neighborhood Search, NP-hard Combinatorial Optimization

1. Introduction

Sustainable transport has become a priority in recent years, leading to a surge of interest in electric vehicles (EVs) for logistics. Governments are pushing for cleaner transportation. For example, according to the International Council on Clean Transportation (ICCT)[1], the EU has targeted a 40% reduction in greenhouse gas emissions by 2030, in which the transport sector plays a major role. In addition, the Canadian government[2] has committed to achieving 100% zero-emission vehicle sales for all new light-duty vehicles by 2035, with interim targets of at least 20% by 2026 and 60% by 2030, as part of Canada's 2030 Emissions Reduction Plan. In response, logistics companies are beginning to deploy EVs in their delivery fleets. EVs offer clear environmental benefits, as they produce no tailpipe emissions [1], and generate minimal noise [2]. Moreover, when powered by renewable energy sources, they can significantly reduce the overall carbon footprint of freight transportation [3]. However, replacing conventional vehicles with EVs also introduces new operational challenges. Unlike vehicles equipped with internal combustion engines, EVs have a limited driving range and require periodic stops for battery recharging [4].

The above challenges increase operational complexity and make route planning and scheduling substantially more difficult for logistics operators. Consequently, classical vehicle routing models are no longer sufficient to accurately capture the specific characteristics of EV operations. This has led to the development of dedicated optimization models for EV routing. Among these models, the Electric Vehicle Routing Problem (EVRP) has emerged as an extension of the classical Vehicle Routing Problem (VRP), explicitly accounting for the operational characteristics of EVs, such as limited battery capacity and the need for recharging [5]. Building on this framework, several EVRP variants have been proposed to better reflect

[1] International Council on Clean Transportation (ICCT), 2020–2030 CO_2 standards for new cars and light-commercial vehicles in the European Union : https://theicct.org/publication/2020-2030-co2-standards-for-new-cars-and-light-commercial-vehicles-in-the-european-union-2/
[2] Government of Canada, *Canada's Zero-Emission Vehicle Sales Targets*: https://tc.canada.ca/en/road-transportation/innovative-technologies/zero-emission-vehicles/canada-s-zero-emission-vehicle-sales-targets

* omaralam@trentu.ca

real-world constraints. In particular, the Electric Vehicle Routing Problem with Time Windows (EVRPTW), first introduced by [6], incorporates customer time window constraints in addition to battery and recharging considerations. Due to the inherent NP-hardness of the VRP and the added complexity of battery management and charging decisions, solving the EVRPTW optimally remains highly challenging. In this paper, we address the EVRPTW using a hierarchical objective structure that prioritizes minimizing the number of vehicles, followed by minimizing the total traveled distance, which directly reduces operational costs, energy consumption, and carbon emissions. To solve the problem, we propose a hybrid solution approach combining a Greedy Time-Oriented Nearest Neighborhood Heuristic (GTONNH) for initialization, Adaptive Large Neighborhood Search (ALNS) for solution exploration, and Tabu Search (TS) for search intensification. The proposed approach is evaluated on the standard EVRPTW benchmark instances [6]. Computational results show that the hybrid GTONNH-ALNS-TS approach consistently achieves solutions with fewer vehicles than baseline ALNS variants, while maintaining competitive total distances and reasonable computational effort across small, medium, and large instances. The remainder of this paper is organized as follows. Section 2 reviews the literature on the EVRP, with a focus on existing solution approaches. Section 3 introduces the proposed optimization model. Section 4 details the hybrid solution approach developed in this work. Section 5 presents the experimental setup while Section 6 discusses the obtained results. Finally, Section 7 concludes the paper and outlines future research perspectives.

2. Related Work

In this section, we review the existing literature on the EVRP, with a particular focus on its main variants and solution approaches.

2.1. From the Classical VRP to Green Routing Problems

The classical VRP was first introduced by Dantzig and Ramser [7] in 1959. It consists of determining the optimal routes for a fleet of vehicles that must serve a set of customers from a central depot while minimizing total routing cost. The VRP is known to be NP-hard and has given rise to numerous variants, including the Capacitated VRP and the VRP with Time Windows (VRPTW). Increasing environmental concerns and rising fuel costs have motivated the development of environmentally oriented extensions of the VRP, often referred to as "green" routing problems [8]. Erdoğan and Miller-Hooks [9] introduced the Green Vehicle Routing Problem (GVRP), which models alternative-fuel vehicles with limited driving range and refueling constraints. Similarly, Conrad and Figliozzi [10] proposed the Recharging Vehicle Routing Problem (RVRP), explicitly considering battery EVs and the need to visit recharging stations during route execution. These contributions paved the way for the EVRP, which explicitly incorporates battery state-of-charge, energy consumption, and charging constraints into the routing process.

2.2. Key variants of the EVRP

Beyond the baseline EVRP, a wide range of problem variants has been proposed in the literature to better capture the diversity and complexity of real-world logistics operations involving EVs. These variants extend the EVRP by incorporating additional operational or strategic constraints, such as time windows [6], heterogeneous or mixed fleets [11, 12], alternative charging strategies (e.g., partial or nonlinear recharging [13, 14]), or more complex distribution structures, including multi-echelon networks [15]. In this work, we focus on the EVRPTW. Although the EVRPTW has been widely studied in the literature, most notably in the work of [6], the variant considered in this paper differs from the original formulation in several aspects, which are described in detail in Section 3.

2.3. EVRP Solution Approaches

Due to the NP-hard nature of the EVRP and its variants, a wide range of solution approaches has been proposed in the literature. Broadly speaking, these approaches can be grouped into two main categories: exact optimization methods and heuristic-based approaches. (i) **Exact methods.** They aim to guarantee optimal solutions, but their applicability is limited to small-scale instances due to the high computational complexity induced by routing, battery, and charging decisions. Typical exact approaches include branch-and-cut and branch-price-and-cut algorithms [16]. As a result, these methods quickly become impractical for realistically sized problems.

(ii) **Heuristic and metaheuristic methods.** For large-scale instances, heuristic and metaheuristic approaches are therefore widely adopted. These methods, such as Variable Neighborhood Search (VNS) [17], Ant Colony Optimization (ACO) [18] and Genetic Algorithms (GA) [19], are designed to efficiently explore the solution space and produce high-quality solutions within reasonable computational times, although they do not guarantee optimality [20]. To mitigate this limitation, many studies propose hybrid approaches that combine metaheuristics with additional components, such as problem-specific heuristics, intensification strategies, or exact optimization subroutines. These hybrid methods aim to balance solution quality and computational efficiency and are particularly well suited for solving large and complex EVRP instances. Table 1 provides an overview of recent hybrid solution approaches for EVRP variants, including the methods employed and the problem variant considered.

Table 1. Example of Hybrid EVRP Solution Methods

Reference	**Hybrid Method**	**EVRP Variant**
[21]	Hybrid VNS	Capacitated EVRP with recharging stations
[22]	Two-Stage Hybrid GA	Capacitated EVRP
[14]	Hybrid Memetic algorithm	EVRPTW with simultaneous pickup-delivery
[23]	GA + 2-opt	EVRPTW
[12]	ACO + VNS	Mixed-fleet EVRPTW
[24]	ALNS+ACO	Time-Dependent Open EVRP
[25]	Simulated annealing+VNS	Close-open EVRP
[26]	GA+VNS	Clustered EVRP
[27]	GSGA	EVRP
This study	GTONNH-ALNS-TS	EVRPTW with single-visit charging stations

Overall, the literature shows that solving EVRP variants is computationally challenging once energy, charging and time-related constraints are taken into account. Most recent works rely on heuristic and metaheuristic approaches, often combined within hybrid frameworks to improve both solution quality and computational efficiency. In this context, our present work focuses on EVRPTW without fictitious charging station duplication and introduces a new hybrid approach adapted to this setting.

3. Problem description and optimization model

In this work, we consider an EVRPTW variant that differs from the formulation of Schneider et al. [6]. As in the original problem, a fleet of homogeneous electric vehicles is routed from a central depot to serve a set of customers while satisfying time windows constraints, vehicle capacity limits, and battery capacity restrictions, with energy consumption assumed to be proportional to the traveled distance and recharging allowed at dedicated stations. However, unlike the formulation of [6], which introduces fictitious copies of recharging stations to allow multiple visits within a single route, we explicitly consider only real charging stations. Charging stations may be visited multiple times by the same or different vehicles, and each visit corresponds to a full battery recharge. This choice reflects real-world charging infrastructure and simplifies the charging model while remaining representative of common operational settings in EV logistics.

Let I denote the set of customers and F the set of recharging stations. Let vertices 0 and $n+1$ represent the start and end depot, respectively. The complete vertex set is defined as

$$V = I \cup F \cup \{0, n+1\}.$$

The problem is defined on a complete directed graph $G = (V, A)$, where

$$A = \{(i, j) \mid i, j \in V, i \neq j\}.$$

Each arc $(i, j) \in A$ is associated with a distance d_{ij} and a travel time t_{ij}. Traveling along arc (i, j) consumes $r \cdot d_{ij}$ units of battery charge, where r denotes the constant energy consumption rate. All vehicles are identical and have a maximum cargo capacity C and a maximum battery capacity Q. Each customer $i \in I$ has a demand $q_i > 0$ and a service time s_i, while for all $i \in V \setminus I$, $q_i = s_i = 0$. Each vertex $i \in V$ is associated with a time window $[e_i, l_i]$. Service cannot start before e_i and must start no later than l_i. At a recharging station $i \in F$, the battery can be recharged up to its full capacity Q with a constant charging rate g. Therefore, the recharging time depends on the battery level upon arrival at the station. The following decision variables are used:

- $x_{ij} \in \{0, 1\}$ equals 1 if arc (i, j) is traveled, and 0 otherwise;
- $\tau_i \geq 0$ denotes the arrival time at vertex i;
- $u_i \geq 0$ denotes the remaining vehicle load upon arrival at vertex i;
- $y_i \geq 0$ denotes the remaining battery charge upon arrival at vertex i.

We formulate EVRPTW as the following mixed-integer program.

$$\min \quad \sum_{i \in V} \sum_{j \in V, j \neq i} d_{ij} x_{ij} \tag{1}$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{i \in V, i \neq j} x_{ij} - \sum_{k \in V, k \neq j} x_{jk} = 0 \quad \forall j \in V \setminus \{0, n+1\} \tag{3}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} \leq \tau_j + M(1 - x_{ij}) \quad \forall i \in I \cup \{0\},\ \forall j \in V,\ i \neq j \tag{4}$$

$$\tau_i + t_{ij} x_{ij} + g(Q - y_i) \leq \tau_j + M(1 - x_{ij}) \quad \forall i \in F,\ \forall j \in V,\ i \neq j \tag{5}$$

$$e_i \leq \tau_i \leq l_i \quad \forall i \in V \tag{6}$$

$$u_j \leq u_i - q_j x_{ij} + C(1 - x_{ij}) \quad \forall i, j \in V,\ i \neq j \tag{7}$$

$$0 \leq u_i \leq C \quad \forall i \in V \tag{8}$$

$$y_j \leq y_i - r d_{ij} x_{ij} + Q(1 - x_{ij}) \quad \forall i, j \in V,\ i \neq j \tag{9}$$

$$0 \leq y_i \leq Q \quad \forall i \in V \tag{10}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V,\ i \neq j \tag{11}$$

The objective function (1) minimizes the total traveled distance. In practice, solutions are compared according to a hierarchical criterion in which the number of vehicles takes priority, as described in Section 4. Constraints (2) ensure that each customer is visited exactly once. Constraints (3) enforce flow conservation by guaranteeing that, at each vertex, the number of incoming arcs equals the number of outgoing arcs. Constraints (4) ensure time feasibility for arcs leaving customers and the depot, while Constraints (5) account for the additional recharging time incurred at recharging stations. Constraints (6) enforce the time window restrictions at all vertices. Constraints (7) and (8) guarantee demand fulfillment and ensure that the vehicle load never exceeds its capacity nor becomes negative. Constraints (9) and (10) ensure that the battery charge level remains within feasible bounds and never drops below zero. Finally, Constraints (11) define the binary nature of the routing decision variables.

4. New Hybrid Metaheuristic for EVRPTW

The EVRPTW is a NP-hard combinatorial optimization problem. Exact solution methods rapidly become impractical due to the combinatorial explosion induced by routing, scheduling and energy constraints. Consequently, our solution approach is primarily based on metaheuristic optimization, which allows us to efficiently explore large solution spaces while maintaining reasonable computational times. The main objective of the solution strategy is to guide the search efficiently toward promising regions of the solution space in order to achieve fast convergence while maintaining solution quality. To this end, we propose a hybrid solution approach primarily based on ALNS metaheuristic. The proposed method is designed to efficiently explore the solution space while ensuring fast convergence toward high-quality solutions. To further enhance performance, we combine ALNS with the greedy heuristic GTONNH initialization procedure and a TS refinement phase.

4.1. Initial Solution Construction

The performance of ALNS is highly dependent on the quality of the initial solution. Instead of using the construction heuristic proposed by [6], we employ a GTONNH strategy to generate an initial feasible solution. The GTONNH incrementally constructs vehicle routes by inserting customers according to greedy criteria based on distance, time window feasibility and battery consumption, as described in Algorithm 1. As a result, the GTONNH strategy can efficiently generate feasible solutions and provides a suitable starting point for the ALNS procedure.

Algorithm 1 GTONNH Initialization

Require: Customers C, stations S, depot D, vehicle capacity C, battery capacity Q, weights $\delta_1, \delta_2, \delta_3$
Ensure: Feasible initial solution Sol

$Sol \leftarrow \emptyset$, $Unserved \leftarrow C$
while $Unserved \neq \emptyset$ **do**
  Initialize route $R \leftarrow [D]$, battery $b \leftarrow Q$
  while $Unserved \neq \emptyset$ **do**
    $Candidates \leftarrow \{c \in Unserved : \text{capacity\_feasible}(R, c)\}$
    if $Candidates = \emptyset$ **then**
      break
    end if
    $c^* \leftarrow \arg\min_{c \in Candidates} \text{CompositeCost}(R, c, \delta_1, \delta_2, \delta_3)$
    if insertion of c^* infeasible w.r.t. time or battery **then**
      break
    end if
    Insert c^* into R (with charging station if required)
    Update route time and battery
    $Unserved \leftarrow Unserved \setminus \{c^*\}$
  end while
  $Sol \leftarrow Sol \cup \{R\}$
end while
return Sol

4.2. Adaptive Large Neighborhood Search

The core of the proposed solution approach is an ALNS metaheuristic [28]. At each iteration, the current solution is partially destroyed and then repaired using a pair of destroy and repair operators. The destroy operators remove a subset of customers from the solution, while the repair operators reinsert them in feasible positions. The selection of destroy and repair operators is performed adaptively based on their historical performance, as described in Algorithm 2. Operator weights are dynamically updated to favor those that contribute to improved solutions. The hierarchical objective is enforced in the acceptance criterion and comparison functions. Algorithms 3 and 4 implement this comparison and acceptance respectively. All neighborhood modifications are designed to preserve feasibility with respect to all constraints (time windows, vehicle capacity, battery feasibility constraints).

Algorithm 2 Adaptive Large Neighborhood Search (ALNS)

Require: Initial solution S_0, maximum iterations $I_{\max}$, segment size
Ensure: Best solution S^*

$S \leftarrow S_0$, $S^* \leftarrow S_0$
Initialize temperature T, operator weights, and scores
for $i = 1$ to $I_{\max}$ **do**
  Select destroy and repair operators according to adaptive weights
  Generate candidate solution $\tilde{S}$ by applying selected operators
  Restore battery feasibility by inserting charging stations during repair
  if HierarchicalAcceptance($\tilde{S}, S, T$) **then**
    $S \leftarrow \tilde{S}$
    if HierarchicalBetter(S, S^*) **then**
      $S^* \leftarrow S$
    end if
    Update operator scores based on solution quality
  end if
  Update temperature T
  if i mod segment_size $= 0$ **then**
    Update operator weights and reset scores
  end if
end for
return S^*

Algorithm 3 Hierarchical Comparison

Require: Solutions S_1, S_2
Ensure: Boolean indicating if S_1 is better than S_2

$v_1 \leftarrow$ number_of_vehicles(S_1)
$v_2 \leftarrow$ number_of_vehicles(S_2)
if $v_1 < v_2$ **then**
  return true
else if $v_1 > v_2$ **then**
  return false
else ▷ Same number of vehicles
  return total_distance(S_1) $<$ total_distance(S_2)
end if

Algorithm 4 Hierarchical Acceptance Criterion

Require: New solution S_{new}, current solution $S_{current}$, temperature T
Ensure: Boolean indicating if S_{new} should be accepted

if hierarchical_better(S_{new}, $S_{current}$) **then**
  return true ▷ Always accept improvements
end if
$v_{new} \leftarrow$ number_of_vehicles(S_{new})
$v_{current} \leftarrow$ number_of_vehicles($S_{current}$)
if $v_{new} > v_{current}$ **then**
  return false ▷ NEVER accept more vehicles
end if
▷ Same vehicles, worse distance: apply simulated annealing
$\Delta \leftarrow$ total_distance(S_{new}) $-$ total_distance($S_{current}$)
$probability \leftarrow \exp(-\Delta/T)$
return random() $<$ $probability$

4.3. Tabu Search Refinement

Once the ALNS procedure is completed, the best solution obtained is further refined using a TS algorithm, as described in Algorithm 5. The purpose of this final phase is to intensify the search around high-quality solutions through local improvements. The TS explores a neighborhood defined by customer relocation moves,where customers are moved within or between routes while maintaining capacity, time windows and battery feasibility constraints. Tabu lists are used to prevent cycling, while aspiration criteria allow tabu moves if they lead to an improved solution. The same hierarchical objective used in the ALNS phase is applied during the TS refinement.

Algorithm 5 Tabu Search Enhancement

Require: Initial solution S_0, tabu tenure τ, maximum iterations I_{TS}
Ensure: Improved solution S^*

$S \leftarrow S_0$, $S^* \leftarrow S_0$
Initialize tabu list
for $i = 1$ to I_{TS} **do**
  Generate neighborhood using customer relocation moves
  Select the best admissible neighbor according to hierarchical objective
  Apply aspiration criterion to allow tabu moves that improve S^*
  if no admissible neighbor exists **then**
    break
  end if
  $S \leftarrow$ selected neighbor
  Update tabu list
  if HierarchicalBetter(S, S^*) **then**
    $S^* \leftarrow S$
  end if
end for
return S^*

5. Experiments and Results

5.1. Experiments

All experiments were conducted on the Nibi HPC cluster operated by Alliance Canada. Each algorithm was submitted as a SLURM batch job and each job was allocated 16 GB of memory and 2 CPU cores on dedicated compute nodes. We use Python 3.8+ with standard scientific libraries (NumPy). We used the EVRPTW benchmark of Schneider et al. [6], which is derived from the Solomon benchmark [29]. These instances include clustered (C), random (R), and mixed (RC) customer distributions, with problem sizes ranging from 5 to 100 customers and 2 to 21 charging stations, while explicitly accounting for battery capacity and charging process constraints. To ensure a fair and consistent comparison, all algorithms were evaluated under the same experimental conditions using the common parameter settings reported in Table 2. We evaluate four ALNS-based variants that differ in their initialization strategy and the inclusion of an additional improvement phase. Two variants rely on the GTONNH for initial solution construction, with and without a subsequent TS refinement phase, resulting in the GTONNH-ALNS and GTONNH-ALNS-TS approaches. For comparison purposes, we also consider two baseline variants based on the initial solution proposed by Schneider et al. [6]: S-ALNS and its extended version with TS, hereafter referred to as S-ALNS and S-ALNS-TS, respectively.

Table 2. Algorithm Parameters

Category	**Parameter**	**Value**
ALNS	Maximum iterations	100
	Segment size	100 iterations
	Cooling rate (α)	0.9995
	Removal percentage	10% – 40% of customers
	Station optimization frequency	Every 100 iterations
	Scores ($\sigma_1, \sigma_2, \sigma_3$)	33, 9, 13
	Initial operator weights	1.0 (equal for all operators)
	Reaction factor	0.1
	Number of operators	10 removal, 5 insertion
Tabu Search	Maximum TS iterations	1000
	Tabu tenure	10
	Early stopping threshold	100 non-improving iterations
GTONNH	Distance weight (δ_1)	1.0
	Time weight (δ_2)	0.5
	Spare time weight (δ_3)	0.3

6. Results & Discussion

The results reported in Tables 3–6 compare the performance of the four proposed ALNS variants. For each instance, the algorithms are evaluated using a lexicographic comparison criterion: priority is given first to minimizing the number of vehicles, followed by minimizing the total traveled distance, and finally by minimizing the computational time (in seconds) in case of ties. Fleet size is prioritized given the substantial economic costs associated with operating and acquiring electric vehicles. The algorithm achieving the best performance according to this criterion is considered the best for the corresponding instance.

Tables 3–5 report results for small and medium-sized instances with clustered (C), random (R), and mixed (RC) customer distributions, respectively, while Table 6 presents results for large-scale instances with 100 customers.

The experimental results highlight clear performance differences between the four ALNS variants, depending on instance size and structure. On small and medium instances, hybrid approaches incorporating a TS phase consistently improve solution quality. In particular, **GTONNH-ALNS-TS** achieves the best performance on C and RC instances, attaining the best solution on 41.7% and 45.5% of the instances, respectively. These improvements are mainly driven by more frequent reductions in the number of vehicles, with GTONNH-ALNS-TS reaching the minimum fleet size on all C instances and on more than 90% of RC

Table 3. Comparison of ALNS variants on C-type EVRPTW instances (small/medium)

Instance	S-ALNS			GTONNH-ALNS			S-ALNS-TS			GTONNH-ALNS-TS		
	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU
c101C10	4	401.57	0.18	4	401.57	0.15	3	414.51	0.60	**3**	**414.51**	**0.51**
c101C5	3	296.09	0.04	3	293.20	0.03	3	250.04	0.12	**2**	**257.75**	**0.08**
c103C15	4	371.70	0.35	4	427.56	0.53	3	414.18	23.36	**3**	**408.97**	**15.53**
c103C5	2	165.67	0.04	1	184.50	0.05	1	184.50	0.05	**1**	**184.50**	**0.04**
c104C10	3	321.97	0.15	3	315.90	0.19	**2**	**273.93**	**5.24**	2	322.10	4.51
c106C15	4	346.96	0.32	3	373.35	0.49	**3**	**275.13**	**13.46**	3	373.35	5.17
c202C10	2	266.66	0.16	2	289.98	0.24	**2**	**265.67**	**5.28**	2	283.82	6.42
c202C15	3	414.93	0.47	2	461.11	1.00	3	507.72	30.50	**2**	**459.34**	**34.90**
c205C10	**2**	**258.66**	**0.17**	2	258.66	0.18	2	258.66	4.02	2	258.66	3.26
c206C5	1	254.13	0.06	1	274.89	0.06	1	**254.13**	**0.06**	**1**	274.89	0.06
c208C15	3	325.92	0.51	2	380.98	0.64	**2**	**300.55**	**30.62**	2	301.83	32.90
c208C5	2	205.00	0.04	**1**	**164.34**	**0.05**	1	183.57	0.08	1	164.34	0.07

Table 4. Comparison of ALNS variants on R-type EVRPTW instances (small/medium)

Instance	S-ALNS			GTONNH-ALNS			S-ALNS-TS			GTONNH-ALNS-TS		
	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU
r102C10	4	263.25	0.13	**4**	**263.25**	**0.13**	4	263.25	3.55	4	263.25	4.06
r102C15	7	466.73	0.35	6	488.70	0.29	5	513.84	5.32	**5**	**504.85**	**5.82**
r103C10	3	207.87	0.14	4	207.87	0.12	**2**	**207.05**	**1.53**	2	210.47	0.61
r104C5	**2**	**136.69**	**0.04**	**2**	**136.69**	**0.04**	2	136.69	0.11	2	136.69	0.09
r105C15	4	344.16	0.36	6	400.33	0.33	**4**	**341.46**	**14.11**	4	344.16	9.09
r105C5	**2**	**156.08**	**0.03**	**2**	**156.08**	**0.03**	2	156.08	0.06	2	156.08	0.06
r201C10	4	247.94	0.12	2	269.37	0.25	**2**	**236.51**	**4.89**	2	269.37	4.89
r202C5	2	142.65	0.04	**1**	**152.59**	**0.06**	**1**	**152.59**	**0.06**	**1**	**152.59**	**0.06**
r203C10	**2**	**273.69**	**0.26**	2	296.39	0.29	2	286.54	12.30	2	296.39	9.15
r203C5	2	199.54	0.04	2	199.54	0.04	**1**	**199.54**	**0.07**	1	199.54	0.09
r209C15	**3**	**362.11**	**0.53**	3	425.74	0.78	3	430.53	28.51	3	378.90	28.10

Table 5. Comparison of ALNS variants on RC-type EVRPTW instances (small/medium)

Instance	S-ALNS			GTONNH-ALNS			S-ALNS-TS			GTONNH-ALNS-TS		
	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU
rc102C10	**5**	**436.05**	**0.12**	5	436.05	0.16	5	436.05	10.79	5	436.05	13.83
rc103C15	5	442.10	0.38	5	442.10	0.40	4	397.67	11.87	**4**	**397.67**	**7.54**
rc105C5	**3**	**238.05**	**0.03**	3	238.05	0.04	3	238.05	0.07	3	238.05	0.08
rc108C10	4	402.08	0.13	4	402.08	0.16	3	417.35	0.55	**3**	**417.35**	**0.47**
rc108C15	**3**	**386.58**	**0.48**	3	386.45	0.59	3	386.45	7.26	3	386.45	7.41
rc108C5	**3**	**316.51**	**0.05**	3	316.51	0.06	3	316.51	0.22	3	316.51	0.26
rc201C10	5	369.73	0.10	3	397.76	0.16	2	367.08	6.95	**2**	**357.35**	**7.03**
rc202C15	5	603.98	0.49	3	644.09	0.54	2	526.25	25.24	**2**	**496.24**	**41.21**
rc204C5	**1**	**185.16**	**0.05**	**1**	**185.16**	**0.05**	**1**	**185.16**	**0.05**	1	185.16	0.06
rc205C10	3	372.67	0.15	3	429.86	0.19	2	440.50	11.39	**2**	**354.07**	**14.39**
rc208C5	2	200.18	0.04	2	200.18	0.04	**1**	**179.87**	**0.06**	1	179.87	0.08

instances. On R instances, the performance gap is less pronounced. S-ALNS and S-ALNS-TS remain highly competitive, achieving the best solutions on approximately 45% of the instances. This suggests that the benefit of initialization and intensification mechanisms is more pronounced when customer locations exhibit structured patterns.

The advantage of GTONNH-ALNS-TS becomes even more evident on large-scale instances with 100 customers. As shown in Figures 2 and 1, this hybrid approach produces the best solutions on more than 60% of tested instances and by reaching the minimum number of vehicles in nearly 95% of the cases. Compared to baseline S-ALNS, this variant achieves a consistent reduction in fleet size, corresponding to an average decrease of more than one vehicle, which represents a substantial improvement given the high acquisition and operational costs of EVs. From a computational perspective, clear trade-offs emerge. Variants

Table 6. Comparison of ALNS variants on large EVRPTW instances

Instance	S-ALNS			GTONNH-ALNS			S-ALNS-TS			GTONNH-ALNS-TS		
	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU	#V	Dist.	CPU
c101_21	15	1419.32	88.60	15	1400.63	99.05	15	1572.85	1533.32	**14**	**1349.72**	**2482.01**
c102_21	17	1678.90	78.88	13	1227.43	110.14	15	1509.42	1031.47	**12**	**1175.37**	**2808.36**
c201_21	6	782.65	111.27	5	664.27	172.24	9	998.20	654.33	**4**	**664.27**	**3937.16**
c202_21	7	957.63	118.41	5	932.69	170.40	6	880.42	1054.14	**5**	**872.02**	**1543.40**
c205_21	6	804.52	129.79	5	762.62	130.89	5	800.72	908.57	**5**	**777.15**	**2077.67**
c206_21	5	891.89	154.11	5	835.07	146.06	5	950.96	1347.68	**5**	**920.77**	**1393.82**
r203_21	6	949.93	128.81	5	999.80	174.06	5	972.14	381.99	**3**	**1066.37**	**973.34**
r204_21	4	811.08	199.10	4	837.69	196.80	3	820.10	872.94	**4**	**816.06**	**540.80**
r205_21	5	1075.73	152.34	4	1134.94	169.94	4	1086.4	680.91	**3**	**1132.87**	**967.64**
r206_21	5	1052.52	143.38	4	1078.33	174.31	4	1104.4	568.08	**4**	**1106.42**	**567.13**
r207_21	4	872.47	174.10	**3**	**857.71**	**271.61**	3	1049.64	661.14	3	955.45	1071.94
r208_21	**3**	**788.14**	**221.42**	4	790.46	219.58	3	820.08	920.34	3	857.57	609.53
r209_21	5	1004.31	160.31	5	1029.82	137.12	4	1037.64	666.17	**3**	**937.26**	**1077.98**
rc201_21	7	1625.63	102.76	7	1537.14	115.91	7	1642.41	433.11	**5**	**1631.11**	**954.95**
rc202_21	6	1272.73	109.30	6	1330.99	136.71	6	1264.67	419.11	**5**	**1489.45**	**827.49**
rc203_21	6	1113.35	137.43	5	1109.69	126.54	5	1267.03	436.57	**5**	**1101.91**	**452.16**
rc204_21	5	983.13	133.15	5	984.84	153.50	4	1051.56	678.17	**4**	**1008.22**	**677.07**
rc205_21	5	1340.20	154.07	**5**	**1245.64**	**134.34**	5	1310.45	504.96	5	1346.66	560.62

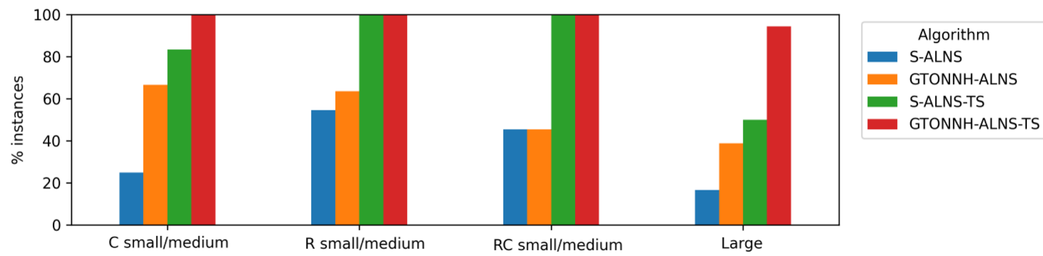

Figure 1. ALNS variants for fleet size minimization

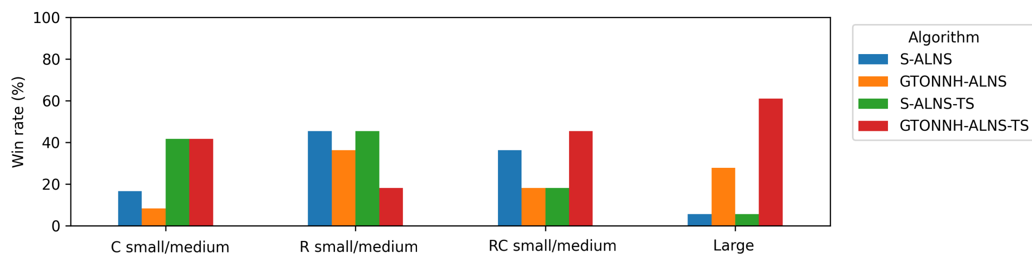

Figure 2. Percentage of best solutions by ALNS variants

relying on the GTONNH greedy initialization generally construct feasible solutions faster than those using the Schneider initialization, often allowing the ALNS phase to converge more efficiently. On small and medium instances, the average CPU-time gap between S-ALNS and GTONNH-ALNS is positive, indicating that GTONNH-based variants are faster on most instances. The inclusion of TS significantly increases computational effort, particularly on large instances, where CPU-time gaps can reach several hundreds or even thousands of seconds compared to non-TS variants. However, this additional cost enables deeper intensification of the search and substantially improves robustness and solution quality, especially in terms of fleet size reduction. Overall, the results suggest that **GTONNH-ALNS-TS provides the most effective balance between solution quality and computational effort** for large and complex EVRPTW instances.

7. Conclusion

This paper investigated the EVRPTW and introduced a hybrid solution framework combining GTONNH, ALNS, and TS. By leveraging fast greedy initialization, large-scale neighborhood exploration, and targeted search intensification, our work produces high-quality feasible solutions while satisfying all operational constraints related to battery capacity, recharging, and time windows. Computational experiments conducted on standard EVRPTW benchmark instances demonstrate that the proposed hybrid variants, in particular GTONNH-ALNS-TS, achieve competitive solutions. The results show that good-quality solutions are obtained within reasonable computational times, with average solution times remaining on the order of seconds for small and medium instances and remaining reasonable for large-scale problems. These findings highlight the effectiveness of the proposed framework in balancing solution quality and computational efficiency, especially when fleet size minimization is prioritized. While the proposed framework builds on established metaheuristic components, its contribution lies in their careful integration, as evidenced by the strong empirical performance demonstrated across a wide range of benchmark instances.

Future research will focus on extending the proposed framework to other EVRP variants in order to better capture the environmental and economic complexities of real-world logistics systems. Promising directions include the integration of additional cost components, such as energy pricing and carbon emissions, as well as the study of more complex problem settings involving heterogeneous fleets and alternative charging technologies. Another important avenue for future work involves dynamic EVRP variants, where customer requests, travel times, or energy availability evolve over time, requiring adaptive and real-time routing decisions.

References

[1] U.S. Environmental Protection Agency. *Electric Vehicle Myths*. https://www.epa.gov/greenvehicles/electric-vehicle-myths. Accessed: 2025-12-19. 2025.

[2] A.P. Møller - Mærsk A/S. *Transitioning to Electric Vehicles in Logistics: Future Outlook*. https://www.maersk.com/insights/sustainability/2025/06/04/transitioning-to-electric-vehicles-in-logistics. Accessed: 2025-12-19. 2025.

[3] J. Mohammed and J. Villegas. "Total impact of electric vehicle fleet adoption in the logistics industry". In: *Frontiers in Sustainability* 4 (2023), p. 1158993.

[4] D. Singh, S. K. Singh, S. Singh, U. Singh, and B. Singh. "Electric Vehicles: Types, Advantages, Difficulties, and Possible Remedies for Broad Adoption: A Review". In: *2025 International Conference on Intelligent Control, Computing and Communications (IC3)*. IEEE. 2025, pp. 1324–1328.

[5] J. Lin, W. Zhou, and O. Wolfson. "Electric vehicle routing problem". In: *Transportation Research Procedia* 12 (2016), pp. 508–521. DOI: 10.1016/j.trpro.2016.02.007. URL: https://doi.org/10.1016/j.trpro.2016.02.007.

[6] M. Schneider, A. Stenger, and D. Goeke. "The Electric Vehicle Routing Problem with Time Windows and Recharging Stations". In: *Transportation Science* 48.4 (2014), pp. 500–520.

[7] G. B. Dantzig and J. H. Ramser. "The Truck Dispatching Problem". In: *Management Science* 6.1 (1959), pp. 80–91. DOI: 10.1287/mnsc.6.1.80.

[8] S. Sabet and B. Farooq. "Green vehicle routing problem: State of the art and future directions". In: *IEEE Access* 10 (2022), pp. 101622–101642.

[9] S. Erdoğan and E. Miller-Hooks. "A Green Vehicle Routing Problem". In: *Transportation Research Part E* 48.1 (2012), pp. 100–114.

[10] R. G. Conrad and M. A. Figliozzi. "The Recharging Vehicle Routing Problem". In: *Transportation Research Part E* 47.6 (2011), pp. 123–145.

[11] P. Gupta, Devanand, A. K. Iannillo, J. A. Meira, R. State, and D. D'Aversa. "Modelling Electric Vehicle Routing Problem with Heterogeneous Fleet for Simultaneous Pickup and Delivery". In: *International Conference on Computer Aided Systems Theory*. Springer. 2024, pp. 107–116.

[12] N. Khanlarzade and B. Yousefi Yegane. "Hybrid Ant Colony optimization and Variable Neighborhood Search algorithm for Electric and Fossil Fuel Vehicle Routing". In: *Journal of Industrial and Systems Engineering* 17.2 (2025), pp. 26–43.

[13] V. F. Yu and P. T. Anh. "The electric vehicle routing problem with time windows, partial recharges, and covering locations". In: *International Transactions in Operational Research* (2025).

[14] Z. Zheng, S. Liu, and Y.-S. Ong. "Hybrid memetic search for electric vehicle routing with time windows, simultaneous pickup-delivery, and partial recharges". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2025).

[15] G. Rahmanifar, M. Mohammadi, M. Hajiaghaei-Keshteli, C. Colombaroni, G. Fusco, and F. Gholian-Jouybari. "Two-Echelon Electric Vehicle Routing Problem with Battery Swap Stations on Real Network". In: *IFAC-PapersOnLine* 58.5 (2024), pp. 46–51.

[16] H. Park and C. Lee. "An exact algorithm for maximum electric vehicle flow coverage problem with heterogeneous chargers, nonlinear charging time and route deviations". In: *European Journal of Operational Research* 315.3 (2024), pp. 926–951.

[17] A. L. Souza, M. Papini, P. H. Penna, and M. J. Souza. "A flexible variable neighbourhood search algorithm for different variants of the Electric Vehicle Routing Problem". In: *Computers & Operations Research* 168 (2024), p. 106713.

[18] Z.-H. Nie, Q. Yang, E. Zhang, D. Liu, and J. Zhang. "Ant colony optimization for electric vehicle routing problem with capacity and charging time constraints". In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2022, pp. 480–485.

[19] Y. Zhang, S. Zhou, X. Ji, B. Chen, H. Liu, Y. Xiao, and W. Chang. "The Mathematical Model and an Genetic Algorithm for the Two-Echelon Electric Vehicle Routing Problem". In: *Journal of Physics: Conference Series*. Vol. 1813. 1. IOP Publishing. 2021, p. 012006.

[20] I. Zaidi, A. Oulamara, L. Idoumghar, and M. Basset. "Electric vehicle charging scheduling problem: Heuristics and metaheuristic approaches". In: *SN Computer Science* 4.3 (2023), p. 283.

[21] J. Euchi and A. Yassine. "A hybrid metaheuristic algorithm to solve the electric vehicle routing problem with battery recharging stations for sustainable environmental and energy optimization". In: *Energy Systems* 14.1 (2023), pp. 243–267.

[22] C.-T. Feng, Y.-H. Jia, Q. Yang, W.-N. Chen, and H. Jiang. "A Bilevel Hybrid Genetic Algorithm for Capacitated Electric Vehicle Routing Problem". In: *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2024, pp. 1–8.

[23] Q. Liu, P. Xu, Y. Wu, and T. Shen. "A hybrid genetic algorithm for the electric vehicle routing problem with time windows". In: *Control Theory and Technology* 20.2 (2022), pp. 279–286.

[24] L. Fan. "A hybrid adaptive large neighborhood search for time-dependent open electric vehicle routing problem with hybrid energy replenishment strategies". In: *PLoS One* 18.9 (2023), e0291473.

[25] T. Stamadianos, N. A. Kyriakakis, M. Marinaki, and Y. Marinakis. "A hybrid simulated annealing and variable neighborhood search algorithm for the close-open electric vehicle routing problem". In: *Annals of Mathematics and Artificial Intelligence* (2023), pp. 1–24.

[26] Y. Jin, X. Bao, and Z. Wang. "A two-stage hybrid heuristic approach combining genetic algorithm and variable neighborhood descent for the clustered electric vehicle routing problem". In: *Expert Systems with Applications* 298 (2026), p. 129848.

[27] V. Q. Hien, T. C. Dao, and H. T. T. Binh. "A greedy search based evolutionary algorithm for electric vehicle routing problem". In: *Applied Intelligence* 53.3 (2023), pp. 2908–2922.

[28] N. Azi, M. Gendreau, and J.-Y. Potvin. "An adaptive large neighborhood search for a vehicle routing problem with multiple routes". In: *Computers & Operations Research* 41 (2014), pp. 167–173.

[29] M. M. Solomon. "Algorithms for the vehicle routing and scheduling problems with time window constraints". In: *Operations research* 35.2 (1987), pp. 254–265.