Some Optimization Strategies for Smart Charging of Electric Vehicles

Minh-Hieu Tran*, Nam-Hai Nguyen*,
Phuong-Nam Nguyen†

Phenikaa University, *School of Computing*
Hanoi, 12116, Vietnam

November 2024

Abstract

Background: The widespread adoption of electric vehicles (EVs) is critical for mitigating climate change and transitioning towards a more sustainable future. As the number of EVs on the roads increases, the demand for efficient smart charging solutions becomes more pressing. However, optimizing smart charging for EVs is a complex area that has not yet been fully explored. This study intends to tackle this pressing challenge by leveraging advanced computational techniques, specifically genetic algorithms (GA) and particle swarm optimization (PSO), to enhance the scheduling processes in smart charging hubs.
Approach: In our research, we thoroughly investigate five existing objective functions that are commonly used in the context of smart charging. In addition to these established methods, we introduce an innovative safety-aware loss function designed to ensure the reliability and safety of the charging process. This leads us to develop a comprehensive framework consisting of a total of twelve distinct optimization strategies, each tailored to optimize different aspects of smart charging operations.
Result: To evaluate the effectiveness and computational efficiency of these strategies, we conduct a series of rigorous numerical experiments. These experiments not only assess the performance of each strategy under varying conditions but also provide insights into their strengths and limitations. Furthermore, we delve into the scalability of the proposed optimization framework, exploring how well it adapts to larger networks of charging hubs and an increasing number of users.

A**Conclusion**: Based on our findings, we offer practical recommendations aimed at facilitating the implementation of smart charging hubs in real-world scenarios. These insights are designed to assist policymakers, urban planners, and technology developers in crafting strategies that support the sustainable integration of EVs into our transportation systems while addressing the complexities of energy management and user safety[1].

*These authors share the same contribution
†Corresponding author: nam.nguyenphuong@phenikaa-uni.edu.vn.
[1]This research was fully supported by RF-GE-N.00001 and RF-EV-N.00001 from G.A.I.A QTech, Vietnam. This article is being peer-reviewed and is not a final publication. Distribution without the author's consent is prohibited.

Contents

1 Introduction

The global transition to electric vehicles (EVs) is driven by the urgent need to mitigate climate change and reduce dependency on fossil fuels. Smart charging, which dynamically adjusts charging patterns based on real-time data such as energy availability, grid demand, and user needs, has emerged as a critical enabler for optimizing energy usage. This technology is particularly significant for integrating renewable energy sources, such as wind and solar, into the power grid. By aligning EV charging demand with periods of high renewable energy generation, smart charging facilitates efficient utilization of green energy, reducing carbon emissions and reliance on conventional power resources.

The global EV market has experienced exponential growth in recent years, driven by technological advancements, supportive policies, and increasing consumer demand. According to the International Energy Agency (IEA)[5], battery electric vehicles (BEVs) represented two-thirds of both new electric car registrations and the global EV stock in 2023. China led with the largest fleet, comprising 4.5 million electric cars, while Europe recorded the highest annual growth, bringing its total to 3.2 million vehicles (see Appendix A.0, Figure 4).

In 2024, the EVs market in Asia is anticipated to generate a remarkable revenue of US$407.3 billion. This market is expected to experience a steady annual growth rate (CAGR of 2.31%) from 2024 to 2029, leading to a projected market value of US$456.5 billion by 2029[10]. According to HSBC business[9], China continues to lead the EV market, with EVs projected to account for approximately 45% of new vehicle sales this year. South and Southeast Asia are emerging as key focus areas for EV manufacturers, driven by a growing middle class, supportive government policies, and aspirations to advance manufacturing capabilities. An EY report predicts an average annual growth rate of 16–39% in the EV market across the six largest ASEAN economies from 2021 to 2035, with annual sales potentially reaching $100 billion by 2035. This growth presents developing markets with opportunities to leapfrog traditional automotive technologies, as the global fleet of passenger and light commercial vehicles is expected to more than double to 2.5 billion by 2050, fueled primarily by demand from these regions. However, Asia's diverse policies, infrastructure readiness, and economic disparities will lead to varied paths toward mobility electrification. While two- and three-wheelers and public transport will dominate some markets, private cars and commercial fleets will take precedence in others. Vietnam exemplifies this diversity, being the world's second-largest electric two-wheeler market (E2Ws). According to HSBC Global Research, combined annual sales of E2Ws and electric cars in Vietnam could increase from under one million in 2024 to over 2.5 million by 2036.

In the EV 2024 report by KPMG[6], although electric vehicles are still emerging in Vietnam's market, they have sparked considerable local interest, with nearly 70% of survey respondents indicating a willingness to purchase an EV, including both fully electric and hybrid models. Individuals aged 25-44 with greater financial independence are more open to adopting new and emerging technologies like fully electric or hybrid vehicles. In contrast, older generations and the youngest group generally favor traditional internal combustion engine (ICE) vehicles. *More importantly*, fast charging capacity and prevalence of charging stations are two key success factors of charging station systems, followed by charging cost, compatibility of types and brands, and user-friendliness. Regarding preferred EV features, power, range, and the availability of charging stations are key considerations. Younger buyers, however, emphasize sustainability, technology, and insurance perks more.

Smart charging has garnered increasing global attention as a critical solution for sustainable EV adoption. For instance, Canada's recent EV Smart Charging Challenge has highlighted the international push towards developing innovative charging solutions to optimize grid efficiency and support renewable integration [1]. However, despite this growing momentum, there remains a notable research gap. While global efforts have focused on expanding EV adoption and building charging networks, there needs to be more emphasis on developing and deploying smart charging solutions, particularly in emerging EV markets such as Vietnam. This gap presents a critical opportunity for advancing research and innovation in this domain.

This study aims to address this gap by investigating smart charging strategies tailored to the unique challenges of emerging markets. Focusing on Vietnam as a case study, this research seeks to contribute to the global discourse on sustainable EV integration, with implications for policy, technology development, and green energy utilization. Specifically, we address the smart charging problem of EVs using three PSO algorithms and newly designed objective functions. The contribution of this project is three-fold:

- We compare the efficiency and effectiveness of two optimization algorithms for the smart charging problem of EVs, including genetic and PSO algorithms;
- We propose an objective function to simultaneously encourage the charging efficiency and presenting overload of electrical systems, named as a safety-aware objective function (Section 3.3.6);
- In our proof of concept, we successfully reduce 30% of the maximum energy required to change 10 EVs concurrently.

This article is organized as follows:

Section 2 introduces the smart charging problem of EVs and investigates optimization algorithms;

Section 3 introduces our proposed framework;

Section 4 reports the numerical results using our experimental design;

Section 5 discusses the scalability; limitation and potential extension of our proposed approach;

Section 6 concludes the research.

2 Backgrounds

2.1 Smart charge problem for EVs

Despite its importance, only some studies have tackled smart charging for electric vehicles (EVs), limiting our understanding of the issue. ACN-Sim[7, 8] is an open-source simulation environment and dataset designed to advance research in smart EV charging. It provides a realistic platform for evaluating algorithms and testing assumptions, modeling the complexities of real charging systems, and integrating with tools like ACN-Data and ACN-Live. It also supports grid simulators such as MATPOWER, PandaPower, OpenDSS, and OpenAI Gym for training reinforcement learning agents. [2] presents a quantum algorithm that demonstrates the potential of the Quantum Approximate Optimization Algorithm (QAOA) to outperform conventional algorithms in certain scenarios. However, more testing on larger instances and actual quantum devices is needed. [4] propose a solution that enhances the EV charging experience by allowing advanced reservations beyond the OCPP standard's limitations. Their algorithm optimizes station use, eliminates overlapping reservations, and improves user satisfaction while ensuring secure transactions through a central system.

Due to the importance of the problem, Quantum City, the University of Calgary, ATCO SpaceLab, Canadian Natural, and Amazon Quantum Solution Lab jointly hosted a challenge during 2023-2024, which seeks efficient quantum algorithms of the smart charging problem[1]. The issue at hand pertains to an urban area that has experienced significant adoption of EVs yet needs more support in home charging access due to its high population density. In response to this challenge, the city intends to establish an independent, publicly accessible network of EV charging stations to meet EV owners' charging requirements effectively. The sketch of a desirable solution by [1] is illustrated in Figure 5. In alignment with this vision, DENSO[3] and FPT, Vietnam, organized a hackathon focused on advancing toward more sustainable manufacturing practices. This research is an extension of a proposal that is in the top 10 finalists in this competition.

2.2 Optimization algorithms

2.2.1 Genetic Algorithm (GA)

A genetic algorithm (GA) is a meta-heuristic optimization technique inspired by the principles of natural selection and genetics. It is commonly used to solve complex optimization problems by iteratively improving a population of candidate solutions based on a fitness function. The algorithm operates through key evolutionary mechanisms:

1. **Initialization**: A population of potential solutions (chromosomes) is randomly generated or seeded.
2. **Selection**: Individuals with higher fitness values are preferentially selected for reproduction, ensuring that favorable traits are more likely to propagate to the next generation.
3. **Crossover** (Recombination): Pairs of selected individuals exchange genetic information to produce offspring, introducing diversity and allowing for the combination of beneficial traits.
4. **Mutation**: Random alterations are introduced in offspring to explore new regions of the solution space and avoid premature convergence to local optima.
5. **Replacement**: A new generation is formed by replacing some or all of the population with the offspring.

This process repeats over multiple generations, with the population gradually converging toward optimal or near-optimal solutions. GAs are particularly effective for solving non-linear, multi-modal, or high-dimensional optimization problems where traditional methods may struggle.

2.2.2 Particle Swarm Optimization (PSO)

PSO is a population-based optimization algorithm inspired by the social behavior of swarms, such as bird flocking or fish schooling. It is widely used for solving continuous and discrete optimization problems. PSO operates by simulating the collective behavior of particles (candidate solutions) that navigate the solution space to find the optimal or near-optimal solution. The key steps of PSO include

1. **Initialization**: A swarm of particles is initialized with random positions and velocities within the solution space.
2. **Evaluation**: Each particle's position is evaluated using a fitness function to determine its quality as a solution.
3. **Personal/Individual Best (pBest)**: Each particle remembers the best position it has found.

4. **Global Best (gBest)**: The swarm identifies the best position found by any particle as the global best.

5. **Velocity update**: Particles adjust their velocities based on their distance to their individual best and the global best, incorporating stochastic coefficients to balance exploration and exploitation.

6. **Position update**: Particles update their positions based on their new velocities.

The process iterates until a stopping criterion is met, such as a maximum number of iterations or convergence to a solution. PSO's strength lies in its simplicity, low computational cost, and ability to handle multidimensional and nonlinear optimization problems. It is particularly effective for problems where the objective function is nondifferentiable or has multiple local optima.

3 Proposed Framework

3.1 Problem statement

Let N be the number of stations in the charging hub, which is equal to the maximum number of EVs (capacity). We assume that each EVs has the same maximum battery capacity of $F := 50,000\text{W} = 50\text{kW}$, similar to [1]. We use a fixed voltage of $U = 220(\text{V})$ for the Vietnamese market. The controllable charging current takes a value in the option

$$I_{\text{option}} = \{8, 16, 32, 48, 64\}. \tag{1}$$

The power is computed by $P = UI$ (W); thus, the range of P is $[1760, 14080]$ (W) or $[1.76, 14.08]$ (kW). In an extreme case, all N stations will be used to charge EVs with request of full capacity; hence, the maximum required power is $10 \times 50 = 500\text{kW}$. However, in practice, these extreme case rarely happens; instead, the requested charging of an EV is an i.i.d random variable B, which is uniformly drawn from $U[0, F]$. Besides, we assume that all stations are requested, simulating the high-demand period, which could lead to overloading of the charging hub. We set the maximum allowable power to 70% of the extreme case, which is $P_{\text{allowance}} = 350$ (kW). The control horizon is 60 minutes, divided into 13 time steps; i.e., $t_0 = 0$, $t_{\max} = 60$, $s = 13$ and $\delta t = (t_1 - t_0)/(s-1) = 5$ minutes. In this setting, we will adjust the charging current in a station adaptively, corresponding to the total power used by all stations in the charging hub.

As a result, we present the power consumption as a matrix

$$\mathbf{P} = U \cdot \mathbf{I} = 220 \cdot \begin{bmatrix} I_{0,0} & I_{0,1} & \dots & I_{0,12} \\ I_{1,0} & I_{1,1} & \dots & I_{1,12} \\ & & & \\ I_{9,0} & I_{9,1} & \dots & I_{9,12} \end{bmatrix} \tag{2}$$

In a generalized case, the power matrix $\mathbf{P}$ is $N \times s$; and the control current $I_{p,q}$ with $p \in [0, N-1]$ and $q \in [0, s-1]$. The sequence

$$\text{CI}_p = (I_{p,0}, I_{p,1}, \dots, I_{p,s-1}) \tag{3}$$

is a charging configuration of an individual EV. Meanwhile, the summation (columns sum of $\mathbf{I}$)

$$\text{CP}(T) = 220 \cdot \sum_{p=0}^{N-1} I_{p,T} \tag{4}$$

is the power of all stations used at time $T \in [0, s-1]$. With the constrain $P_{\text{allowance}}$, we want

$$\mathcal{C} := \text{CP}(T) \le P_{\text{allowance}}; \quad \forall T \in [0, s-1]. \tag{5}$$

For some objective function $\mathcal{L}$ (or fitness/loss function), we find

$$\mathbf{I}^* = \arg\max \mathcal{L}(\mathbf{I}|\mathcal{C}), \tag{6}$$

where $\mathbf{I}^*$ is the optimal value of I that minimizes the objective function $\mathcal{L}(\mathbf{I}|\mathcal{C})$ with the constrain $\mathcal{C}$ in Equation 5. In the next section, we apply two optimization algorithms (denoted as $\mathcal{M}$): (1) genetic algorithm and (2) PSO algorithm to solve for $\mathbf{I}^*$ (Section 3.2), under six objective functions (see Section 3.3).

3.2 Algorithms

3.2.1 GA for smart charging of EVs

In the parameter space, we use

$$\begin{aligned} \text{Population Size: } & P = 50, \\ \text{Generations: } & G = 100, \\ \text{Mutation Rate: } & \mu = 0.1 \end{aligned} \tag{7}$$

Initialization of population: The initial population consists of P individuals:

$$\mathbf{I}_k \sim \text{Uniform}(\mathcal{I}_{\text{options}}) \tag{8}$$

where $\mathcal{I}_{\text{options}}$ is the set of possible current values, and $\mathbf{I}_k \in \mathbb{R}^{N \times s}$ is an individual's charging schedule.

Crossover operation: The initial population consists of P individuals:

$$\mathbf{I}_k \sim \text{Uniform}(\mathcal{I}_{\text{options}}) \tag{9}$$

where $\mathcal{I}_{\text{options}}$ is the set of possible current values, and $\mathbf{I}_k \in \mathbb{R}^{N \times s}$ is an individual's charging schedule.

Crossover operation: Given two parents $\mathbf{I}_1, \mathbf{I}_2$, crossover is performed at a random point c:

$$\begin{aligned} \text{Child 1: } & \mathbf{I}_c^{(1)} = [\mathbf{I}_1[:, :c], \mathbf{I}_2[:, c:]] \\ \text{Child 2: } & \mathbf{I}_c^{(2)} = [\mathbf{I}_2[:, :c], \mathbf{I}_1[:, c:]] \end{aligned} \tag{10}$$

Mutation operation: For an individual $\mathbf{I}_k$, each element $I_{i,t}$ is mutated with probability μ:

$$I_{i,t} = \begin{cases} \text{Random}(\mathcal{I}_{\text{options}}), & \text{if probability less than} < \mu \\ I_{i,t}, & \text{otherwise.} \end{cases} \tag{11}$$

Selection process: The top $P/2$ individuals are selected based on their fitness values:

$$\{\mathbf{I}_k : k \in \text{argsort}(f(\mathbf{I}))[:P/2]\} \tag{12}$$

Update of charging matrix I:

$$M_{i,0} = B_i$$
$$M_{i,t} = \begin{cases} \min(M_{i,t-1} + U \cdot I_{i,t}, F), & \text{if } M_{i,t-1} < F \\ F, & \text{otherwise.} \end{cases} \tag{13}$$

where B_i is the initial battery level of EV i, F is the maximum battery capacity, U is the power conversion factor (voltage) and $I_{i,t}$ is the current applied to EV i at time t.

GA algorithm for smart charging of EVs

Step 1. initialization: Randomly generate P individuals

Step 2. evaluate: Compute $\mathcal{L}(\mathbf{I})$ for each individual using the charging matrix

Step 3. selection: Choose the top $P/2$ individuals

Step 4. generation:

Step 4.1. Perform crossover to create offspring

Step 4.2. Apply mutation to offspring

Step 5. iterate: Repeat steps 2-4 for G generations

Step 6. return: The best charging schedule $\mathbf{I}^*$

3.2.2 PSO for smart charging of EVs

Initialization: We initialize the global best value as

$$\begin{aligned} \text{Global Best Value} &= -\infty, \\ \text{Global Best Position} &= \emptyset \end{aligned} \tag{14}$$

Then, we randomly generate the particle positions and velocities

$$\begin{aligned} \mathbf{X}_i &\sim \text{Random Choice}(\mathcal{I}, (N, s)), \\ \mathbf{V}_i &\sim \mathcal{U}(-1, 1, (N, s)) \end{aligned} \tag{15}$$

and copy initial positions to best positions using

$$\text{IBP}_i := \text{Individual Best Positions}_i = \mathbf{X}_i \tag{16}$$

We evaluate the fitness of each particle's position by $\mathcal{L}(\mathbf{I})$ (discussed later in Section 3.3). The update rule is defined as

$$\begin{aligned} \text{GBV} &:= \text{Global Best Value} = \max \mathcal{L}(\mathbf{I}), \\ \text{GBP} &:= \text{Global Best Position} = \mathbf{I}^*. \end{aligned} \tag{17}$$

PSO algorithm for smart charging of EVs
For each iteration $t = 1, 2, \ldots, G$:

Step 1. velocity update: for each particle i:

$$\begin{aligned} \mathbf{V}_i(t+1) &= w\mathbf{V}_i(t) \\ &+ \phi_p \mathbf{R}_p \odot (\text{IBP}_i - \mathbf{X}_i(t)) \\ &+ \phi_g \mathbf{R}_g \odot (\text{GBP} - \mathbf{X}_i(t)) \end{aligned} \tag{18}$$

where $\mathbf{R}_p, \mathbf{R}_g \sim \mathcal{U}(0, 1, (N, s))$ and $\odot$ is element-wise multiplication.

Step 2. position update:

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1). \tag{19}$$

Clip positions to nearest allowable values in $\mathcal{I}$:

$$\mathbf{X}_i(t+1) = \text{Nearest}(\mathbf{X}_i(t+1), \mathcal{I}) \tag{20}$$

Step 3. objective evaluation: compute

$$\text{IBP}_i = \begin{cases} \mathbf{X}_i(t+1) \\ \quad \text{if } \mathcal{L}(\mathbf{X}_i(t+1)) > \mathcal{L}(\text{IBP}_i) \\ \text{IBP}_i \text{ otherwise.} \end{cases} \tag{21}$$

Step 4. update global best:

$$\text{GBV}, \text{GBP} = \begin{cases} \mathcal{L}(\text{IBP}_i), \text{IBP}_i \\ \quad \text{if } \mathcal{L}(\text{IBP}_i) > \text{GBV}; \\ \text{GBV}, \text{GBP otherwise.} \end{cases} \tag{22}$$

Step 5. return: the global best solution $\mathbf{I}^*$

3.3 Objective functions

Let $\mathbf{I}$ be the optimized matrix, whose entries $I_{p,q}$ are RVs randomly drawn values from current options in Equation 1. We denote the fitness value as fit. Objective functions discussed in Section 3.3.2, 3.3.3 and 3.3.4 are proposed in[2, 7, 8] and reviewed in[1]. Nevertheless, we adjust these loss functions for our problem statement in Section 3.1. Besides, we propose two new objective functions, which are a weighted version of the existing objective function (Section 3.3.5) and a safety-aware cost function (Section 3.3.6).

3.3.1 Generic objective function $[\mathcal{L}^{\text{Gen}}(\mathbf{I}|\mathcal{C})]$

First, we compute the quantity $\text{CP}(T)$ using Equation 4. Then, the fitness value is computed as

$$\text{fit} = -\max\{0, \text{CP}(T) - P_{\text{allowance}}\}, \tag{23}$$

where a negative penalty is given for an exceeding power over $P_{\text{allowance}}$. The `python` code implementation of this objective function is given in Section A1.1.

3.3.2 Quick charging objective $[\mathcal{L}^{\mathbf{QC}}(\mathbf{I}|\mathcal{C})]$

This objective function gives incentive for the rapid charging of all EVs by

$$\text{fit} := \mathcal{L}^{\text{QC}}(\mathbf{I}) = \sum_{T=0}^{s-1} \frac{s-T}{s-1} \cdot \text{CP}(T); \tag{24}$$

where the pre-factor $\gamma := (s-t)/(s-1)$ decreases linearly over time and the incentive for early charging at time T is $\gamma\text{CP}(T)$ for T running from 0 to $s-1$. The pre-factor γ linearly decreases over the controlling horizon, encouraging the charging hub to deliver the total power consumption as quickly as possible. This approach facilitates efficient throughput by ensuring prompt servicing of electric vehicles (EVs) and freeing capacity for subsequent EVs in the queue, particularly when the number of requested charging slots exceeds the available capacity N. The `python` code of this loss function is given in Section A1.2.

3.3.3 Non-competition penalty $[\mathcal{L}^{\mathbf{NC}}(\mathbf{I}|\mathcal{C})]$

This objective function emphasizes meeting the energy charging needs of EV owners within the designated charging time constraints. We modify the original loss function for our problem as

$$\begin{aligned} \text{fit} := \mathcal{L}^{\text{NC}}(\mathbf{I}) &= -\text{Penalty}^{1/\alpha} \\ \text{Penalty} &= \sum_{i=0}^{N-1} \Big(\text{Energy Shortfall}_i\Big)^{\alpha} \\ \text{Energy Shortfall}_i &= |\text{Total Energy}_i - E_{\text{required},i}| \\ \text{Total Energy}_i &= \sum_{j=0}^{s-1} I_{i,j} \cdot U \cdot \Delta t; \end{aligned} \tag{25}$$

where Δt is the step size (in our experiment is 0.5) and $E_{\text{required},i}$ is the required energy for the i-th EV. Increasing α to a value greater than 1 leads to prioritizing EVs with larger energy demands and shorter available charging durations. The python implementation is given in Section A1.3.

3.3.4 Load variation objective function $[\mathcal{L}^{\mathbf{LV}}(\mathbf{I}|\mathcal{C})]$

Load variation refers to the changes or fluctuations in the electrical load (demand for power) over a given period. We want to minimize the total load variations to protect the electrical system, which is defined as

$$\text{fit} := \mathcal{L}^{\text{LV}}(\mathbf{I}) = -\sum_{q=0}^{s-1} \Big(\sum_{p=0}^{N-1} I_{p,q} \Big)^2 \tag{26}$$

The python implementation is given in Section A1.4.

3.3.5 Composite objective function $[\mathcal{L}^{\mathbf{Comp}}(\mathbf{I}|\mathcal{C})]$

We simply weight the existing objective functions $\mathcal{L}^{\text{QC}}$, $\mathcal{L}^{\text{NC}}$ and $\mathcal{L}^{\text{LV}}$ by weights w_{QC}, w_{NC} and w_{LV}, respectively. The composite objective function is given as

$$\mathcal{L}^{\text{Comp}}(\mathbf{I}) = w_{\text{QC}}\mathcal{L}^{\text{QC}}(\mathbf{I}) + w_{\text{NC}}\mathcal{L}^{\text{NC}}(\mathbf{I}) + w_{\text{LV}}\mathcal{L}^{\text{LV}}(\mathbf{I}), \tag{27}$$

where $w \in [0, 1]$.

3.3.6 Safety-aware objective function $[\mathcal{L}^{\mathbf{SA}}(\mathbf{I}|\mathcal{C})]$

To protect the electrical grid, we construct a reward-penalty objective function, which is

$$\text{fit} := \text{Reward} - \text{Penalty}. \tag{28}$$

The reward encourages the total power at each time step $\text{CP}(T)$ to stay close to the maximum power limit ($P_{\text{allowance}}$), with a tolerance margin of $\pm 5\%$. Meanwhile, the penalty term penalizes power exceeding the maximum limit $P_{\text{allowance}}$, with the penalty growing quadratically with the excess. Specifically, we construct

$$\begin{aligned} \text{Reward} &= \sum_{T=0}^{s-1} \left[-\left|\text{CP}(T) - P_{\max}\right| + P_{\text{allowance}} \right] \\ \text{Penalty} &= \sum_{T=0}^{s-1} \left[\max\left(0, \text{CP}(T) - P_{\text{allowance}}\right)\right]^2 \end{aligned} \tag{29}$$

Our proposed objective function is an extension of the generic loss function in Section 3.3.1. However, the reward-penalty mechanism allows us to push the total power consumption as close as possible to the maximum capacity but avoid overload. The `python` code to implement this loss function is given in Section A1.5.

4 Numerical Results

4.1 Experimental environment

We use `python` 3.7.0, and `numpy` for numerical computation. Both GA and PSO are implemented from scratch, following Section 3.2.1 and 3.2.2. For the numerical results, we simulate each experiment using 100 independent runs and report the mean and standard deviation of optimal charging schedule (see Figure 1 and 2). In each sub-figure, the left panel reports the optimal solution, while the right panel illustrates the control current in each station. The shared region represents one standard deviation around the mean.

4.2 Effectiveness analysis

Figure 1 and Figure 2 illustrate the optimization results achieved using the GA and PSO algorithms introduced in Section 3.2.1 and Section 3.2.2, respectively, under the six objective functions described in Section 3.3.

Overall, only two strategies, GA-$\mathcal{L}^{\mathrm{QC}}$ and GA-$\mathcal{L}^{\mathrm{Comp}}$, resulting in overloads beyond the allowable capacity, $P_{\mathrm{allowance}}$ in some simulation. In contrast, all other strategies maintain total power consumption below the allowable threshold. The GA algorithm demonstrates a tendency to schedule high power loads within the initial 0–10 minutes of the control horizon (except for GA-$\mathcal{L}^{\mathrm{Comp}}$). Subsequently, the total power consumption gradually decreases over time, ensuring that all EVs are fully charged by the horizon's end. In contrast, the PSO algorithm distributes the power load more uniformly across the entire control horizon, except for PSO-$\mathcal{L}^{\mathrm{Gen}}$. However, PSO algorithms incorporating load variation and composite loss objectives (PSO-$\mathcal{L}^{\mathrm{LV}}$ and PSO-$\mathcal{L}^{\mathrm{Comp}}$) are less efficient, as a few of EVs remain uncharged after 60 minutes. This phenomenon may result from the load variation constraint, which limits fluctuations in the control current. Notably, the mean charging current for PSO-$\mathcal{L}^{\mathrm{LV}}$ shows minimal variation, whereas GA-$\mathcal{L}^{\mathrm{LV}}$ exhibits significant variation. This suggests that the GA algorithm explores a larger solution space than the PSO algorithm.

Finally, the proposed safety-aware objective function proves to be an effective fitness function. Specifically, the loss function prioritizes solutions that approach the allowable capacity without exceeding the electrical system's limit in both algorithms.

4.3 Efficiency analysis

Table 1 compares the mean running times for two optimization strategies under six different objective functions. The times are measured in seconds ([s]) and represent the approximate time per run.

PSO demonstrates slightly better efficiency than GA, as it has lower runtime values across most objective functions. The average runtime across all objective functions for GA and PSO are ≈ 2.10 [s] and ≈ 1.83 [s]. PSO is approximately 12.9% average faster than GA. PSO maintains consistent runtime improvements across most objective functions. The largest efficiency gap is observed for $\mathcal{L}^{\mathrm{LV}}$, where PSO is 25% faster than GA. The smallest difference is noted for $\mathcal{L}^{\mathrm{QC}}$, with PSO being only 1.5% faster. Although the differences are small for some objective functions, PSO's improved efficiency, particularly for $\mathcal{L}^{\mathrm{Gen}}$, $\mathcal{L}^{\mathrm{NC}}$, and $\mathcal{L}^{\mathrm{LV}}$, makes it a favorable choice for time-critical applications.

5 Discussion

5.1 Scalability of the framework

We examine the scalability of the proposed framework using $N = 100$ stations and calculate the maximum power allowance for the charging hub with $P_{\mathrm{allowance}} = 0.7 \cdot N \cdot F$. For stress testing, we focus on the time-consuming objective function $\mathcal{L}^{\mathrm{Comp}}$. Our findings reveal that the current approach struggles to scale effectively. The Genetic Algorithm (GA) for $\mathcal{L}^{\mathrm{Comp}}$ takes 2,153 seconds (35.9 minutes) to optimize for 100 stations, whereas Particle Swarm Optimization (PSO) takes only 1,595 seconds (26.6 minutes).

Figure 3a shows results from 100 independent runs. The GA-based method peaks at approximately 3,500 kW, reaching the maximum power limit within 0 to 10 minutes. Its mean optimization curve gradually reduces total power consumption but exhibits a wider deviation band. In contrast, the PSO-based approach peaks slightly below 3,000 kW, with a more consistent decline in power consumption and a narrower deviation band, indicating steadier optimization. This suggests that PSO adheres better to constraints while GA engages in more exploration. These trends align with our previous experiments.

5.2 Our recommendations

For practitioners interested in building an actual smart charging hub based on the findings from these optimization strategies, we have several recommendations:

- **Prioritize load management and safety-aware objectives**: Implement a safety-aware objective function to ensure power consumption remains within allowable limits, especially during peak times. This will help maintain system safety while optimizing efficiency.
- **Address load variation constraints in PSO**: PSO variants with load variation constraints (e.g., PSO-$\mathcal{L}^{\mathrm{LV}}$) can struggle with complex demands, leaving some EVs uncharged. Consider relaxing these constraints for more dynamic charging profiles.
- **Dynamic solution space exploration with GA**: For diverse charging loads, GA-based algorithms like GA-$\mathcal{L}^{\mathrm{LV}}$ can explore the solution space more effectively, optimizing schedules early in the charging period.
- **Real-time power monitoring and adjustment**: Both GA and PSO can benefit from real-time feedback. While GAs may cause load spikes, PSO might undercharge some EVs if constraints are too strict. Dynamic adjustments can enhance charging efficiency within capacity limits.

5.3 Limitations and future works

Firstly, our approach does not rely on data-driven methods as seen in [7, 8], which incorporate external datasets for smart charge scheduling. We aim to overcome this limitation in future work by collecting a local dataset to enable a data-driven solution. Secondly, despite evaluating 12 optimization strategies, we still need to address their comparison using formal metrics. The development of such evaluation metrics warrants further investigation. Thirdly, while we acknowledge the existence of other PSO-based algorithms and combinatorial optimization techniques applicable to this problem, we regard both GA and PSO as valuable benchmarks for encouraging future research on algorithmic development. Besides, the hyperparameter optimization of GA and PSO should be investigated to determine the robustness of these algorithms under different configurations. Lastly, the scalability of our proposed methods is currently limited to more than a single charging hub. We anticipate that more efficient implementations, such as utilizing distributed or parallel computing and leveraging more powerful computational resources, will be necessary to extend this approach to multiple hubs.

6 Conclusion

To achieve the goal of this project, we introduced the smart charging problem for electric vehicles (EVs) in Section 3.1 and provided a review of existing approaches in Section 2.1. Adaptations of GA and PSO for this problem are discussed in Section 3.2, along with six objective functions outlined in Section 3.3. As a result, we analyzed 12 optimization strategies regarding their effectiveness in Section 4.2 and efficiency in Section 4.3. Following this, we addressed the scalability of our proposed approach, offered recommendations for practical deployment, highlighted the limitations of the current research, and outlined several directions for future research in Section 5.

References

[1] Q. C. Challenge. Tracks, 2024. Accessed: 2024-11-23.

[2] C. Dalyac, L. Henriet, E. Jeandel, W. Lechner, S. Perdrix, M. Porcheron, and M. Veshchezerova. Qualifying quantum approaches for hard industrial optimization problems. a case study in the field of smart-charging of electric vehicles. *EPJ Quantum Technology*, 8(1):12, 2021.

[3] DENSO and FPT. Denso factory hacks 2024. `https://densohackathon.com/home`, 2024. Accessed: 2024-11-23.

[4] R. Flocea, A. Hîncu, A. Robu, S. Senocico, A. Traciu, B. M. Remus, M. S. Răboacă, and C. Filote. Electric vehicle smart charging reservation algorithm. *Sensors*, 22(8):2834, 2022.

[5] I. E. A. (IEA). Global ev outlook 2021: Trends and developments in electric vehicle markets, 2021.

[6] KPMG. The ev landscape in vietnam, 2024. Accessed: 2024-11-23.

[7] Z. Lee, D. Johansson, and S. H. Low. Acn-sim: An open-source simulator for data-driven electric vehicle charging research. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 411–412, 2019.

[8] Z. J. Lee, T. Li, and S. H. Low. Acn-data: Analysis and applications of an open ev charging dataset. In *Proceedings of the tenth ACM international conference on future energy systems*, pages 139–149, 2019.

[9] H. G. Research. Driving new opportunities in asia's auto sector, 2024.

[10] Statista. Electric vehicles - asia, 2024. Accessed: 2024-11-23.

Declarations

Ethics Approval and Consent to Participate

Not applicable.

Consent for Publication

All authors consent to the publication of the pre-print article.

Availability of Data and Materials

The Python code used for the implementation of the objective functions is provided in Appendix A.1. Full code implementation can be shared upon request. No new data were generated as part of this research.

Competing Interests

The authors declare that there are no known competing financial interests related to this paper.

Funding

This research was fully supported by RF-GE-N.00001 and RF-EV-N.00001 from G.A.I.A QTech, Vietnam.

Acknowledgments

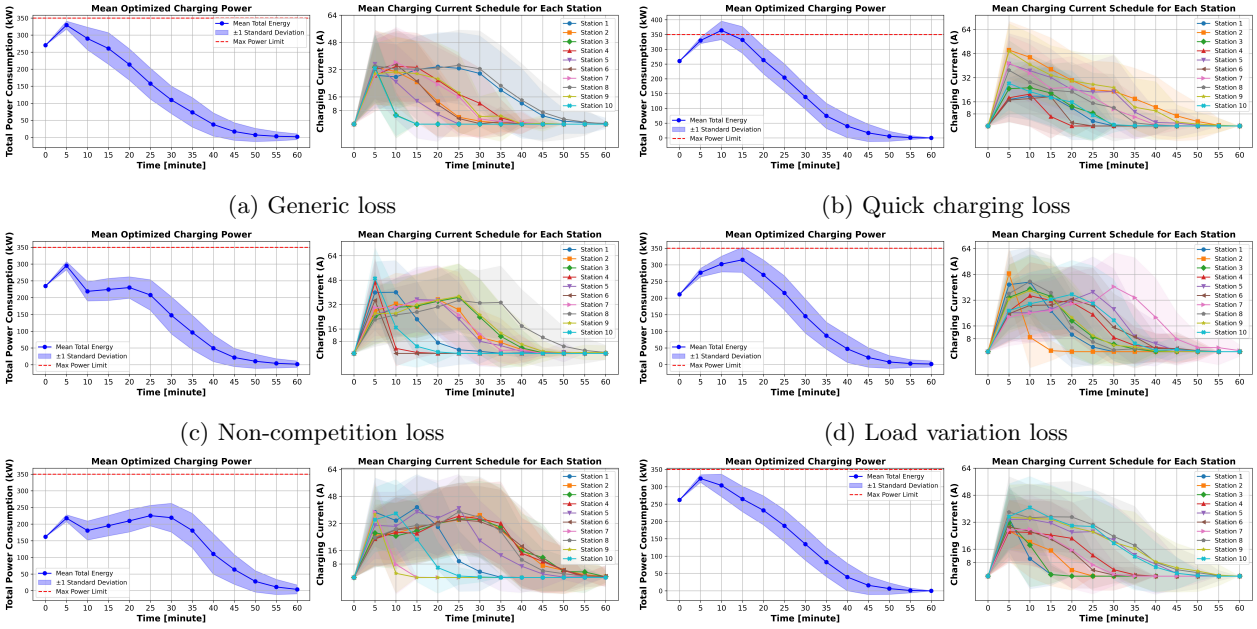
The authors would like to express their gratitude to colleagues at Phenikaa University of FPT for their stimulating discussions and support during the hackathon.

Author Contributions

PN conceptualized the problem and formulated the genetic algorithm. MH conceptualized the particle swarm optimization algorithm and implemented the existing loss functions. NH conceptualized the safety-aware objective function. All authors contributed to performing the numerical experiments, analyzing and discussing the results, preparing the first draft, and reviewing the final article. PN was responsible for securing the funding for the research.

Table 1: Efficiency analysis of optimization strategies using GA and PSO under the six investigated objective functions, measured using the mean running time in [s]econd; i.e., approximate time elapsed per run

	GA	**PSO**
$\mathcal{L}^{\text{Gen}}$	1.63[s]	1.26[s]
$\mathcal{L}^{\text{QC}}$	2.01[s]	1.98[s]
$\mathcal{L}^{\text{NC}}$	2.09[s]	1.82[s]
$\mathcal{L}^{\text{LV}}$	1.96[s]	1.47[s]
$\mathcal{L}^{\text{Comp}}$	3.19[s]	3.13[s]
$\mathcal{L}^{\text{SA}}$	1.71[s]	1.34[s]



(a) Generic loss

(b) Quick charging loss

(c) Non-competition loss

(d) Load variation loss

(e) Composite loss

(f) Safety-aware loss

Figure 1: Optimization result for smart charge scheduling of EVs using GA algorithm in Section 3.2.1

Appendix

A0. EV market research by IEA and desirable solution of smart charging system

We reuse market research of EVs form IEA[5] under BY-CC-4.0, demonstrated in Figure 4. A sketch for a good optimized solution by [1] is in Figure 5.

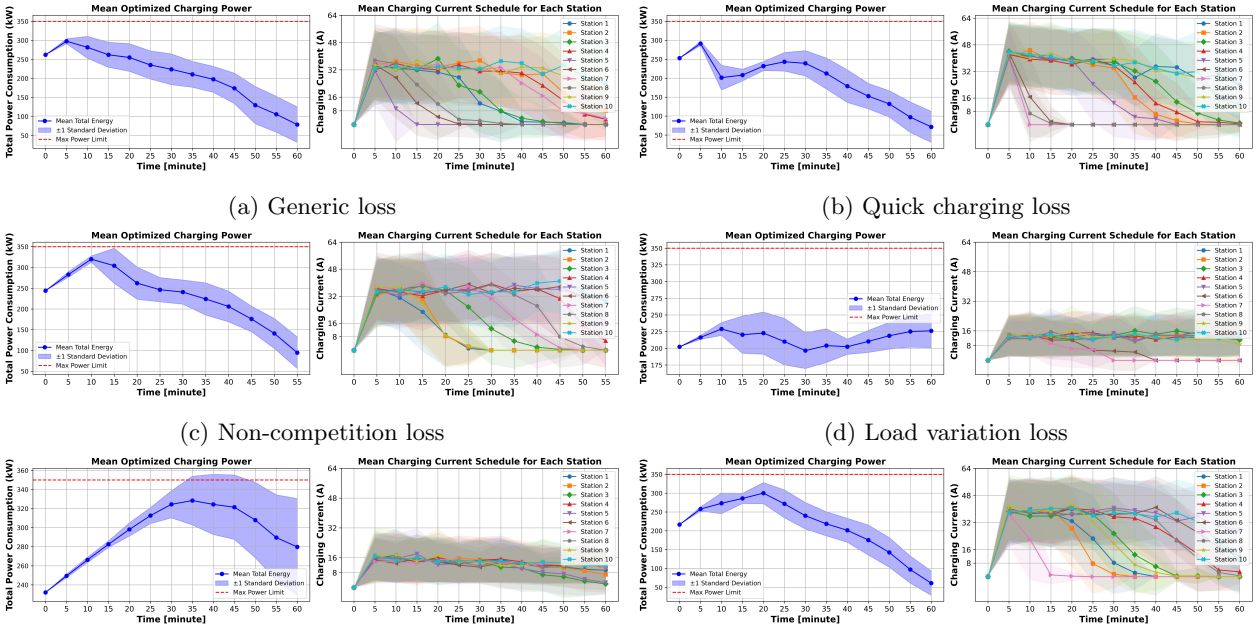
A1. `Python` code to implement objective functions

A1.1 Generic objective function

```python
def GenericLoss(I):
    column_sum = I.sum(axis=0)
    excess_power = np.maximum(0, column_sum - max_power_limit)
    # Negative penalty for exceeding power
    fitness_value = -np.sum(excess_power)
    return fitness_value
```

Listing 1: The python code of generic objective function

A1.2 Quick charging objective function



(a) Generic loss

(b) Quick charging loss

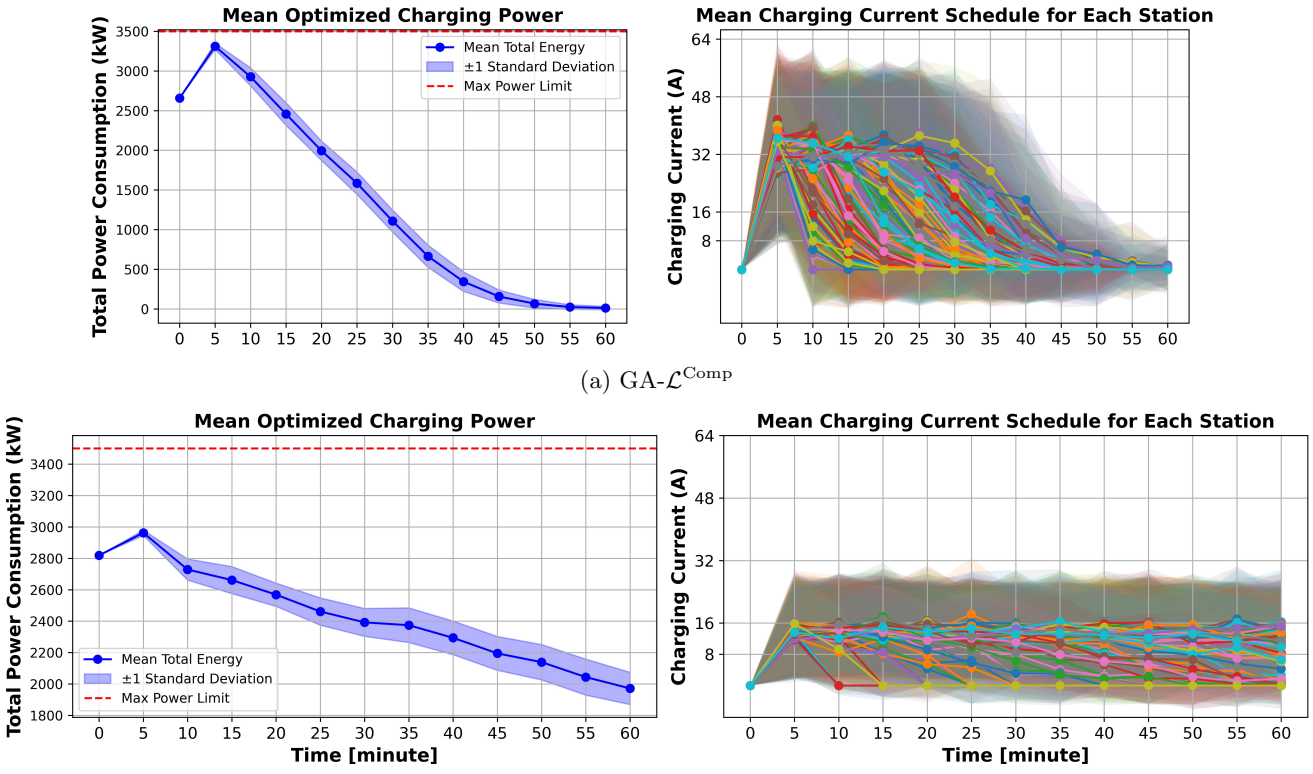
(c) Non-competition loss

(d) Load variation loss

(e) Composite loss

(f) Safety-aware loss

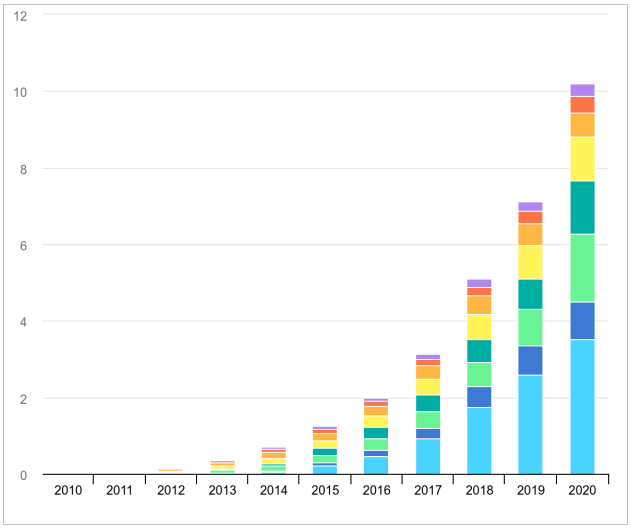
Figure 2: Optimization result for smart charge scheduling of EVs using PSO algorithm in Section 3.2.2



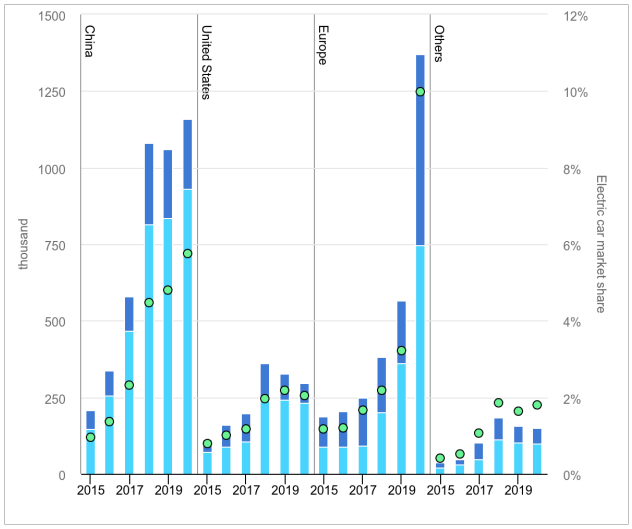
(a) GA-$\mathcal{L}^{\text{Comp}}$

(b) PSO-$\mathcal{L}^{\text{Comp}}$

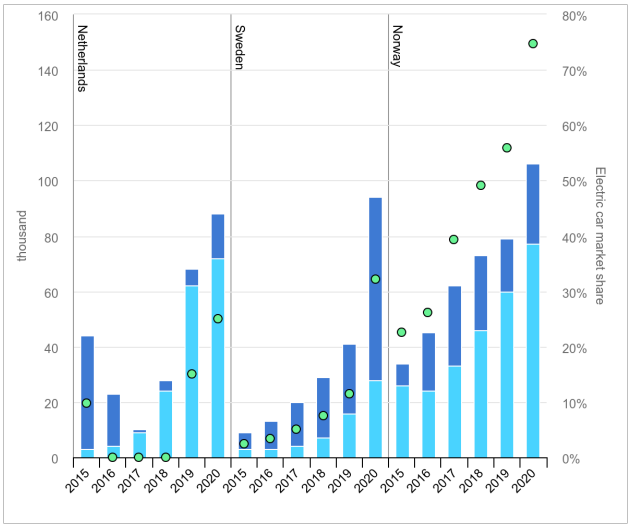
Figure 3: Scalability analysis of GA and PSO algorithm using composite objective function for $N = 100$ charging stations



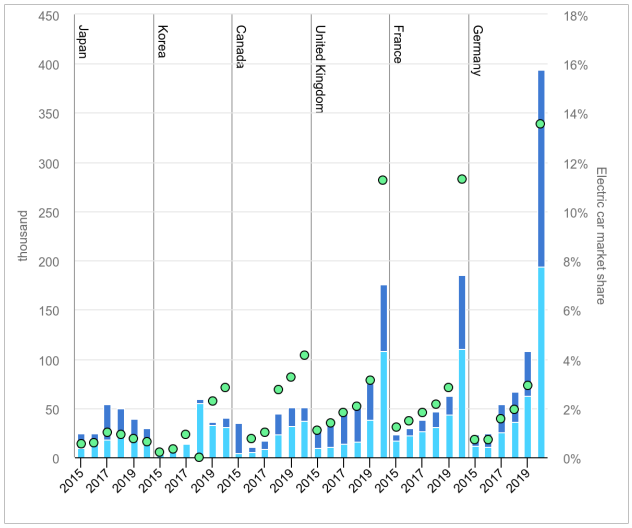
(a) Global electric passenger car stock (2010-2020)



(b) Global electric car registrations and market share (2015-2020)



(c) Electric car registrations and market share in North-Western Europe (2015-2020)



(d) Electric car registrations and market share in selected countries (2015-2020)

Figure 4: Trends and developments in electric vehicle markets analyzed by IEA[5]. The figures are reused under IEA. License: CC-BY-4.0-The Organisation for Economic Co-operation and Development (OECD), on behalf of the IEA (collectively the OECD/IEA) remains the owner of any intellectual property rights in the CC-licensed Content

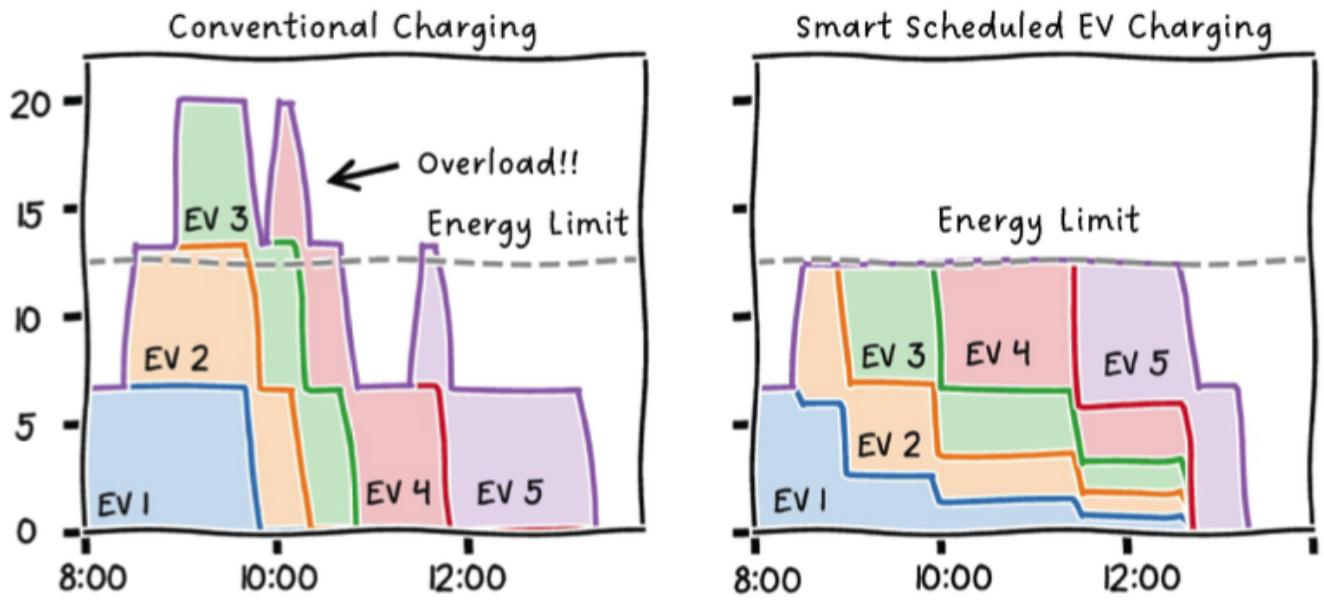


Figure 5: A sketch for a good solution by[1]

```python
def QuickChargeLoss(particle):
    total_quick_charge = 0
    for T in range(s):
        # Prefactor decreases linearly over time
        prefactor = (s - T) / (s - 1)
        # Sum current for all EVs at time t - CP(T)
        CP_T = np.sum(particle[:, t])
        # Incentivize early charging
        fitness_value += prefactor * CP_T
    return fitness_value
```

Listing 2: The python code of quick charging objective function

A1.3 Non-competition objective function

```python
def NonCompetitionLoss(particle, alpha):
    penalty = 0
    for i in range(N):
        total_energy = np.sum(particle[i, :] * U * Delta_t)
        energy_shortfall = abs(total_energy - E_required[i])
        penalty += energy_shortfall ** alpha
    fitness_value = -penalty**(1/alpha)
    return fitness_value
```

Listing 3: The python code of non-competition objective function

A1.4 Load variation objective function

```python
def LoadVariationLoss(particle):
    fitness_value = 0
    for t in range(s):
        N_t = np.sum(particle[:, t])
        fitness_value += -N_t ** 2
    return fitness_value
```

Listing 4: The python code of load variation objective function

A1.5 Safety-aware objective function

```
def SafetyAwareLoss(I):
    column_sum = I.sum(axis=0)
    target_range = 0.05 * max_power_limit
    reward = np.sum(-np.abs(column_sum - max_power_limit) + max_power_limit)
    excess_power = np.maximum(0, column_sum - max_power_limit)
    penalty = np.sum(excess_power ** 2)
    fitness_value = reward - penalty
    return fitness_value
```

Listing 5: The python code of safety-aware objective function