Bi-objective Optimization of Electric Vehicle Charging Scheduling with Heterogeneous Chargers Under Time-Varying Electricity Prices

Imene Zaidi[1(✉)], Latifa Belhocine[1,2], Baptiste Buchi[1], and Hajar Nouinou[1]

[1] CESI LINEACT, UR 7527, Nancy, France
{izaidi,bbuchi,hnouinou}@cesi.fr, latifa.belhocine@mines-albi.fr
[2] Centre de Génie Industriel, IMT Mines Albi, Albi, France

Abstract. This paper addresses the optimization of charging schedules in EV charging stations with heterogeneous chargers under time-varying electricity prices. The objective is to decide which demands to accept and how to schedule them to maximize the number of demands served while minimizing electricity costs. We formulate the problem as a bi-objective model and compare two approaches: an exact ε-constraint method and a hybrid NSGA-II enhanced with repair mechanisms and refinement procedures. Computational results on 70 benchmark instances with up to 400 EVs and four price scenarios show that the hybrid NSGA-II produces well-distributed Pareto fronts and achieves competitive performance on small instances, while significantly outperforming the exact method on large instances, with improvements of up to +59.89% in normalized hypervolume and an average gain of +40.8% for instances with 300–400 EVs. In addition, it reduces computation time by up to 77%, with an average reduction of 65% for instances ranging from 50 to 400 EVs.

Keywords: Electric Vehicle · Charging Scheduling · Multi-Objective Optimization · Electricity Cost Optimization

1 Introduction

Although electric vehicles (EVs) emerged alongside the first automobiles, their adoption declined during the twentieth century before reemerging in the 2010s as part of the clean energy transition. According to the International Energy Agency [7], global EV sales reached 17 million units in 2024, representing over 20% of total car sales. Progress in EV battery technology is shortening charging times and could make them comparable to refuelling, provided that charging infrastructure expands quickly to support the growing adoption of EVs. In 2024, the global number of public charging points has already exceeded 5 million [7]. However, when many EVs charge simultaneously, particularly during

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16769, pp. 568–586, 2027.
https://doi.org/10.1007/978-3-032-30494-0_36

peak hours, local transformers and distribution lines risk overload. Upgrading grid infrastructure to accommodate these higher loads is expensive, complex, and time-consuming [18]. Urban areas face additional constraints, as retrofitting old buildings or underground parking facilities to accommodate charging equipment is complicated [2], and suitable parking spots are often scarce. Given these infrastructure limitations and the high cost of capacity expansion, operators must prioritize the intelligent use of existing resources: chargers, parking spots, and total power capacity. While power capacity allocation has received considerable attention in the literature, parking spot availability is often overlooked despite being equally constraining [19]. Safety regulations stipulate that an EV cannot be unplugged by anyone other than its owner once connected, meaning a vehicle may occupy both a charger and a parking spot for extended periods, even after charging is complete. This behavior is widespread in workplace or residential settings, where users leave their vehicles parked throughout the day [11]. Consequently, deploying too few chargers creates queues and user dissatisfaction, while deploying too many leads to low utilization rates and financial losses. Operators must therefore develop intelligent scheduling strategies to maximize the use of limited resources while preventing grid overload.

In this paper, parking spaces, chargers, and power grid capacity are treated as constrained resources. We consider a smart charging setting in which EV owners submit their charging demands in advance, and the station operator aims to maximize customer satisfaction by serving as many charging demands as possible. Studying the offline version of the problem is important as it provides a reference for evaluating online scheduling strategies and is also relevant in practical settings such as reservation-based charging infrastructures in airports and large cities. Under time-varying electricity prices, accepting more demands may increase electricity costs during peak-price periods, while prioritizing low-price periods may reduce the number of satisfied demands. Therefore, we formulate a bi-objective optimization problem that maximizes demand satisfaction while minimizing the total electricity charging cost of the accepted demands. We analyze how pricing profiles purchased from the grid utility influence charging scheduling decisions. The charging station resells electricity to EV users at a fixed price per kWh. The objective is to capture the trade-off between electricity procurement cost and demand satisfaction. By providing a Pareto front, our approach supports decision-making by allowing operators to determine how many charging requests to accept or reject under different pricing conditions, and to select a pricing strategy aligned with their desired level of service. This decouples the optimization model from any specific price-cost relationship, ensuring the two objectives remain structurally conflicting regardless of whether service prices exceed operational costs. Moreover, the introduction of heterogeneous chargers and the need to approximate a Pareto front significantly increase the computational complexity. To address this challenge, we propose and compare both an ε-constraint method and a Non-dominated Sorting Genetic Algorithm II (NSGA-II) enhanced with repair mechanisms and Pareto-front refinement to efficiently approximate the Pareto-optimal front.

The paper is organized as follows. Section 2 reviews the literature and positions our contributions. Section 3 describes the problem, while Sect. 4 presents the mathematical formulation and the ε-constraint method. Section 5 introduces the proposed NSGA-II. Section 6 reports the computational results. Finally, Sect. 7 concludes the paper.

2 Related Work

The EV Charging Scheduling Problem (EVCSP) determines when and where EVs should be charged under technical and operational constraints, in order to optimize given objectives. EVCSP formulations depend on optimization objectives that reflect stakeholder priorities (e.g., charging station or grid operators), the type of infrastructure (home, public, or private), and constraints related to EVs, charging facilities, and the power grid. Our analysis focuses on public and private charging stations, such as residential parking areas, shopping malls, campuses, and hospitals, while acknowledging that cross-study comparisons remain challenging due to heterogeneous operational contexts. In this context, existing research typically focuses on maximizing satisfied charging demand [22,24,25], reducing grid capacity usage [25], or minimize charging costs for users while still satisfying their demand [21]. In [13], a multi-objective formulation is considered, aiming to maximize both operator profit and customer satisfaction.

Below, we summarize the main constraints commonly encountered in EVCSPs, starting with EV arrival and departure times, which define the permissible charging window. An EV can only be charged within its plug-in period, i.e., from arrival to departure time [13,21,22,25]. The energy demand of each vehicle is typically defined either as a fixed quantity of energy (in kWh) to be delivered within the plug-in period [13,22,24,25], or as a flexible desired state-of-charge (SoC) interval [23]. In [16,17,22,24,25], the energy needs of each vehicle have to be fulfilled before departure. The number of available chargers and their technical characteristics, such as charging power rates, directly constrain the number of vehicles that can be charged simultaneously. While many studies assume homogeneous chargers [16,21,24], where all installed chargers have identical characteristics, other works consider non-identical chargers [13,22,23,25]. In these settings, particularly when the number of chargers is limited [16,21,22,24,25], charging schedules must prevent overlapping assignments on the same charger. Computational results reveal that deploying heterogeneous chargers (with varying output powers) is more advantageous in reducing the overall grid capacity requirement [25]. Chargers can follow constant power models [16,24,25], where EVs are charged at a fixed rate, or variable power models [12,21], which allow the charging rate to vary continuously. Preemptive scheduling, which allows a charging session to be interrupted and resumed at a later time, is considered in several studies [21–25]. In large EV charging stations, grid capacity constraints limit the simultaneous use of chargers due to the transformer's maximum power capacity, which caps the total power that can be drawn at any time [12,17,21,22,25]. Electricity prices are often represented using Time-of-Use tariffs, where predefined

time intervals reflect variations in grid demand and encourage charging during low-cost periods [17].

NSGA-II is one of the most widely used methods for multi-objective optimization in the EVCSP [5]. It consistently produces competitive Pareto fronts, closely approximating optimal solutions and outperforming several alternative algorithms [10,13].

Although multi-objective optimization is widely used in the EVCSP literature, most studies adopt either a power grid or a user-centric perspective, rather than that of charging station operators [5,13], particularly when addressing electricity cost minimization under time-varying prices. However, charging stations also purchase electricity from utilities and must balance operational costs with service levels. This work addresses this gap by integrating bi-objective optimization into a realistic charging station framework. Based on settings similar to those in [24,25], we extend the EVCSP model to account for heterogeneous chargers and charging efficiency, and introduce a bi-objective formulation that captures the marginal electricity cost of each additional accepted demand under time-varying prices. This formulation provides a decision-support tool for demand acceptance and pricing. Since real-world stations typically include chargers with different power ratings, this heterogeneity increases scheduling complexity by requiring joint assignment and timing decisions. To address this, we adopt a type-based formulation that reduces computational complexity. We further propose two solution approaches: an ε-constraint method and a NSGA-II algorithm enhanced with repair mechanisms and Pareto-front refinement. Finally, both methods are evaluated on instances with 10–400 EVs under four pricing scenarios.

3 Problem Description

We address the EVCSP in a charging station with m parking spaces, each equipped with a charger delivering constant power. Chargers are classified into k types according to their power rate, denoted by the set $K = \{1, \ldots, k\}$, where type $i \in K$ delivers power $w_i \in \mathbb{R}_{>0}$ (in kW), with $m_i \in \mathbb{N}$ chargers of type i available. Thus, $\sum_{i=1}^{k} m_i = m$. The station operates under a global power constraint $w_G \in \mathbb{R}_{>0}$ (in kW): at any time, the total power delivered by all active chargers cannot exceed w_G. We consider the realistic case where the total power that could be delivered if all chargers were active exceeds the grid capacity, i.e., $\sum_{i \in K} w_i \cdot m_i > w_G$. The scheduling horizon $H = \{1, \ldots, T\}$ is discretized into time slots of duration τ. The station operator provides charging services to customers who submit charging demands before arrival. Let $J = \{1, \ldots, n\}$ denote the set of charging demands. Each demand $j \in J$ is characterized by an arrival time $r_j \in \mathbb{N}$, a departure time $d_j \in \mathbb{N}$ (with $r_j < d_j$), and an energy requirement e_j (in kWh). Without loss of generality, we assume that arrival and departure times are expressed as integer time slot indices in H. The parking time slots for demand j are denoted by $H_j = \{r_j, \ldots, d_j - 1\}$, representing the discrete time intervals during which the EV is present at the station. Each demand can be

either accepted or rejected. An accepted demand is assigned to a charger and occupies it throughout the interval H_j, with energy demand e_j fully satisfied.

Charging is *preemptive*: it can be interrupted and resumed during the EV's plug-in time H_j. However, the EV remains parked and occupies the assigned charger throughout this interval. Let $\eta \in (0, 1]$ denote the charging efficiency of the station, defined as the ratio between the useful energy delivered to the EV battery and the electrical energy drawn from the grid. The charging time required to fulfill demand e_j using charger type i is $p_{ij} = \left\lceil \frac{e_j}{\eta w_i \cdot \tau} \right\rceil$, where p_{ij} is the number of required time slots.

The charging station purchases electricity from the power grid according to a time-varying tariff profile, where ep_t (€/kWh) denotes the unit price paid by the charging station at time slot $t \in H$. EV users are charged a fixed price per kWh by the station operator. Consequently, reducing electricity procurement costs increases the charging station's operational profit. The model schedules all charging demands to optimize two conflicting objectives: (i) Maximize the number of accepted charging demands to maximize user satisfaction and serve as many customers as possible, and (ii) Minimize the total cost of electricity paid by the charging station. Since these objectives are conflicting, we aim to approximate the Pareto front, i.e., the set of non-dominated solutions where no objective can be improved without degrading another. This enables operators to decide how many demands to accept or reject under a given pricing policy.

4 Mathematical Model Formulation

The original problem was formulated as a single-objective mathematical model in [22]. In the proposed bi-objective model, charger assignments are relaxed by considering chargers of the same type as identical, assigning EVs to charger types instead of specific chargers. To ensure parking feasibility, it is sufficient to guarantee that, at each time slot, the number of simultaneously used chargers of type i does not exceed the available number m_i. This relaxation reduces both the number of variables and the number of constraints, significantly lowering computational complexity. We introduce two sets of binary decision variables: $X = (x_{ij})$, where $x_{ij} = 1$ if charging demand j is assigned to charger type i (and 0 otherwise); and $Y = (y_{jt})$, where $y_{jt} = 1$ if demand j is charged during time slot t (and 0 otherwise). The bi-objective model then optimizes the following two objective functions:

$$f_1(X) = \sum_{j=1}^{n} \sum_{i=1}^{k} x_{ij}, \qquad f_2(X, Y) = \sum_{j=1}^{n} \sum_{t=r_j}^{d_j - 1} y_{jt}\, ep_t \sum_{i=1}^{k} x_{ij} \tau w_i. \tag{1}$$

Here, f_1 denotes the number of accepted charging demands, while $f_2(X, Y)$ represents the total electricity cost (€) paid by the charging station. To ensure the feasibility of any solution, the following constraints must hold:

$$\sum_{i=1}^{k} x_{ij} \leq 1, \qquad \forall j \in J, \tag{2}$$

$$\sum_{j=1}^{n} a_{jt} x_{ij} \leq m_i, \quad \forall i \in K, \ \forall t \in H, \tag{3}$$

$$\sum_{t=r_j}^{d_j-1} y_{jt} = \sum_{i=1}^{k} p_{ij} x_{ij}, \qquad \forall j \in J, \tag{4}$$

$$\sum_{j=1}^{n} \sum_{i=1}^{k} w_i x_{ij} y_{jt} \leq w_G, \quad \forall t \in H, \tag{5}$$

Each EV can be assigned to at most one charger type (constraints 2). Once assigned, an EV j continuously occupies a charger from its arrival r_j to its departure d_j, so the number of EVs using chargers of a given type at any time cannot exceed the available units (constraints 3). This is modeled through the parameters a_{jt}, equal to 1 if $t \in H_j$ and 0 otherwise, indicating whether EV j is present at time t. Each accepted demand must receive exactly its required energy e_j within its plug-in window (constraints 4). Finally, the total power consumption at any time must remain below the grid capacity w_G (constraints 5). Charging is allowed only within the EV's presence window, i.e., $y_{jt} = 0$ for all $t \notin H_j$. Given that the product of two binary variables is itself binary, the linearization is performed by introducing additional binary variables $Z = (z_{ijt})$ to represent the conjunctions $(x_{ij} y_{jt})$. The variable z_{ijt} equals 1 if and only if both $x_{ij} = 1$ and $y_{jt} = 1$. Accordingly, the second objective can be rewritten as: $f_2(Z) = \sum_{j=1}^{n} \sum_{t=r_j}^{d_j-1} \sum_{i=1}^{k} e p_t z_{ijt} \tau w_i$ and constraints (5) are replaced by the following linear constraints: $z_{ijt} \geq y_{jt} + x_{ij} - 1$, $z_{ijt} \leq y_{jt}$, and $z_{ijt} \leq x_{ij}$, $\forall i \in K$, $j \in J$, $t \in H$, together with $\sum_{i=1}^{k} \sum_{j=1}^{n} w_i z_{ijt} \leq w_G \ \forall t \in H$.

4.1 ε-Constraint Method

A Pareto front for the bi-objective problem can be computed using the ε-constraint method [6]. We first solve the single-objective problem max f_1 (ignoring f_2) to obtain $f_1^{\max}$, the maximum number of satisfied demands. Then, at each iteration ℓ, the mathematical model is solved by minimizing f_2 while adding the constraint $f_1(x) \geq \varepsilon^{(\ell)}$. The parameter $\varepsilon^{(\ell)}$ is updated at each iteration as: $\varepsilon^{(\ell)} = f_1^{\max} - (\ell - 1), \quad \ell = 1, \ldots, f_1^{\max} + 1$. Each iteration yields a Pareto-optimal solution. Since f_1 is integer-valued, a unit decrement is sufficient to enumerate all distinct Pareto-optimal points.

5 NSGA-II with Hybrid Repair and Pareto Refinement

The design of the evolutionary algorithm is based on the NSGA-II framework proposed by [3]. Classical genetic operators, namely crossover and mutation can produce infeasible solutions. Therefore, repair procedures are applied selectively: crossover violations are handled by a lightweight corrective procedure, since crossover is applied at higher frequency; a heavier ILP-based repair procedure is reserved for mutation. In addition, a periodic refinement phase is applied to the current Pareto front to recover rejected EVs and improve solution quality beyond what the genetic operators alone can achieve.

5.1 Solution Representation

We propose a type-based representation for the EVCSP, where a solution is represented as the set S of pairs $S = \{(\pi_j, \sigma_j) \mid j \in J\}$, where $\pi_j \in \{-1, 1, \ldots, k\}$ indicates the charger type assigned to EV j ($\pi_j = i$ if assigned to charger type i, and $\pi_j = -1$ if the EV is rejected). The vector $\sigma_j = (\sigma_{j,t})_{t \in H_j} \in \{0,1\}^{|H_j|}$ specifies the charging schedule of EV j over its parking time slots $H_j = \{r_j, \ldots, d_j - 1\}$, where $\sigma_{j,t} = 1$ if EV j is charging at time t, and 0 otherwise. This new representation offers a compact encoding by associating EVs with charger types rather than individual chargers, thereby reducing the search space. Since all chargers of type i are identical, any assignment remains feasible as long as the number of EVs charging on type i does not exceed m_i. w_{π_j} denotes the power rate of the charger type assigned to EV j. Let $J' = \{j \in J : \pi_j \neq -1\}$ be the set of accepted charging demands. Feasibility is assessed via aggregate measures: $U_{it} = |\{j \in J' : \pi_j = i \text{ and } t \in H_j\}|$ is the number of EVs assigned to type i that are present at time t, and $w_{Gt} = \sum_{\substack{j \in J' \\ t \in H_j}} w_{\pi_j} \sigma_{j,t}$ is the total power consumption at time t. The solution S is feasible if and only if:

$$U_{it} \leq m_i, \qquad \forall\, i \in K,\ \forall\, t \in H, \tag{6}$$

$$w_{Gt} \leq w_G, \qquad \forall\, t \in H, \tag{7}$$

$$\sum_{t \in H_j} \sigma_{j,t} = p_{\pi_j j}, \qquad \forall\, j \in J'. \tag{8}$$

Algorithm 1: ILP-NSGA-II

Input: N, G, α_c, α_m, R_{lp}, $I^{\text{lp}}_{\max}$, g_{ref}, T_{lp}, g_{start}
Output: Pareto front $\mathcal{F}_1$

Initialize, evaluate, and rank P_0;
for $g = 1$ **to** G **do**
  $Q_g \leftarrow \emptyset$;
  while $|Q_g| < N$ **do**
    Select p_1, p_2 by binary tournament from P_{g-1};
    if $\mathcal{U}(0,1) < \alpha_c$ **then**
      $(c_1, c_2) \leftarrow$TwoPointCrossover(p_1, p_2);
      foreach $c \in \{c_1, c_2\}$ *infeasible* **do** $c \leftarrow$WorstEVRejectionRepair(c)
    else $(c_1, c_2) \leftarrow (p_1, p_2)$;
    foreach $c \in \{c_1, c_2\}$ **do**
      if $\mathcal{U}(0,1) < \alpha_m$ **then** $c \leftarrow$Mutate(c)
      if c *infeasible* **then** $c \leftarrow$IterativeILPRepair$(c, I^{\text{lp}}_{\max}, R_{\text{lp}})$
      if c *infeasible* **then** $c \leftarrow$WorstEVRejectionRepair(c)
      Evaluate and add c to Q_g;
  $P_g \leftarrow$EnvironmentalSelection$(P_{g-1} \cup Q_g, N)$;
  if $g \geq g_{\text{start}}$ ***and*** $(g - g_{\text{start}}) \bmod g_{\text{ref}} = 0$ **then** RefineParetoFront(P_g, T_{lp})
if *refinement not applied at* $g = G$ **then** RefineParetoFront(P_G, T_{lp})
return $\mathcal{F}_1(P_G)$;

Since the EVCSP has conflicting objectives, we approximate the *Pareto front* of non-dominated solutions. A solution S_1 dominates S_2 if $f_1(S_1) \geq f_1(S_2)$ and $f_2(S_1) \leq f_2(S_2)$ with at least one strict inequality. We maintain a population $\mathcal{P}$ ranked by *fast non-dominated sorting* and diversified via *crowding distance* [3].

5.2 Framework of NSGA-II

Algorithm 1 presents the proposed NSGA-II, which evolves a population of N individuals over G generations. The initial population is generated using a constructive heuristic (Sect. 5.3). At each generation, offspring are produced via binary tournament selection and genetic operators (Sect. 5.4). Infeasible schedules are restored by selective repair procedures (Sect. 5.5), and a periodic refinement phase improves the current Pareto front (Sect. 5.6). Population diversity is maintained through fast non-dominated sorting and crowding distance [3]. All stochastic decisions are sampled from $\mathcal{U}(0, 1)$.

5.3 Initial Population Generation.

Before describing the initial population generation process, we first present the heuristic used to construct a single feasible solution.

Constructive Solution Heuristic. To construct a solution S, the heuristic begins by marking all EVs as rejected, i.e., $\pi_j = -1$ and $\boldsymbol{\sigma}_j = \mathbf{0}$ for all $j \in J$. A candidate set $J' \subseteq J$ of EVs to be considered for scheduling is then built according to one of four configurations (EVs in $J \setminus J'$ remain rejected): (1) all EVs in J, ordered by non-decreasing r_j; (2) all EVs in J, ordered by non-decreasing r_j, with ties broken by non-decreasing d_j; (3) all EVs in J in uniformly random order; and (4) a random subset of J in uniformly random order. Configurations (1)–(3) generally lead to solutions that accept a near-maximal number of EVs, thereby concentrating the initial population in the high-f_1 region of the Pareto front. Configuration (4) is therefore introduced to ensure that the initial Pareto front contains more than a single solution. Next, each EV $j \in J'$ is then scheduled using the heuristic described below, with charger types tried in increasing order of power rate. In configuration (3), this sequence is additionally shuffled uniformly at random before processing the EVs. If no charger type can successfully schedule the EV, then the EV remains rejected.

EV Scheduling Heuristic. Given a partial solution S, a charging demand j, a charger type i, a Boolean parameter `respectGridCapacity`, and a parameter `strategy`, the procedure attempts to assign EV j to charger type i and schedule its charging slots. It returns `true` if the assignment succeeds and `false` otherwise. The parameter `respectGridCapacity` determines whether the grid capacity constraint is respected (Eq. 7). If it is set to `false`, the procedure may generate intermediate solutions that violate the grid capacity constraint; such

solutions are later repaired by a correction procedure. The parameter `strategy` determines the time slot selection criterion for charging assignment.
Step 1: Parking Feasibility. Every slot $t \in H_j$ must have an available charger of type i, that is, $U_{it} < m_i$. Otherwise, the procedure returns `false`.
Step 2: Candidate-Slot Construction. The set of candidate slots $\mathcal{T}_{\text{cand}} \subseteq H_j$ is constructed as follows. If `respectGridCapacity = true`, only slots satisfying $w_{Gt} + w_i \leq w_G$ are considered. Otherwise, all slots in H_j are included in $\mathcal{T}_{\text{cand}}$. If $|\mathcal{T}_{\text{cand}}| < p_{ij}$, the procedure returns `false`.
Step 3: Slot Ordering. The candidate set $\mathcal{T}_{\text{cand}}$ is sorted according to the chosen `strategy`: (i) *Electricity Price-Based Ordering (EPO)*: slots are ordered by non-decreasing electricity price ep_t. (ii) *Grid Load-Aware Ordering (GLO)*: slots are ordered by non-decreasing grid load w_{Gt}, with ties broken by ep_t.
Step 4: Assignment. The EV is assigned to charger type i, i.e., $\pi_j \leftarrow i$. The first p_{ij} slots in the ordered set $\mathcal{T}_{\text{cand}}$ are selected. The indicators and counters are updated accordingly, namely $\sigma_{j,t}$ for the selected slots, as well as U_{it} and w_{Gt}.

ILP-Based Cost Minimization. The EV scheduling heuristic selects charging time slots greedily according to a predefined ordering criterion, which does not guarantee optimality with respect to electricity cost. Therefore, once charger type assignments $\{\pi_j\}_{j \in J'}$ are fixed, the remaining decision – selecting which time slots to charge each EV – can be formulated as an ILP and solved to optimality. Let S be the solution and $J' \subseteq J$ the set of accepted EVs with assigned charger types $\pi_j \neq -1$ for all $j \in J'$. Let $w_j = w_{\pi_j}$ denote the charging power rate assigned to EV j. With fixed charger assignments, we formulate a simplified ILP model to minimize total electricity cost by deciding only charging slots via binary variables y_{jt}: $\min f_2(Y) = \sum_{j \in J'} \sum_{t=r_j}^{d_j-1} y_{jt}\, ep_t\, \tau\, w_j$ subject to $\sum_{t=r_j}^{d_j-1} y_{jt} = p_j$, $\forall j \in J'$, $\sum_{j \in J'} w_j\, y_{jt} \leq w_G$, $\forall t \in H$, and $y_{jt} = 0$, $\forall j \in J'$, $t \notin H_j$. If the model is infeasible, the original solution S is returned.

Population Generation. The initial population is generated by running the constructive heuristic with varying parameter combinations to produce diverse individuals, all with `respectGridCapacity = true`. Two deterministic individuals are always generated first: configuration (2) with *GLO* followed by ILP-Based Cost Minimization (Sect. 5.3), and configuration (1) with *EPO* without ILP-Based Cost Minimization. The remaining $N - 2$ individuals are generated stochastically by sampling parameter combinations in fixed proportions 25% / 50% / 25%: configuration (3) with *GLO* followed by ILP-Based Cost Minimization; configuration (4) with *GLO* followed by ILP-Based Cost Minimization; configuration (4) with *EPO* without ILP-Based Cost Minimization. These combinations are shuffled before generation to avoid ordering bias in the initial population.

5.4 Generating Offspring

Two-Point Crossover. Given two parent solutions p_1 and p_2, two cut points $l_1 \leq l_2$ are drawn uniformly from $\{1, \ldots, n\}$. Offspring c_1 inherits the schedule (π_j, σ_j) of each EV j from p_1 for EVs outside the segment $j \notin [l_1, l_2]$ and from p_2 for EVs within; c_2 inherits complementarily. The Worst EV Rejection Repair procedure (Sect. 5.5) is applied to each infeasible offspring to restore feasibility.

Mutation. Mutation is applied with probability α_m. One operator is selected according to the probabilities α_{orig}, α_{cons}, and α_{swap}. Original mutation: a random EV is selected; if it is scheduled, it is removed and rescheduled on another charger type; otherwise, a scheduling attempt is performed. Destruct–Construct: $Q \sim \mathcal{U}(1, Q_{\text{des}})$ accepted EVs are removed, then up to $L \sim \mathcal{U}(1, L_{\text{cons}})$ rejected EVs are reinserted using the GLO scheduling strategy. Swap charger types: two charger types are randomly selected; their assigned EVs are unscheduled and reassigned to the opposite type. All operators invoke the EV Scheduling Heuristic (Sect. 5.3) with allowed grid capacity constraint violations (`respectGridCapacity = false`). The resulting infeasible offspring are subsequently repaired by Iterative ILP Repair (Sect. 5.5).

5.5 Feasibility Repair Procedures

Worst EV Rejection Repair. This lightweight procedure restores feasibility in two sequential phases. In the first phase, while the grid capacity constraint is violated, the EV that accumulates the highest total charging power across overloaded time slots is identified and rejected. This is repeated until the grid constraint is satisfied. In the second phase, while a charger capacity constraint is violated for some type k, the EV of type k that occupies the greatest number of time slots in excess of the available charger count is rejected. Both phases terminate as soon as the corresponding constraint is satisfied.

Iterative ILP Repair. This heavier repair procedure proceeds in two phases. In the first phase, while the grid constraint is violated and fewer than $I^{\text{lp}}_{\max}$ iterations have been performed, the EV contributing most to the violation is identified according to strategy $R_{\text{lp}} \in \{1, 2\}$: $R_{\text{lp}} = 1$ selects the EV with maximum energy density $e_j/(d_j - r_j)$; $R_{\text{lp}} = 2$ selects the EV with maximum cumulative contribution to grid overload. The selected EV is temporarily rejected, and an ILP-based cost minimization model (Sect. 5.3) is solved under the blended objective $\min(0.5, f_2 + 0.5, \tilde{w}G)$ within a time limit $T\text{grid}$. In this formulation, $\tilde{w}_G$ is introduced as a decision variable representing the grid capacity and is minimized as a secondary objective in order to restore feasibility. In the second phase, rejected EVs are sorted by increasing energy density and greedily reinserted while respecting the grid capacity w_G. If the solution remains infeasible, Worst EV Rejection Repair is applied as a final fallback.

5.6 Pareto Front Refinement

Due to its computational cost, refinement is performed periodically from g_{start} every g_{ref} generations, and at generation G if not previously applied. Rejected EVs are prioritized based on increasing energy density, with ties broken by decreasing temporal slack. Each rejected EV is processed sequentially, with at most 5 insertion attempts and up to 3 accepted variants per solution, under a time budget of 500 ms. Insertion relies on a cascade of operators: a randomized regret-k insertion ($k = 3$) first evaluates all charger types and selects randomly among the k best feasible assignments; if it fails, a single-ejection operator removes one conflicting EV (selected based on temporal overlap, energy density, and charging power), inserts the candidate, and attempts to reinsert the removed EV; if necessary, a multi-ejection operator (depth 2) removes multiple conflicting EVs and is accepted only if it strictly improves the number of satisfied EVs. A candidate is accepted only if it satisfies grid feasibility, strictly improves the number of satisfied EVs, and respects a cost tolerance $(1 + \varepsilon_{\text{cost}})$ with $\varepsilon_{\text{cost}} = 1.0$. Finally, an ILP-based cost minimization procedure is applied to the refined Pareto front within a time limit T_{lp}.

6 Computational Results

This section introduces the instances, details the parameter settings and tuning, and presents a comparative performance analysis.

6.1 Instances

EV Instance Generation. Synthetic benchmark instances are derived from the real-world EV charging transaction dataset of Zhang et al. [26], comprising 441,077 sessions collected across 13 stations in Jiaxing, China over a two-year period from late 2019 to 2021. After removing abnormal records, zero-energy sessions ($e_j \leq 0.5$ kWh), and duration outliers ($(d_j - r_j) \notin [5, 720]$ min), a hybrid non-parametric generation approach is applied. For each instance, 70% of charging demands are generated by a Gaussian kernel resampling of the empirical joint distribution $(r_j, d_j - r_j, e_j)$ [20], while the remaining 30% are drawn from a three-component Gaussian Mixture Model whose parameters are estimated directly from the cleaned data using the Expectation–Maximization algorithm [4,9]. This strategy is motivated by Kolmogorov–Smirnov tests that reject all standard parametric families (normal, log-normal, Weibull, gamma) for all three variables ($p < 0.001$), due to the multimodal structure of arrival times and the heavy-tailed distributions of charging duration and energy demand. To avoid trivial rejections and ensure that acceptance decisions are driven by resource constraints, generated charging sessions are required to be feasible, assuming a charger power of 22 kW. Therefore, any demand violating the condition is rejected and resampled. Seven instance sizes $n \in \{10, 20, 50, 100, 200, 300, 400\}$ are considered. For each size, 10 instances are generated, yielding 70 benchmark instances in total, indexed sequentially from 1 to 70. We consider three charger types $i \in \{1, 2, 3\}$

with power levels $w_1 = 11$ kW, $w_2 = 22$ kW, and $w_3 = 43$ kW. Instances 1–30 share the same settings ($w_G = 43$, $m_i = 1$, $m = 3$), while instances 31–40, 41–50, and 51–70 use $(w_G, m_i, m) = (114, 3, 9)$, $(190, 5, 15)$, and $(380, 10, 30)$, respectively. The grid capacity is defined as $w_G = \max(\max_i w_i,\ 0.5 \sum_i w_i)$, ensuring both feasibility with the fastest charger and a cap at half the total demand. All chargers operate with efficiency 0.92 [15]. The generated benchmark instances, descriptive statistics, and validation figures are publicly available at[1].

Electricity Prices Scenarios. Four electricity price scenarios were sourced from [8]. Their hourly tariffs over a 24-h horizon are shown in Fig. 1. Scenario 1 applies a flat two-level step tariff; Scenario 2 follows a smooth single-peak profile with a gradual morning increase and afternoon decrease; Scenario 3 exhibits continuously varying prices; and Scenario 4 is peak-heavy, with high prices dominating most of the day and only two short off-peak windows. Varying the price profile allows assessing the sensitivity of the scheduling algorithm to different tariff structures, and in particular its ability to shift charging load toward low-price periods under both simple and highly dynamic pricing patterns.

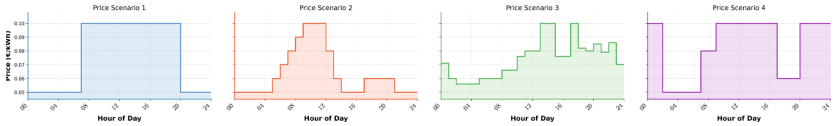

Fig. 1. price scenarios.

6.2 Parameter Settings and Tuning

The two multi-objective approaches described above are implemented: the ε-constraint method and the proposed NSGA-II, enhanced with multithreaded offspring generation. All algorithms are implemented in C++ and executed on a laptop (Intel Core Ultra 5 235U, 2.00 GHz, 16 GB RAM, Windows 11 64-bit). The time slot length is set to 6 min ($\tau = 0.1$ h). A time limit of 30 min per instance is imposed for the ε-constraint method, and the ILP models are solved using Gurobi 12.0.3.

NSGA-II remains sensitive to the selection of its parameter values. Therefore, we employed Optuna [1], a state-of-the-art Python framework for parameter optimization. The optimization process is guided by the hypervolume (HV) indicator defined in [27]. For each instance, we set the reference point for HV calculation as $r = \left(0,\ \max_t e_p^t \cdot \sum_j e_j\right)$ based on problem-specific bounds. We report the normalized hypervolume (NHV) in $[0, 1]$, where values closer to 1 indicate better Pareto front quality.

[1] https://github.com/imyzz/mo-evcsp-benchmarks.



Fig. 2. Parameter importance for small (left) and large (right) instances.

We use Optuna's TPE sampler. To account for instance size sensitivity, we conducted two separate analyses: one on three small instances ($\leq$100 EVs) and another on three large instances (>100 EVs). Parameter importance was assessed via the absolute Pearson correlation between parameter values and NHV. For each configuration, the minimum NHV across the evaluated instances was maximized. To assess the benefit of refinement within NSGA-II, we introduced a boolean parameter `useRefine`: when set to true, the refinement is applied; otherwise, it is not. Parameter ranges and selected values are given in Table 1.

Table 1. "Parameter search space used in Optuna tuning.

Parameter	Type	Min	Max	Step	Small	Large	Parameter	Type	Min	Max	Step	Small	Large
N	Integer	10	100*	5	100	150	α_{orig}	Float	0.10	0.80	0.01	0.38	0.12
G	Integer	10	100	5	100	100	α_{cons}	Float	0.10	0.50	0.01	0.13	0.12
α_m	Float	0.01	0.30	0.01	0.18	0.18	α_{swap}	Float	0.10	0.60	0.01	0.17	0.21
α_c	Float	0.60	1	0.01	0.67	0.76	Q_{des}	Integer	1	15	1	1	13
L_{cons}	Integer	1	20	1	8	7	T_{lp}	Float	1	30	1	15	15
T_{grid}	Float	0.5	10	0.5	7.5	2.5	$I^{\text{lp}}_{\text{max}}$	Integer	1	10	1	6	6
R_{lp}	Categorical	1	2		2	2	`useRefine`	Boolean	True	False		True	True
g_{start}	Integer	0	G	1	44	90	g_{ref}	Integer	1	50	1	18	50

*Upper bound set to 150 for large instances ($n > 100$ EVs).

Results in Fig. 2 show a clear shift in sensitivity between instance sizes. For small instances, N, T_{Grid}, and g_{ref} are the most influential parameters. This is explained by the lower combinatorial complexity, where a larger population (N) improves Pareto front coverage, and frequent refinement (g_{ref}) enhances solution quality by reinserting rejected EVs and performing local improvements. The parameter T_{Grid} controls the time allocated to ILP-based repair used during mutation. Therefore, ILP repair has a significant influence, even though it is applied only in mutation, as it transforms infeasible solutions into feasible ones while minimizing electricity cost. For large instances, T_{grid} becomes the dominant parameter ($\approx$ 0.30), followed by $\alpha_{Q_{\text{des}}}$ and `useRefine`. Since mutation

operators deliberately allow temporary infeasibility, ILP-based repair becomes critical, and insufficient time allocation degrades solutions. The importance of $\alpha_{Q_{\text{des}}}$ highlights the need for greater diversification. Finally, the growing relevance of `useRefine` suggests that refinement procedures, although computationally expensive, are becoming essential for improving solution quality.

6.3 Comparative Performance Analysis

In this section, we compare the two proposed approaches using three key performance metrics: the NHV as described in Sect. 6.2; Pareto-front distribution, assessed through the spacing indicator as defined in [14]; and running time. For the ε-constraint method, a single deterministic run is performed per instance. The developed NSGA-II, hereafter referred to as ILP-NSGA-II, is stochastic and is therefore executed 10 times independently for each instance to ensure statistical robustness and stability.

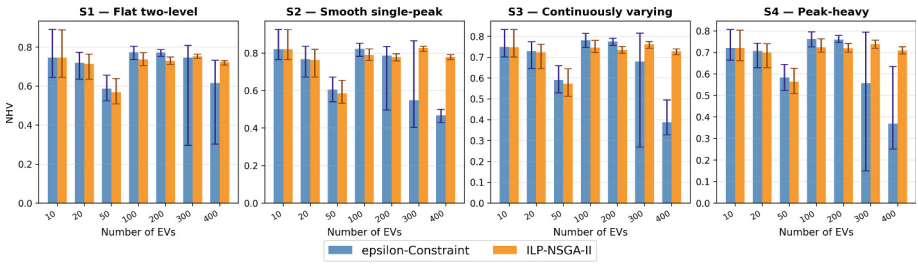

Fig. 3. NHV (mean, min, and max) per group and price scenario.

Figure 3 reports the mean NHV for each instance size group and price profile, with bars indicating the [min, max] range across the ten instances. For instances with 10 EVs, both methods achieve nearly identical performance (0.7590 vs 0.7584, i.e., −0.08%). For 20–50 EVs, ILP-NSGA-II performs slightly worse on average, with limited differences (−0.74% and −3.28%), a trend that continues for 100–200 EVs (−4.60% and −4.34%), although similar NHV values are reached in several runs. In contrast, for large instances (300–400 EVs), ILP-NSGA-II clearly outperforms the ε-constraint method, achieving gains of +21.71% and +59.89%, respectively, highlighting its superior scalability and stability. Note that results are reported as the average NHV over 10 runs, rather than the best value achieved by ILP-NSGA-II. An important observation concerns the influence of the price scenarios. ILP-NSGA-II shows consistent performance across all four scenarios, with limited variability. In contrast, the ε-constraint method is more sensitive to price variations. Scenario S1 (flat two-level) appears to be the easiest, yielding relatively high NHV values, whereas scenarios with more pronounced price variations (S2, S3, and especially S4) lead to lower NHV values, particularly for large instances.

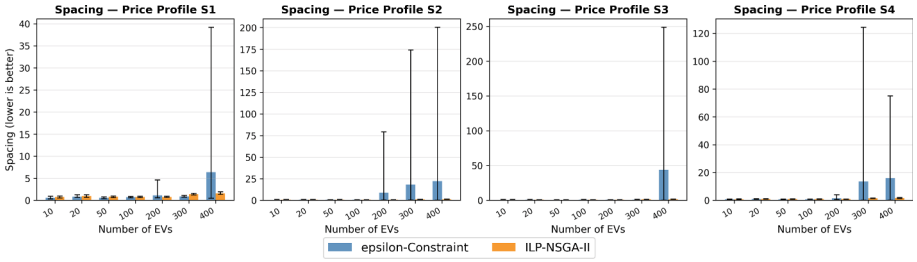

Fig. 4. Spacing (mean, min, and max) per group and price scenario.

Beyond NHV, Pareto-front quality is evaluated using the spacing indicator (lower is better), which measures the uniformity of solution distribution. Low spacing values indicate well-distributed non-dominated solutions. Results in Fig. 4 show that both methods achieve comparable spacing on smaller instances. As instance size increases, the spacing of the ε-constraint method rises more sharply and irregularly, reflecting a degradation in front quality due to time limits. In contrast, ILP-NSGA-II maintains more stable spacing values, demonstrating better robustness in preserving solution diversity.

Figure 5 reports the running time as a function of instance size for each price scenario. For instances with 10–50 EVs, both methods solve the problem very quickly (in less than 40 s). Starting from 100 EVs, differences begin to emerge: the ε-constraint method exhibits a sharp increase in CPU time. The gap becomes pronounced from 200 EVs onward. At this scale, the ε-constraint method requires more than 1000 s and reaches the imposed time limit (1800 s) for 300 and 400 EVs in most scenarios. ILP-NSGA-II maintains moderate execution times that grow smoothly with instance size and remains up to 77% faster, with an average reduction of 65% for instances ranging from 50 to 400 EVs. ILP-NSGA-II shows stable behaviour across all price scenarios, while the ε-constraint method is more sensitive to price structures, with lower execution times in S1 and significantly higher ones in scenarios with more pronounced price variations (S3 and S4). Both methods perform similarly up to 100 EVs, whereas ILP-NSGA-II becomes clearly preferable from around 200 EVs.

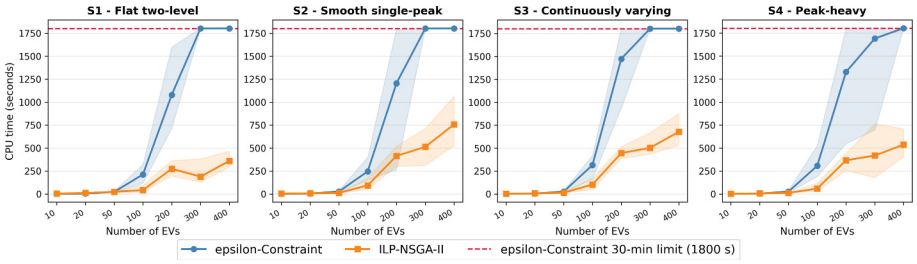

Fig. 5. Execution time per instance size.

Due to the large number of instances, only two representative cases are presented to illustrate the methods' Pareto front behaviour under different price scenarios. Figures 6 and 7 show Pareto front approximations for instances with 100 EVs and 400 EVs, respectively. The 100-EV instance is selected to represent a regime where the ε-constraint method slightly outperforms ILP-NSGA-II in terms of NHV, while the 400-EV instance illustrates a regime where the exact method collapses due to computational limitations, and ILP-NSGA-II provides a much better coverage of the Pareto front. Each colored dot corresponds to a non-dominated solution obtained from one of the ten ILP-NSGA-II runs, while blue stars represent solutions from the ε-constraint method.

For the 100-EV instance, both methods produce almost similar Pareto fronts across all price scenarios. However, differences appear in the low-satisfaction region (below approximately 40 satisfied EVs), where the ε-constraint method provides a denser and more complete coverage of the front, while ILP-NSGA-II yields fewer solutions. A clear impact of the price profile on the Pareto fronts can be observed. Scenario S2 consistently yields the lowest costs. Scenarios S1 and S3 exhibit similar cost ranges, leading to intermediate performance. In contrast, scenario S4 results in significantly higher costs.

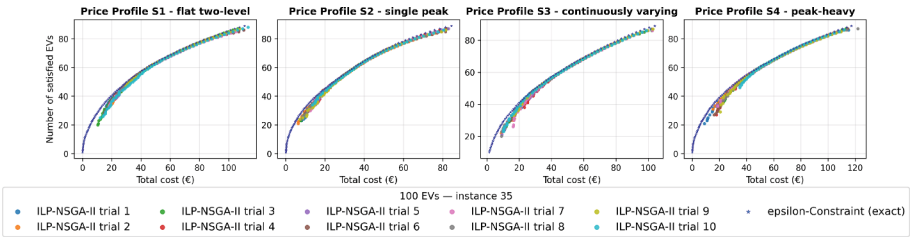

Fig. 6. Pareto front for instance 35 (100 EVs) across all price scenarios.

The 400-EV instance (Fig. 7) exhibits a markedly different behaviour. The ε-constraint front is partially truncated, especially under scenarios S2 and S4, due to the 30-minute time limit, resulting in sparse solutions confined to a limited region of the objective space. This is consistent with the near-1800 s CPU times

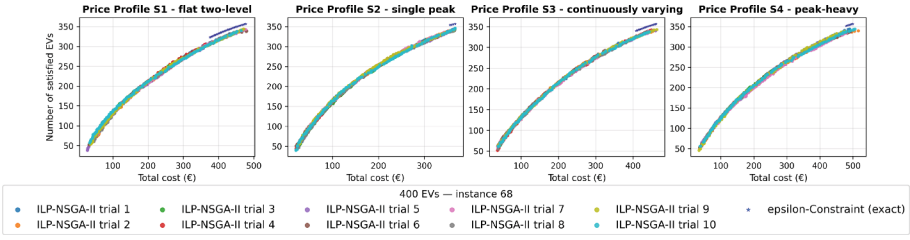

Fig. 7. Pareto front for instance 68 (400 EVs) across all price scenarios.

(Fig. 5) and the observed NHV degradation. In contrast, ILP-NSGA-II provides a dense and well-distributed approximation over a wider region. However, in the high-satisfaction region, its solutions remain slightly below those of ε-constraint, as the search is limited by the imposed computational budget (population size and number of generations). This suggests that additional search effort could further improve the approximation of extreme solutions. Consistent with previous observations, scenario S2 yields the lowest costs.

7 Conclusion

This paper studies a bi-objective EV charging scheduling problem with heterogeneous chargers and time-varying electricity prices, and compares an ε-constraint method with NSGA-II. Experiments conducted on 70 instances under four pricing scenarios show that the proposed NSGA-II significantly improves performance on large-scale problems, achieving up to +59.9% in normalized hypervolume for instances with 400 EVs (and +40.8% on average for instances with 300–400 EVs), while reducing computation time by 65% on average for instances with ≥ 50 EVs. Moreover, the method maintains well-distributed Pareto fronts. In addition, electricity price profiles are shown to significantly affect algorithm performance, particularly for exact methods. This conclusion holds for w_G set to 50% of the total charging power, with the understanding that lower values of w_G increase the problem's difficulty. Future work includes improving the ε-constraint method by avoiding unit-step enumeration, enhancing the NSGA-II framework, and conducting comparisons with other multi-objective algorithms. Additional directions involve exploring alternative charging station configurations and integrating more complex objective functions.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)
2. Crouse, M., Trousdale, D., Mulley, B., Mahmud, M., Cunningham, S.: Electric vehicle enabled buildings: evidence review of installation costs. Technical report, ClimateXChange (May 2022), report commissioned by the Scottish Government (2022)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B **39**(1), 1–38 (1977)
5. Elghanam, E., Abdelfatah, A., Hassan, M.S., Osman, A.H.: Optimization techniques in electric vehicle charging scheduling, routing and spatio-temporal demand coordination: a systematic review. IEEE Open J. Veh. Techno. **5**, 1294–1313 (2024)

6. Haimes, Y.: On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. Syst. Man Cybern. **3**, 296–297 (1971)
7. Global ev outlook 2025, International Energy Agency (2025). licence: CC BY 4.0
8. Karatzinis, G., et al.: Chargym: an EV charging station model for controller benchmarking. In: Maglogiannis, I., Iliadis, L., Macintyre, J., Cortez, P. (eds.) AIAI 2022. IFIPAICT, vol. 652, pp. 241–252. Springer. Cham (2022). https://doi.org/10.1007/978-3-031-08341-9_20
9. Lahariya, M., et al.: Synthetic data generator for electric vehicle charging sessions: modeling and evaluation using real-world data. Energies **13**(16), 4211 (2020)
10. Mahato, D., Aharwal, V.K., Sinha, A.: Multi-objective optimisation model and hybrid optimization algorithm for electric vehicle charge scheduling. J. Exp. Theoret. Artif. Intell. **36**(8), 1645–1667 (2024)
11. Mahlberg, J.A., Desai, J., Bullock, D.M.: Evaluation of electric vehicle charging usage and driver activity. World Electric Veh. J. **14**(11), 308 (2023)
12. Mathioudaki, A., Tsaousoglou, G., Varvarigos, E., Fotakis, D.: Online menu-based scheduling of electric vehicle charging. IEEE Trans. Smart Grid (2025)
13. Mishra, S., Mondal, A., Mondal, S.: A multi-objective optimization framework for electric vehicle charge scheduling with adaptable charging ports. IEEE Trans. Veh. Technol. **72**(5), 5702–5714 (2022)
14. Schott, J.R.: Fault tolerant design using single and multicriteria genetic algorithm optimization (1995)
15. Sevdari, K., Calearo, L., Bakken, B.H., Andersen, P.B., Marinelli, M.: Experimental validation of onboard electric vehicle chargers to improve the efficiency of smart charging operation. Sustain. Energy Technol. Assess. **60**, 103512 (2023)
16. Sone, S.P., Lehtomäki, J.J., Khan, Z., Umebayashi, K., Kim, K.S.: Robust EV scheduling in charging stations under uncertain demands and deadlines. IEEE Trans. Intell. Transp. Syst. (2024)
17. Tan, M., Ren, Y., Pan, R., Wang, L., Chen, J.: Fair and efficient electric vehicle charging scheduling optimization considering the maximum individual waiting time and operating cost. IEEE Trans. Veh. Technol. **72**(8), 9808–9820 (2023)
18. Tayri, A., Ma, X.: Grid impacts of electric vehicle charging: a review of challenges and mitigation strategies. Energies **18**(14), 3807 (2025)
19. Tran, C.Q.: Optimal deployment of charging infrastructures for urban networks (2022)
20. Van Kriekinge, G., et al.: Electric vehicle charging sessions generator based on clustered driver behaviors. Energies **14**(2), 37 (2023)
21. Wu, J., Su, H., Meng, J., Lin, M.: Electric vehicle charging scheduling considering infrastructure constraints. Energy **278**, 127806 (2023)
22. Zaidi, I., Oulamara, A., Idoumghar, L., Basset, M.: Maximizing the number of satisfied charging demands in electric vehicle charging scheduling problem. In: Legrand, P., et al. (eds.) EA 2022 LNCS, vol. 14091, pp. 89–102. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-42616-2_7
23. Zaidi, I., Oulamara, A., Idoumghar, L., Basset, M.: Electric vehicle charging scheduling problem: Heuristics and metaheuristic approaches. SN Comput. Sci. **4**(3), 283 (2023)
24. Zaidi, I., Oulamara, A., Idoumghar, L., Basset, M.: Maximizing the number of satisfied charging demands of electric vehicles on identical chargers. Omega **127**, 103106 (2024)

25. Zaidi, I., Oulamara, A., Idoumghar, L., Basset, M.: Minimizing grid capacity in preemptive electric vehicle charging orchestration: complexity, exact and heuristic approaches. Eur. J. Oper. Res. **312**(1), 22–37 (2024)
26. Zhang, Y., et al.: A high-resolution electric vehicle charging transaction dataset with multidimensional features in China. Sci. Data **12**, 643 (2025)
27. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, HP. (eds.) PPSN 1998. LNCS, vol. 1498 pp. 292–301. Springer, Heidelberg (1998). https://doi.org/10.1007/bfb0056872