A Novel Bi-Level Framework for Electric Vehicle Charging Scheduling Using Reinforcement Learning and Transfer Learning

1[st] Abdennour Azerine
IRIMAS UR 7499
Université de Haute-Alsace
F-68100 Mulhouse, France
abdennour.azerine@uha.fr

2[nd] Mahmoud Golabi
IRIMAS UR 7499
Université de Haute-Alsace
F-68100 Mulhouse, France
mahmoud.golabi@uha.fr

3[rd] Lhassane Idoumghar
IRIMAS UR 7499
Université de Haute-Alsace
F-68100 Mulhouse, France
lhassane.idoumghar@uha.fr

***Abstract*—The growing adoption of electric vehicles (EVs) presents significant challenges for efficiently managing charging infrastructure, a critical component in the transition toward sustainable transportation. This study addresses the Electric Vehicle Charging Scheduling Problem (EVCSP), a computationally complex optimization problem aimed at maximizing the number of satisfied charging demands while adhering to grid and infrastructure constraints. To address this challenge, a novel hybrid bi-level optimization framework is proposed. At the upper level, advanced reinforcement learning (RL) methods, specifically Q-learning, Double Q-learning, SARSA, and Expected SARSA, are employed to generate adaptive EV-to-charger assignment policies by iteratively exploring and learning from decision environments. These assignments, which may not fully satisfy operational constraints, are subsequently refined at the lower level through a mathematical programming model that optimizes energy allocation under realistic grid capacity and charger availability constraints. Additionally, transfer learning is incorporated to enhance computational efficiency by transferring knowledge gained during the RL-based assignment process to the optimization phase. Extensive experimental evaluations demonstrate the framework's scalability and effectiveness, with SARSA combined with transfer learning emerging as the most reliable method, particularly for larger and more complex problem instances.**

***Index Terms*—Electric Vehicle, Charging Scheduling, Bi-level Optimization, Reinforcement Learning, Transfer Learning**

I. Introduction

The transportation sector remains a major contributor to global energy consumption and greenhouse gas emissions, with private vehicles and light commercial vehicles accounting for nearly half of transportation-related CO_2 emissions in 2022 [1]. In the EU alone, transportation constituted 31% of final energy consumption and emitted 903 million metric tons of CO_2, reflecting a 7.6% increase compared to 2021 [2]. To address this, the EU aims to cut car and van emissions by 50% by 2030 and transition to zero-emission vehicles by 2035 [3]. Achieving these targets hinges on the widespread adoption of electric vehicles (EVs), which reached 14 million units in 2023, representing 18% of global car sales [4]. However, this rapid growth has revealed critical infrastructure bottlenecks. In the EU, battery electric vehicle (BEV) sales rose 18-fold from 2017 to 2023, while public charging infrastructure expanded only six-fold [5]. Similar trends are evident in the U.S., where EV registrations increased by 140% between early 2023 and September 2024, while public charging stations grew by only 22% [6]. To meet demand, the EU must install 1.4 million new charging points annually by 2030, a tenfold increase over current rates [5]. Additionally, the rising prevalence of EVs strains power grids, causing congestion, voltage instability, and reduced efficiency [7]. These challenges highlight the urgent need for optimized charging operations to balance user satisfaction, grid reliability, and infrastructure constraints.

The Electric Vehicle Charging Scheduling Problem (EVCSP) has emerged as a critical research focus to address the challenges posed by the rapid adoption of electric vehicles (EVs). This problem focuses on optimizing the allocation of limited charging resources to balance operational efficiency with user-centric objectives. Early studies emphasized station-oriented goals, such as maximizing station revenue [8], minimizing grid power losses [9], and optimizing grid utilization [10]. Recently, the focus has shifted toward user-centric objectives, including minimizing charging delays [11] and maximizing the fulfillment of charging requests [12], reflecting a growing emphasis on aligning optimization strategies with user needs to enhance the practical applicability of EV charging systems.

Realistic implementations of the EVCSP must address various constraints, including EV-specific factors such as arrival and departure times and energy requirements, as well as infrastructure limitations like charger heterogeneity, grid capacity restrictions, and limited charger availability [13]. To effectively manage these constraints, preemption strategies have been proposed, enabling interruptions and resumptions of charging sessions to alleviate congestion and optimize resource utilization [14]. A prominent contribution to this area is the work of Zaidi et al. [15], who introduced a comprehensive framework for the EVCSP under realistic constraints. Their approach defined demand satisfaction as fulfilling exact energy

979-8-3315-3431-8/25/$31.00 ©2025 IEEE

requests and prioritized maximizing fully satisfied demands. Using a hybrid framework based on heuristic and metaheuristic techniques, their study highlighted the computational complexity of the problem and the critical need for scalable, high-quality solutions.

The EVCSP is a strongly $\mathcal{NP}$-hard problem, making exact methods impractical for large-scale instances [15]. To address this, metaheuristic approaches such as genetic algorithms [16], [17], particle swarm optimization [12], and tabu search [18] have been widely adopted due to their flexibility in navigating complex and constrained solution spaces. However, the growing complexity of real-world scenarios has driven the search for alternative methodologies.

Reinforcement learning (RL) has recently emerged as a powerful alternative for solving EVCSP by leveraging iterative learning to tackle decision-intensive problems in high-dimensional environments [19]. Applications of RL in EVCSP have targeted objectives such as maximizing station profit [20], minimizing charging costs [21], and reducing charging delays [22], offering dynamic optimization capabilities through interaction with the environment. Despite its promise, RL often struggles with convergence and computational efficiency in highly constrained settings with large decision spaces [19]. These challenges underscore the need for hybrid frameworks that combine RL with complementary optimization methods to improve scalability, efficiency, and solution quality.

This study proposes a hybrid bi-level framework combining reinforcement learning (RL) and mathematical optimization. In the first stage, RL algorithms (Q-learning, Double Q-learning, SARSA, Expected SARSA) assign charging demands to chargers, though initial solutions may violate constraints. In the second stage, a mathematical model validates these assignments, keeping only those that meet grid capacity and feasibility requirements. Through iteration, RL learns to produce high-quality, constraint-compliant solutions while optimization ensures alignment with grid limits.

To further enhance efficiency, transfer learning is integrated into the framework. In the first phase, the RL agent focuses on learning feasible demand-to-charger assignments while respecting charger availability constraints. In the second phase, the pre-trained RL policy is refined using feedback from the mathematical optimization model, which incorporates energy allocation constraints. This phased transfer of knowledge ensures the framework efficiently balances exploration (via RL) and rigorous constraint satisfaction (via optimization), ultimately improving scalability and solution quality for the EVCSP.

The key contributions of this study are fourfold. First, it introduces a bi-level optimization framework specifically tailored to the EVCSP, where the upper-level learning-based assignment is tightly coupled with a lower-level exact energy allocation model. Second, it leverages reinforcement learning as the core mechanism for solving the combinatorial assignment task, enabling adaptive, data-driven decision-making in a complex, high-dimensional space. Third, the framework incorporates a phased transfer learning strategy that significantly enhances convergence speed and generalization, allowing the RL agent to reuse knowledge from earlier phases of training and adapt to evolving feasibility constraints. Fourth, and most notably, the proposed integration of reinforcement learning, transfer learning, and exact mathematical optimization within a bi-level structure represents, to the best of the authors' knowledge, a novel contribution not only within the literature on EVCSPs but also in the broader context of combinatorial optimization. This work establishes a generalizable architecture for learning-driven optimization under hard constraints, bridging adaptive decision-making and rigorous feasibility enforcement in a unified framework.

The paper is organized as follows: Section II defines the studied problem, while Section III details the proposed bi-level solution framework. An upper bound on the maximum number of accepted charging demands is derived in Section IV. Section V presents computational experiments and analyzes the results. Finally, Section VI concludes with key findings and future research directions.

II. Problem Definition

This study investigates the preemptive realistic EVCSP introduced by Zaidi et al. [15]. While the mathematical formulation of the problem is detailed in the referenced work, a summarized description is provided here to outline its key structure and constraints.

The EVCSP involves assigning a set of charging demands, denoted as $J = \{1, \ldots, n\}$, to a set of chargers, $M = \{1, \ldots, m\}$, over a discrete time horizon $\mathcal{T} = \{1, \ldots, T\}$. Each charger $i \in M$ delivers power at a constant rate w_i (kW), while the global grid capacity w^G (kW) imposes an upper limit on the total power consumption of all chargers at any time. This constraint reflects the practical limitations of grid infrastructure, as w^G is typically insufficient to support all chargers operating simultaneously at full capacity.

The scheduling process incorporates several operational constraints to ensure feasibility and efficient utilization of resources. Each charger can accommodate only one charging demand at a time, preventing overlapping sessions on the same charger. Once initiated, a charging session occupies the assigned charger for its entire duration, but the session may be paused and resumed later. This preemptive scheduling mechanism provides flexibility, enabling the system to fulfill energy requirements within the specified time window while balancing competing demands.

Each charging demand $j \in J$ is characterized by arrival time $r_j \in \mathcal{T}$, departure time $d_j \in \mathcal{T}$, and energy requirement e_j (kWh). When assigned to charger i, it requires cumulative processing time $\lfloor \frac{e_j}{w_i} \rceil$, which may be scheduled preemptively in one or more intervals within $[r_j, d_j)$. The demand is accepted if the total allocated processing time across all scheduled intervals equals p_{ij}; otherwise, it is rejected.

The scheduling horizon is partitioned into uniform discrete time intervals τ, within which decisions are made regarding the allocation of charging resources. The primary objective is to maximize the number of charging demands that are fully satisfied, while ensuring compliance with constraints related to charger availability, global grid capacity, and individual energy requirements.

III. Solution Methodology

The proposed framework addresses the studied EVCSP using a hybrid *bi-level optimization approach*. At the upper level, RL generates adaptive EV-to-charger assignment policies, iteratively refining these policies through interactions with the system environment. At the lower level, a mathematical programming model optimizes energy allocation to ensure that grid capacity and charger availability constraints are satisfied. This combination ensures that the RL-generated policies are both efficient and operationally feasible.

A key feature of this approach is the *proportional discretization* of processing time into discrete intervals, referred to as *energy bins*. These bins divide the proportion of energy allocated to a demand into n_b intervals. In case of no allocation, it corresponds to the first index of the energy bin, while the full satisfaction is given in the last index ($n_b - 1$). This discretization will be explained further in the context of the reinforcement learning framework.

A. Reinforcement Learning Framework

The RL framework learns adaptive EV-to-charger assignment policies while processing demands J sorted in ascending order of their arrival times r_j. Sorting ensures that the next demand $j+1$ is processed no earlier than the current demand j, guaranteeing deterministic state transitions without requiring explicit sequence management.

1) State and Action Representation: The state in the RL framework is represented as a tuple (j, k, b), where j is the index of the current demand, k indicates charger availability, and b represents the proportion of the demand's energy requirement that can be satisfied. Charger availability (k) is binary, with $k_i = 1$ indicating availability of charger i, and $k_i = 0$ otherwise. The energy bin index (b) quantifies partial satisfaction levels and is discretized into n_b equal intervals.

The action a determines whether to assign a demand to a charger or reject it. Specifically, a can either select one of the m chargers ($a = 0, \ldots, m-1$) or reject the demand ($a = m$).

2) Reward Function: The reward function R_j for demand j is designed to maximize accepted demands while penalizing infeasible assignments to unavailable chargers as invalid actions, and encouraging strategic rejection of infeasible demands by taking the action fixed by value m, as described in the previous section. Let N_{A} denote the number of accepted demands. For each demand j:

$$R_j = \begin{cases} N_{\text{A}}, & \text{if assigned to charger and accepted,} \\ -N_{\text{A}}, & \text{if assigned to unavailable charger,} \\ -0.1 \cdot N_{\text{A}}, & \text{if assigned to charger} \\ & \text{but energy allocation fails,} \\ 0.1 \cdot N_{\text{A}}, & \text{if rejected via dedicated index } m. \end{cases} \quad (1)$$

B. Computation of State Components

The key components of the state representation, k (availability index) and b (energy bin index), are calculated in the following subsections.

1) Availability Index: The availability index k is a binary vector of size $m+1$, where each element k_i ($i = 0, \ldots, m$) represents the availability of a charger or the rejection action for a given demand j with arrival time r_j. For each charger i, $k_i = 1$ if $tc_i \leq r_j$, where tc_i is the availability time of charger i; otherwise, $k_i = 0$. The rejection action is always available, and thus $k_m = 1$. The vector $k = [k_0, k_1, \ldots, k_{m-1}, k_m]$ provides a compact representation of the availability of all chargers and the rejection action. The values of tc_i are initially provided as inputs, representing the current availability times of the chargers. However, tc_i is updated dynamically when a demand is accepted, following the procedure described in Algorithm 2, which will be introduced later.

2) Energy Bin Index: Algorithm 1 computes the proportional charging time allocated to each demand for a given action by simulating the charging process over the interval $[r_j, d_j)$. The algorithm calculates the total processing time p_{ij} required to satisfy demand j and iteratively decrements p_{ij} based on the available energy. The proportion of the satisfied energy, denoted as vb_i, is then mapped to the largest feasible energy bin b, where full satisfaction is represented by the highest bin index, $b = n_b - 1$.

To facilitate the representation of partial energy satisfaction, the interval $[0, 1]$ is divided into n_b equal bins, where each bin corresponds to a specific range of energy proportions. For example, if $n_b = 5$, the bins are defined as $[0.0, 0.25, 0.5, 0.75, 1.0]$. In this case, each bin represents a range of 25% of the energy requirement. If $vb_i = 0.7$, this indicates that 70% of the requested energy is satisfied. According to the discretization scheme, $vb_i = 0.7$ is mapped to the third bin (index $b = 3$), which corresponds to the interval $[0.5, 0.75]$. This mapping process provides a structured mechanism for representing and incorporating partial energy satisfaction levels into the reinforcement learning framework, ensuring computational efficiency and consistency.

C. Reinforcement Learning Algorithms

In this study, four variations of RL are utilized to address the EVCSP: Q-learning, Double Q-learning, SARSA, and Expected SARSA. While these algorithms share the same general framework, they differ in their Q-value update mechanisms, each offering unique advantages for learning optimal policies in the problem context.

Q-learning [23] is a model-free RL algorithm that updates Q-values using the Bellman equation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \quad (2)$$

where s is the current state, a is the action taken, r is the reward, s' is the next state, α is the learning rate, and γ is the

Algorithm 1 Calculate Remaining Energy Proportion

Input: Demand j, charger power rates w_i, energy requirement e_j, arrival time r_j, departure time d_j, remaining energy W_t^{left}.
Output: Proportion vb_i for each charger i.
for each charger $i = 0$ to $m-1$ **do**
  $p_{ij} \leftarrow \frac{e_j}{w_i}$. ▷ Total processing time for demand j on charger i and $pr \leftarrow p_{ij}$.
  for each time slot $t = r_j$ to $d_j - 1$ **do**
    if $W_t^{left} \geq w_i$ **then**
      $pr \leftarrow pr - 1$.
      if $pr = 0$ **then**
        Break.
      end if
    end if
  end for
  $vb_i \leftarrow 1 - \frac{pr}{p_{ij}}$.
end for
Return: vb_i.

discount factor. This approach enables the agent to learn an optimal policy by maximizing expected future rewards.

Double Q-learning [24] addresses the overestimation bias inherent in Q-learning by maintaining two Q-value functions, Q_1 and Q_2. To enhance stability and accuracy in estimating Q-values, the update function alternates between the two functions:

$$\begin{aligned} Q_1(s,a) \leftarrow\, & Q_1(s,a) + \\ & \alpha \left[r + \gamma Q_2(s', \arg\max_{a'} Q_1(s',a')) - Q_1(s,a) \right] \end{aligned} \tag{3}$$

SARSA (State-Action-Reward-State-Action) [25] is an on-policy RL algorithm that updates Q-values based on the current policy. Its update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s',a') - Q(s,a) \right] \tag{4}$$

where a' is the action selected under the current policy in state s'. SARSA is particularly effective in environments with stochastic transitions.

Expected SARSA [25] extends SARSA by incorporating the expected value of Q-values under the current policy. The update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \sum_{a'} \pi(a'|s') Q(s',a') - Q(s,a) \right] \tag{5}$$

where $\pi(a'|s')$ is the probability of selecting action a' in state s'. This approach reduces the variance in updates, enabling faster and more stable convergence.

D. The integration of Transfer Learning

The proposed framework combines RL algorithms with transfer learning to address the studied EVCSP. RL algorithms iteratively learn optimal EV-to-charger assignment policies by updating Q-values, while transfer learning enhances computational efficiency and ensures solution feasibility. This integration leverages RL's adaptability for dynamic policy learning and transfer learning's ability to utilize prior knowledge, improving both the efficiency and effectiveness of the learning process. Transfer learning addresses the following two key challenges in this integration:

Refinement of Learned Knowledge: The two-phase approach of transfer learning allows the framework to separate the initial exploration of the environment dynamics from the subsequent refinement of assignment policies. In the first phase, the algorithm focuses on learning patterns such as charger availability and demand conflicts without the computational burden of validating against the mathematical model. This foundational knowledge is then refined in the second phase, where the learned policies are validated and adjusted to ensure compliance with energy and operational constraints.

Reduction of Computational Costs: Validating assignment decisions against the mathematical model during every episode is computationally expensive. Transfer learning mitigates this by isolating the computationally intensive validation process to the second phase.

The integration of transfer learning follows a two-phase mechanism to enhance the learning process:

- **First Phase (Learning Environment Dynamics):** In the initial phase, the Q-table is initialized with zero values and trained without incorporating the mathematical model. This phase allows the algorithm to explore the environment and learn essential patterns, such as charger availability, demand conflicts, and the impact of assignment decisions. By excluding the mathematical model during this phase, the framework avoids unnecessary computational overhead while building foundational knowledge.
- **Second Phase (Validation with the Mathematical Model):** In the second phase, the Q-table trained during the first phase is reused. However, in this phase, each assignment decision is validated against the mathematical model to ensure compliance with energy constraints and other operational requirements. This validation refines the learned policy, resulting in higher solution quality and feasibility.

The integration of RL and transfer learning within the framework is detailed in Algorithm 2. This algorithm highlights how the RL training process is conducted over multiple episodes, incorporating transfer learning to separate the exploratory and validation phases. The boolean flag $using\_MM$ determines the phase of the transfer learning process, toggling between exploratory learning and validated refinement. This phased approach allows the framework to efficiently learn and refine assignment policies, ensuring a balance between computational efficiency and operational feasibility.

E. Mathematical Model for the Lower-Level Optimization

The lower level of the framework refines preliminary assignments generated by the upper-level RL algorithms by optimizing energy allocation while rigorously enforcing constraints on grid capacity and charger availability. This process finalizes the set of accepted demands, rejecting some of the assigned demands if they fail to satisfy the lower-level constraints.

Let A denote the set of demands that are potentially assigned to chargers in the upper level without violating individual time constraints. Specifically, a demand $j \in A$ can be assigned to a charger for its duration $[r_j, d_j)$ without interruptions or conflicts with other demands. Let C_j represent the index of the charger to which demand $j \in A$ is assigned in the upper-level solution.

To determine the final set of accepted demands and optimize energy allocation, the lower level employs an integer linear programming (ILP) model. This model evaluates which of the potential assignments from the upper level can be accepted while ensuring compliance with all operational constraints. The formulation involves two binary decision variables. The variable x_j determines whether demand $j \in A$ is accepted on its assigned charger C_j. Specifically, $x_j = 1$ indicates that the demand is accepted, while $x_j = 0$ indicates that it is not. The variable y_{jt} specifies whether demand $j \in A$ is being charged by charger C_j during time slot t. In this case, $y_{jt} = 1$ means the demand is being charged at time t, whereas $y_{jt} = 0$ means it is not.

The complete mathematical formulation of the lower-level problem is presented below. In this formulation, the objective function (Eq. 6) maximizes the number of accepted demands, reflecting the primary goal of the optimization. Constraint 7 ensures that the allocated charging time for each demand j satisfies its required duration. Constraint 8 ensures that the total energy consumption across all chargers at any given time remains within the grid's available capacity.

$$\max \sum\nolimits_{j \in A} x_j \tag{6}$$

$$\sum\nolimits_{t=r_j}^{d_j-1} y_{jt} = p_{C_j,j} \cdot x_j \quad \forall j \in A \tag{7}$$

$$\sum\nolimits_{j \in A, t \in [r_j, d_j)} w_{C_j,j} \cdot y_{jt} \leq W \quad \forall t \in T \tag{8}$$

IV. Upper Bound Calculation

To obtain a precise upper bound, the scheduling horizon is divided into intervals determined by vehicle arrival and departure times. A sorted set H is created, containing all unique arrival and departure times arranged in non-decreasing order. Each element of this set is represented by H_h, where h denotes its index, and the total number of elements is given by $|H|$. For any interval $[H_h, H_k]$, with $h, k \in \{1, \ldots, |H|\}$ and $h < k$, the subset J' is defined as the set of all demands j satisfying $r_j \geq H_h$ and $d_j \leq H_k$, where r_j and d_j represent the arrival and departure times of demand j, respectively.

Within each interval $[H_h, H_k]$, the maximum number of demands that can be accommodated is calculated by solving a knapsack problem. In this context, demands are treated as items, with their energy requirements (e_j) serving as their weights. The knapsack capacity corresponds to the energy available within the interval, computed as $W_G \cdot (H_k - H_h)$, where W_G is the grid capacity. Solving the knapsack problem for this interval provides nb, the upper bound on the number of demands that can be feasibly served within it.

Algorithm 2 Reinforcement Learning with Transfer Learning

Input: Q-table $Q(s, a)$,Number of episodes $n\_episodes$, boolean $using\_MM$, RL update function fn.
Output: Q-table $Q(s, a)$.
for each $episode = 0$ to $n\_episodes - 1$ **do**
  Initialize charger times $tc[i] = 0$ for all $i \in M$.
  Initialize state $s = (j = 0, k, b)$.
  Initialize energy left $W_t^{left} = W_G$ for all $t \in \mathcal{T}$.
  Initialize solution $sol_j = m$ for all demands j.
  for each demand $j = 0$ to $n - 1$ **do**
    Calculate k and b for demand j.
    Select action a using ϵ-greedy.
    Execute action a:
    if $a < m$ and $k = 1$ **then**
      Assign demand j to charger a.
      Update charger available time: $tc_a = d_j$.
      Update energy left: $W_t^{left} \leftarrow W_t^{left} - w_a$ for the selected processing times.
    end if
    $sol_j = a$ and $s \leftarrow s'$.
  end for
  if $using\_MM$ **then**
    Evaluate solution sol and return the number of accepted demands.
  else
    Count the conflict-free assigned demand.
  end if
  Update Q-values with RL function fn.
  Decay exploration rate ϵ.
end for
Return: Trained Q-table $Q(s, a)$.

The overall upper bound is determined by evaluating all possible intervals across the scheduling horizon. For each interval, the number of rejected demands is subtracted from the total number of demands, n, and the smallest value among these results is selected as the final upper bound. This refined method ensures that the upper bound accurately accounts for the problem's operational constraints while maintaining computational efficiency.

V. Computational Results

The computational experiments were conducted on a system featuring an Intel Xeon W-2295 processor (3.0 GHz base frequency), 128 GB of RAM, and the Windows 10 operating system. The proposed algorithms were implemented in C++ and interfaced with the IBM CPLEX Optimization Studio API to construct and solve the linear programming models.

The experimental dataset was developed following the methodology described in [15]. Computational experiments were conducted on scenarios grouped into three categories of problem instances, defined by the number of charging demands ($n \in \{10, 40, 50\}$), the number of chargers ($m \in \{15, 24, 27\}$), and grid capacities ($w_G \in \{50, 75, 100\}$). Each group combined specific values of n, m, and w_G to create consistent experimental configurations. Chargers were divided into three types based on their power outputs, with one-third operating at $w_1 = 11$ kW, one-third at $w_2 = 22$ kW, and the remaining at $w_3 = 43$ kW. For each configuration, 10 random instances were generated to ensure a diverse evaluation of the proposed methods.

TABLE I: Intervals for α Based on Charging Times (p_{1j})

$p_{1j} \in$	**[0.5, 1)**	**[1, 2)**	**[2, 3)**	**[3, 4)**	**[4, 5)**	**[5, 6]**
$\alpha \in$	$[0.1, 1]$	$[0.1, 0.9]$	$[0.1, 0.8]$	$[0.1, 0.7]$	$[0.1, 0.6]$	$[0.1, 0.5]$

Random instances were generated by sampling vehicle arrival times (r_j) uniformly over the interval $[0, 0.2n]$ hours. Energy requirements (e_j) were drawn from a range of $[5.5, 66]$ kWh, and charging times (p_{1j}) for type 1 chargers ($w_1 = 11$ kW) were computed as $p_{1j} = \frac{e_j}{11}$. Departure times (d_j) were calculated as $d_j = r_j + (1 + \alpha)p_{1j}$, where α was assigned based on the value of p_{1j}, as outlined in Table I.

To promote reproducibility, the generated problem instances and the developed codes are publicly available via a GitHub repository: https://github.com/abdennourAzerine/ijcn-2026-ev-charging-data. This ensures open access for the research community.

The results of the proposed solution techniques across various benchmark instances are summarized in Tables II, III, and IV. Each table lists the instance identifiers in the first column, followed by the calculated upper bounds. Subsequent columns detail the computational time (T) and objective values (z) for the Q-learning, Double Q-learning, SARSA, and Expected SARSA algorithms. For each reinforcement learning method, separate sub-columns report results for scenarios with transfer learning enabled (*TL: ON*) and disabled (*TL: OFF*). The last two columns indicate the results obtained from the simulated annealing (SA) algorithm proposed in the reference study of [15]. A summary row in each table provides the average computational time and the number of instances (out of 10) where each method achieved the best objective value. All methods were executed with a time limit of 1200 seconds or 20,000 episodes, whichever was reached first. If the specified limit was reached without full convergence, the best solutions identified during the optimization process were recorded.

In Table II, representing the smallest instance, all RL methods with transfer learning achieve optimal results, precisely matching the reported UB. Among the RL methods without transfer learning, only Q-learning maintains this level of optimality. Double Q-learning without transfer learning, while the most computationally efficient among the RL methods, delivers the worst solution quality, reaching the optimal value in only 4 instances. The SA developed from the baseline study matches the optimal result in 4 instances but yields inferior performance in all others. Although SA requires significantly less computational time, this comes at the expense of degraded solution quality.

Table III highlights the benefits of enabling transfer learning, particularly as the problem complexity increases. Among the RL methods, SARSA and Expected SARSA with transfer learning outperform the others, achieving the best objective value in 9 instances. These results demonstrate the effectiveness of transfer learning in improving solution quality for moderate-sized instances. In contrast, the SA, despite requiring the same computational time as the other methods, failed to obtain the best solution in any instances.

As instance sizes increase, the performance disparity among the reinforcement learning methods becomes more pronounced. According to Table IV, SARSA with transfer learning achieves the best objective values in 9 instances. However, the failure to reach the upper bounds for these larger instances confirms that optimality is not achieved. The Double Q-learning with disabled transfer learning, while the most computationally efficient among the reinforcement learning methods, delivers the worst solution quality, highlighting the trade-off between efficiency and accuracy. The results from the SA show that it achieves the best known solution in only one instance, and this is attained with a computational time comparable to that of the RL methods. Nevertheless, SA outperforms all RL methods when transfer learning is deactivated, indicating its competitive advantage in the absence of knowledge transfer mechanisms.

It is worth mentioning that the direct comparison with the results reported by Zaidi et al. [15] was not feasible due to the lack of access to their benchmark instances. Therefore, their SA-based method was re-implemented and evaluated on the benchmark introduced in this study, providing a representative metaheuristic baseline. The results show that SA performs comparably to RL methods without transfer learning on small- to medium-sized instances and begins to outperform them as problem size increases. This performance degradation highlights the difficulty RL methods face when trained from scratch in large and highly constrained environments. A key factor contributing to this challenge is the computational cost of repeatedly solving the exact mathematical model for energy allocation within each learning episode, which limits the number of episodes that can be executed within a fixed time budget. In contrast, when transfer learning is incorporated into the framework, the proposed method consistently outperforms SA across all instance sizes, including the most complex cases. This improvement stems from a staged training process in which the RL agent initially learns charger assignment policies using a lightweight heuristic, avoiding the overhead of solving the exact model. This allows for a substantially larger number of training episodes during the early learning phase. In the later phase, the exact optimization model is reintroduced, enabling the agent to refine its policy using accurate feasibility feedback. This phased learning strategy significantly enhances convergence and solution quality by balancing computational efficiency with rigorous constraint enforcement.

It is worth mentioning that all performance evaluations are statistically validated using multiple tests under a 0.05 significance level, confirming the reliability and robustness of the observed improvements. These results demonstrate that while pure metaheuristics like SA offer competitive baselines, the proposed framework, especially when enhanced with transfer learning, offers superior scalability, adaptability, and overall performance.

TABLE II: Results with $n = 10$ demands and $m = 15$ chargers

id	UB	Q-learning				Double Q-learning				SARSA				Expected SARSA				SA [15]	
		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF			
		T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z
1	10	918.53	**10**	795.37	**10**	1007.58	**10**	816.63	9	942.18	**10**	893.68	**10**	899	**10**	930.54	**10**	81.66	**10**
2	10	991.23	**10**	921.82	**10**	893.45	**10**	826.83	**10**	780.66	**10**	1052.60	**10**	1201.67	**10**	925.90	**10**	336.67	9
3	9	821.91	**9**	867.07	**9**	894.68	**9**	747.95	8	892.64	**9**	775	8	882.70	**9**	754.43	8	372.41	7
4	10	888	**10**	975.51	**10**	1200.01	**10**	843.68	**10**	913.97	**10**	941.94	**10**	728.63	**10**	893.59	**10**	167.8	9
5	9	1017.89	**9**	798.04	**9**	917.78	**9**	819.91	**9**	1020.43	**9**	908.03	**9**	948.63	**9**	915.98	**9**	385.2	**9**
6	10	1030.86	**10**	794.49	**10**	956.83	**10**	861.66	9	1097.71	**10**	881.28	**10**	1137.73	**10**	804.73	9	317.27	9
7	10	749.75	**10**	985.39	**10**	950.15	**10**	829.90	**10**	915.28	**10**	1168.20	**10**	948.75	**10**	904.61	**10**	286.53	9
8	10	758.70	**10**	755.84	**10**	827.30	**10**	620.83	9	672.21	**10**	779.69	**10**	819.06	**10**	704.19	**10**	66.51	**10**
9	10	834.77	**10**	829.46	**10**	833.92	**10**	455.21	8	878.53	**10**	849.74	**10**	829.65	**10**	891.62	**10**	86.08	**10**
10	10	988.42	**10**	1041.31	**10**	975.66	**10**	784.22	9	1139.60	**10**	881.80	**10**	906.55	**10**	1145.25	**10**	369.84	9
Statistics		900.01	**10**	876.43	**10**	945.74	**10**	830.08	4	925.32	**10**	928.55	9	930.24	**10**	913.96	8	246.99	4

TABLE III: Results with $n = 40$ demands and $m = 24$ chargers

id	UB	Q-learning				Double Q-learning				SARSA				Expected SARSA				SA [15]	
		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF			
		T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z
11	34	1200.07	**33**	1201.60	28	1200.61	32	1200.01	23	1200.01	**33**	1200.60	30	1200.04	**33**	1200.01	25	1200.26	31
12	31	1202.22	**30**	1200.96	26	1200	**30**	1200.28	22	1200.13	**30**	1200.17	28	1200.01	**30**	1200.03	24	1200.74	28
13	32	1200.09	**30**	1200.01	25	1200.06	28	1200.05	25	1200.01	29	1200.15	28	1200.12	29	1200.03	29	1200.78	28
14	32	1200.02	**30**	1200.33	25	1201.80	29	1200.84	23	1200.14	**30**	1200.01	27	1200.08	**30**	1200.08	24	1200.63	28
15	30	1200.37	**28**	1200.23	25	1200.15	27	1200.22	23	1200.09	**28**	1200.66	26	1201.22	**28**	1201.35	24	1200.9	25
16	32	1200.08	30	1200.09	26	1200.17	29	1200.03	22	1200.18	**31**	1200.14	25	1200.15	**31**	1200.01	23	1200.8	28
17	31	1200.23	28	1200.01	24	1201.06	28	1200.19	21	1200.79	**29**	1200.44	24	1200.13	**29**	1200.15	24	1200.09	25
18	33	1200.13	**32**	1200.15	27	1200.96	31	1200.17	24	1200.15	**32**	1200.08	26	1201.13	**32**	1200.17	26	1200.59	30
19	31	1200.02	**29**	1202.39	25	1200.08	**29**	1200.02	22	1200.11	**29**	1200.04	24	1200.17	**29**	1200.11	25	1200.66	28
20	34	1201	32	1200.27	28	1200.06	32	1200.53	26	1200.92	**33**	1200.29	27	1202.06	**33**	1200.23	27	1200.52	31
Statistics		1200.42	7	1200.6	0	1200.49	2	1200.23	0	1200.25	**9**	1200.26	0	1200.51	**9**	1200.22	0	1200.6	0

TABLE IV: Results with $n = 50$ demands and $m = 27$ chargers

id	UB	Q-learning				Double Q-learning				SARSA				Expected SARSA				SA [15]	
		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF		TL:ON		TL:OFF			
		T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z	T (s)	z
21	46	1200.07	41	1200.26	33	1200.03	38	787.25	28	1201.67	**42**	1200.24	35	1201.66	41	1200.13	33	1200.47	41
22	45	1200.15	42	1200.01	34	1200.24	40	735.31	34	1201.85	**43**	1200.04	38	1202.73	**43**	1200.21	35	1200.07	41
23	44	1200.38	**41**	1200.18	30	1201.71	35	1200.17	29	1200.79	**41**	1200.29	33	1200.37	**41**	1200.16	30	1200.86	39
24	47	1202.65	43	1200.16	34	1200.30	40	1200.25	33	1201.41	**45**	1200.27	35	1200.28	44	1200.20	31	1200.74	44
25	48	1200.26	**46**	1200.32	35	1200.38	41	1200.19	36	1201.13	**46**	777.47	34	1200.74	**46**	1200.01	34	1200.31	45
26	44	1200.32	40	1201.76	31	1202.49	38	1200.24	32	1200.80	**41**	1200.12	31	1200.46	40	1200.20	29	1200.93	39
27	50	1200.27	**47**	1200.06	36	1200.51	43	682.87	33	1200.04	**47**	1200.16	38	1202.09	**47**	1200.19	34	1200.82	46
28	43	1200.59	39	1200.34	30	1200.64	35	1200.33	34	1200.34	**40**	1200.33	33	1200.66	39	1200.35	29	1200.64	38
29	47	1201.20	41	1200.41	36	1201.51	39	1200.34	29	1202.08	**42**	1200.23	34	1201.57	**42**	1200.17	32	1200.19	**42**
30	50	1200.09	48	1200.16	37	1200.24	44	1200.29	36	1200.08	48	1200.12	40	1200.06	**49**	1200.27	36	1200.31	48
Statistics		1200.6	3	1200.37	0	1200.8	0	1060.72	0	1201.02	**9**	1157.93	0	1201.06	6	1200.19	0	12000.53	1

VI. Conclusion

This study addresses the Electric Vehicle Charging Scheduling Problem (EVCSP) for maximizing satisfied charging demands under real-world constraints. A novel bi-level optimization framework was developed, dividing the problem into an upper-level demand-to-charger assignment and a lower-level energy allocation optimization. The upper-level problem was solved using reinforcement learning (RL) methods comprising Q-learning, Double Q-learning, SARSA, and Expected SARSA, while incorporating transfer learning to enhance computational efficiency and solution quality. The lower-level problem utilized an exact mathematical model to ensure optimal energy allocation for the assignments generated at the upper level, combining adaptive learning with precision optimization to effectively address the problem's complexity.

Extensive computational experiments validate the proposed framework. For smaller instances, all RL methods with transfer learning achieved optimal solutions, precisely matching the reported upper bounds. For larger instances, SARSA with transfer learning consistently delivered the best objective values, demonstrating its ability to handle high-dimensional and challenging scenarios.

To provide a meaningful comparison with established metaheuristic approaches, the simulated annealing algorithm—previously proposed in the literature as an effective solution for the EVCSP—was re-implemented and evaluated on the same benchmark set. The results indicate that while simulated annealing outperforms reinforcement learning methods operating without transfer learning in several larger instances, it is consistently outperformed by SARSA when transfer learning is enabled. This demonstrates the critical role of transfer learning in enhancing the scalability and effectiveness of learning-based approaches in constrained optimization problems. Overall, the results confirm the strength of the proposed bi-level framework, where reinforcement learning, when combined with transfer learning and exact optimization, achieves superior performance relative to classical metaheuristics.

Future research could extend this framework by incorporating stochastic and dynamic demand patterns or exploring advanced reinforcement learning approaches, such as deep reinforcement learning, to further enhance adaptability and scalability. Additionally, integrating real-time optimization and testing on large-scale EV charging networks would expand the framework's applicability to dynamic and practical scenarios.

Acknowledgment

This work is supported by the CyMoVe project, funded under the AMI CMA France 2030 program, operated by the French *Agence Nationale de la Recherche* and *Groupe Caisse des Dépôts*.

References

[1] J. Fan, X. Meng, J. Tian, C. Xing, C. Wang, and J. Wood, "A review of transportation carbon emissions research using bibliometric analyses," *Journal of Traffic and Transportation Engineering (English Edition)*, 2023.

[2] Eurostat, "Final energy consumption in transport - detailed statistics," Sep. 2024, accessed on 4 December 2024. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php

[3] EU-Parliament, "Fit for 55: zero co2 emissions for new cars and vans in 2035," February 2023, accessed on 4 December 2024. [Online]. Available: https://www.europarl.europa.eu/news/en/press-room/20230210IPR74715/fit-for-55-zero-co2-emissions-for-new-cars-and-vans-in-2035

[4] IEA, "Global ev outlook 2024: Trends in electric cars," 2024, accessed on 4 December 2024. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2024/trends-in-electric-cars

[5] ACEA, "Automotive insights – charging ahead: accelerating the rollout of eu electric vehicle charging infrastructure," 2024, accessed on 4 December 2024. [Online]. Available: https://www.acea.auto/publication/automotive-insights-charging-ahead-accelerating-the-rollout-of-eu-electric-vehicle-charging-infrastructure/

[6] G. Maguire, "Slow charge point rollout risks stalling us ev sales momentum," 2024, accessed on 4 December 2024. [Online]. Available: https://www.reuters.com/markets/commodities/slow-charge-point-rollout-risks-stalling-us-ev-sales-momentum-maguire-2024-10-09

[7] P. Barman, L. Dutta, S. Bordoloi, A. Kalita, P. Buragohain, S. Bharali, and B. Azzopardi, "Renewable energy integration with electric vehicle technology: A review of the existing smart charging approaches," *Renewable and Sustainable Energy Reviews*, vol. 183, p. 113518, 2023.

[8] S. Shao, H. Sartipizadeh, and A. Gupta, "Scheduling ev charging having demand with different reliability constraints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 11 018–11 029, 2023.

[9] S. Velamuri, M. P. Kantipudi, R. Sitharthan, D. Kanakadhurga, N. Prabaharan, and A. Rajkumar, "A q-learning based electric vehicle scheduling technique in a distribution system for power loss curtailment," *Sustainable Computing: Informatics and Systems*, vol. 36, p. 100798, 2022.

[10] I. Zaidi, A. Oulamara, L. Idoumghar, and M. Basset, "Minimizing grid capacity in preemptive electric vehicle charging orchestration: Complexity, exact and heuristic approaches," *European Journal of Operational Research*, vol. 312, no. 1, pp. 22–37, 2024.

[11] N.-Q. Nguyen, F. Yalaoui, L. Amodeo, H. Chehade, and P. Toggenburger, "Reactive rescheduling method for electric vehicles charging in dedicated residential zone parking," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–6.

[12] T. Panayiotou, M. Mavrovouniotis, and G. Ellinas, "On the fair-efficient charging scheduling of electric vehicles in parking structures," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1627–1634.

[13] Y. Luo, T. Zhu, S. Wan, S. Zhang, and K. Li, "Optimal charging scheduling for large-scale ev (electric vehicle) deployment based on the interaction of the smart-grid and intelligent-transport systems," *Energy*, vol. 97, pp. 359–368, 2016.

[14] S. Shao, F. Harirchi, D. Dave, and A. Gupta, "Preemptive scheduling of ev charging for providing demand response services," *Sustainable Energy, Grids and Networks*, vol. 33, p. 100986, 2023.

[15] I. Zaidi, A. Oulamara, L. Idoumghar, and M. Basset, "Maximizing the number of satisfied charging demands in electric vehicle charging scheduling problem," in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 2022, pp. 89–102.

[16] L. Asadzadeh, "A local search genetic algorithm for the job shop scheduling problem with intelligent agents," *Computers & Industrial Engineering*, vol. 85, pp. 376–383, 2015.

[17] A. Azerine, M. Golabi, A. Oulamara, and L. Idoumghar, "Enhancing electric vehicle charging schedules: A surrogate-assisted approach," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2024, pp. 183–186.

[18] A. Azerine, A. Oulamara, M. Basset, and L. Idoumghar, "Improved methods for solving the electric vehicle charging scheduling problem to maximize the delive," in *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2024, pp. 1–8.

[19] Z. Zhao, C. K. Lee, X. Yan, and H. Wang, "Reinforcement learning for electric vehicle charging scheduling: A systematic review," *Transportation Research Part E: Logistics and Transportation Review*, vol. 190, p. 103698, 2024.

[20] S. Wang, S. Bi, and Y. A. Zhang, "Reinforcement learning for real-time pricing and scheduling control in ev charging stations," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 849–859, 2019.

[21] N. Mhaisen, N. Fetais, and A. Massoud, "Real-time scheduling for electric vehicles charging/discharging using reinforcement learning," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 1–6.

[22] C. Zhang, Y. Liu, F. Wu, B. Tang, and W. Fan, "Effective charging planning based on deep reinforcement learning for electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 542–554, 2020.

[23] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[24] H. Hasselt, "Double q-learning," *Advances in neural information processing systems*, vol. 23, 2010.

[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.