A Novel Online Scheduling Algorithm and Hierarchical Protocol for Large-Scale EV Charging Coordination

SHAOBING YANG, (IEEE Member), SHANSHAN ZHANG, (IEEE Student Member), JINGJING YE
School of Electrical Engineering, Beijing Jiaotong University, Beijing, 100044, China
Corresponding author: Jingjing Ye (e-mail: jjye@bjtu.edu.cn).
This work was supported by the Fundamental Research Funds for the Central Universities (2018JBZ101).

ABSTRACT For electric vehicle (EV) charging coordination, relaxation and approximation methods are usually employed to improve scheduling efficiency in solving integer linear programming (ILP) problems. Nevertheless, they have a few inherent flaws such as frequent charging interruption and large deviation in final SOC. Based on the recursive method, this paper proposes a search-swapping algorithm (SSA) for optimal approximation. It leverages the non-aftereffect property of charging scheduling to approximate the optimal solution, as well as uses single recursion instead of multiple recursion to reduce computation cost from exponential time to polynomial time. Compared with the conventional algorithms, it has much higher efficiency, less charging interruption, and better final SOC uniformity. Furthermore, a single-round interactive protocol is designed for hierarchical coordinated framework to achieve power dispatch. Numerous simulations show that the SSA has outstanding efficiency and satisfactory optimality.

INDEX TERMS Charging coordination, electric vehicle (EV), hierarchical protocol, integer linear programming (ILP), recursive method.

I. INTRODUCTION

A. MOTIVATION
Electric vehicles (EVs) are expected to play an important role in road transport systems since they promise for higher energy efficiency and lower emission. For the past few years, EVs have been developing rapidly due to incentives for new energy vehicles as well as restrictions on emissions, sales and licensing of internal combustion engine (ICE) vehicles. In 2018, more than one million EVs were sold in China. Most of them are used for commuting by urban residents. However, EV charging load was not taken into account in the capacity configuration of distribution power systems. Uncoordinated charging loads will severely threaten the operations of power distribution networks [1, 2].

Due to economy and convenience, EV owners prefer to choose slow charging overnight by using a wide variety of private chargers [3, 4], most of which do not support the variable-rate control. Hence, the discrete charging method is expected to be dominating in the near future and will co-exist with the continuous charging method in the long run [5]. In this case, the coordination can be formulated as an integer linear programming (ILP) problem in a minimization or maximization fashion, which is classified as NP-hard and can be coped via relaxation and approximation methods [6]–[10]. Nevertheless, relaxation-based algorithms have some flaws that should not be ignored, especially when used in the heavy-demand scenario (i.e., power capacity cannot fully meet charging demands). For instance, those conventional algorithms generally result in frequent charging interruption and non-uniformity of final state-of-charges (SOCs). Apart from that, the delicate algorithms, such as branch and bound (B&B) and branch and cut (B&C) [11], take exponential time to solve an ILP problem. In real applications, the flaws will reduce the efficiency of coordination as well as impair the fairness of charging service.

This paper aims to develop a centralized base-level online scheduling algorithm to reduce computational cost, improve final SOC uniformity, and suppress charging interruption, as well as to design an interactive protocol via a single round of negotiation to improve the overall efficiency of large-scale hierarchical coordination.

B. LITERATURE REVIEW
Relaxation-based linear programming (LP) approaches have
been widely utilized for charging coordination [12]–[19]. In many studies, the discrete-charging decision is described by an integer/binary number, thus the scheduling is converted to an ILP problem. The commercial solver CPLEX [20] is usually employed for such scheduling problem since it is regarded as the most successful ILP solver to date. For real-time charging scheduling, the algorithm needs to achieve a delicate balance between computational cost and optimality.

In order to improve efficiency, modified relaxation methods have been developed for EV coordination. Yao et al. [9] proposed a two-stage convex relaxation approach to tackle the solving difficulty: first obtaining the real-valued optimal scheme, and then rounding each decision to binary one, and finally solving the small-scale ILP problem for each step with the power constraints. Jin et al. [14] proposed a simple heuristic algorithm through LP rounding, which can give near-optimal solutions in polynomial time. Logically, when using the real-valued LP, excessive relaxation will result in a large drop in optimality.

In addition to high computation cost, charging interruption is another issue that has attracted a lot of attention. Sun [5] et al. pointed out that frequent charging interruption will lead to additional deterioration of batteries as well as threaten the operation stability of distribution systems, thus they set a constraint in the objectives to smooth the charging profile. However, such constraint increases the modeling complexity. In [21], the authors introduced a penalty parameter into the cost function to prevent frequent on-off switching. Evidently, adding extra constraints may interfere with the achievement of other major optimization goal such as cost saving.

The equalization of final SOC is deemed as an important performance index of coordinated algorithms, especially in heavy-demand scenarios. A large deviation in the final SOC means unfairness in meeting charging requirements. When the power capacity is limited, conventional ILP algorithms tend to preferentially charge a portion of EVs, since they use the divide-and-conquer method for dimensionality reduction. To overcome this drawback, a lower bound is usually put into the objective function [9]. That is to say, every EV must be charged to the lower bound before leaving. Definitely, a strict bound may cause solving failures. Another way is set to prioritize lower-SOC vehicles in scheduling. In [22], a completeness factor was used to motivate the algorithms to charge lower-SOC EVs first. Nevertheless, using such factor might impair the cost-saving goal.

Due to excellent scalability and flexibility, hierarchical coordination is regarded as the most promising option for near-term realization of EV coordinated charging by both academia and industry [10]. In theory, its performance relies on the efficiency of base-level algorithms and the overall overhead of inter-level communication. Since the interests of a charging operator is related to power dispatch and local constraints, a multiple-round negotiation is usually required to converge dispatching to a balance point. In [23], the base level strictly follows the dispatch instructions from the upper level, where the power dispatch needs iterative adjustments. Similarly, Qi et al. [24] proposed a two-level coordinated framework, where the iteration is used to ensure the centrally feasible of solutions. In [25], the dispatch from the upper controller is as a loose reference for the base level, avoiding iterative optimization. In large-scale charging coordination, the number of inter-level interactions will greatly affect the overall efficiency. Accordingly, reducing the interactions is very meaningful.

C. CONTRIBUTION OF THIS PAPER

To improve service capability and profitability, the basic targets of a charging operator usually include avoiding local power overload, maximizing power capacity utilization, and minimizing electricity bill [10]. According to the objectives, this paper formulates the base-level coordination as 0-1 ILP optimization in a centralized fashion, where each decision is described by a binary number and denotes whether an EV is allowed to charge at a time step. Based on recursive method, a novel search-swapping algorithm (SSA) is proposed for real-time coordination, which is essentially different from conventional ILP algorithms. Besides, a single-round inter-level protocol is developed based on the SSA to achieve reasonable power dispatch between coordination levels. The contributions and technical novelty are as follows:

1) To minimize charging cost or maximize power capacity utilization, the SSA utilizes single recursion to find a route to transfer a charging task to a more appropriate time step. In each search, the SSA only chooses one node (i.e., one binary decision) for branching and employs two taboo lists to ensure that each node only be accessed once. Accordingly, the SSA has a dominant advantage in terms of computational cost.

2) In the scheduling, the SSA processes EVs one by one, and rearranges the charging decisions one by one. Hence, it moves a whole charging process to an appropriate period as completely as possible, minimizing charging interruptions.

3) In order to minimize electricity bill, the SSA postpones high-priced charging tasks to low-priced period. It is set to do this in a SOC descending order. That is, it will postpone high-SOC EVs first. When handling local power violations, the SSA also suspends the charging for high-SOC EVs first. These techniques will improve the uniformity of final SOC.

4) The designed inter-level protocol can be used for power dispatch through a single round of negotiation. For large-scale coordination, it promises higher efficiency as well as lower communication overhead.

The rest of this paper is as follows: Section II illustrates a bi-level hierarchical framework for centralized coordination. Section III introduces the design of SSA in detail. In Section IV, a single-round inter-level protocol is designed for power dispatch. Numerous simulations are implemented, verifying
the SSA and the protocol in Section V and VI, respectively, followed by the conclusion in Section VII.

II. HIERARCHICAL COORDINATED FRAMEWORK

Fig. 1 illustrates a bi-level coordinated framework, which has two kinds of controllers, i.e., the upper controller (UC) and the station controller (SC). The power supply may serve regular load and EV charging. The parking operator uses SC for local scheduling, while the aggregator or the grid operator uses UC for power dispatching. Evidently, each UC/SC has an individual power capacity threshold, taking into account the limits of transformers and cables, voltage deviations, network loss, etc. Remaining power capacity for charging can be estimated based on load prediction.

**FIGURE 1. Schematic illustration of a bi-level coordinated system.**

The power dispatching steps are as follows:
1) At the current step, SCs obtain the optimal schemes by the SSA according to local power limits, electricity prices, and the knowledge of participating EVs. Each SC then sends its charging demand and its demand transfer capability (DTC) to the UC, where the DTC refers to the charging demand that can be transferred to future (instead of being abandoned) if charging is not allowed.
2) After collecting charging demands and DTCs from all SCs, the UC makes dispatching decisions based on its rest capacity (i.e., deducting regular power demand), and then sends the dispatching instructions to the SCs separately.
3) After receiving the dispatch, a SC must update its local power limit and re-optimizes its charging scheme by the SSA to make it globally feasible. At last, the SC performs the final scheme by sending on/off instructions to its chargers.

The communication overhead between a SC and a charger, including EV arrival and departure times, battery capacity, initial and expected SOCs, charging rates, on/off instructions, etc., can be described by only hundreds of bytes. In this case, a low-cost LAN (e.g., RS485 or CAN) is able to meet the communication requirements, since the scheduling interval is typically more than ten minutes.

The overall efficiency of a multilevel coordination system depends on the scheduling time cost of the base level and the power-dispatching time cost of the upper levels. This paper focuses on the base-level centralized algorithm and the inter-level protocol. For the purpose of independent comparisons, the early-charging strategy is not leveraged to cope with the uncertainty of EV mobility behaviors.

III. STATION-LEVEL COORDINATED ALGORITHMS

A. PROBLEM FORMULATION

The SC aims to find an optimal charging scheme in a specific time period. This scheme not only has to satisfy the needs of EV owners, but also must avoid power violations. In order to find the optimal scheme, the following considerations are assumed:
1) When an EV is plugged in, its initial SOC is known.
2) The SC can obtain the forecasts and real-time values of base load, where the real-time value is for overload control.
3) The driver may provide a departure time when an EV is plugged in; otherwise, the SC will estimate the time based on the past mobility behaviors of this EV.

The basic targets for station-level charging coordination include maximizing charging completion rate (CCR, i.e., the ratio of charging consumption to demand) and minimizing electricity bill. For the purpose of CCR maximization, the objective function is defined as follows [8]–[10]:

$$\text{maximize} \sum_{j=1}^{J} \left[ u_j \sum_{n \in \psi(j)} (p_n s_n^j) \right], \ k = 1, \ldots, J \quad (1a)$$

subject to

$$\sum_{n \in \psi(j)} (p_n s_n^j) \leq P_{max}^j, \ j = 1, \ldots, J, \quad (1b)$$

$$\gamma_n^k + \sum_{j=k}^{J} \frac{\eta_n p_n s_n^j T_j}{E_n} \leq \gamma_{exp,n}, \ \forall n \in \psi^k \quad (1c)$$

$$\gamma_n^k > \gamma_{min,n}, \ \forall n \in \psi^k \quad (1d)$$

$$u_j = \frac{\theta_{max} - \theta_{min}}{\theta_{max} - \theta_{min}}, \ \forall n \in \psi^k \quad (1e)$$

where $j$ and $n$ respectively denote a time step and an EV, $E_n$, $p_n$ and $h_n$ are the battery capacity in kWh, charging rate in kW, and charging efficiency lying in $[0, 1]$ of EV $n$, $T_j$ is the fixed time interval, $s_n^j$ is the binary decision (i.e., 1: on and 0: off), $\gamma_n^k$, $\gamma_{exp,n}$, and $\gamma_{min,n}$ stand for the current, the expected, and the lower SOC bound of EV $n$, $\theta^j$ and $P_{max}^j$ are the price and the power limit of time step $j$, $\psi^k$ denotes the set of participating EVs at time step $k$, $\theta_{max}$ and $\theta_{min}$ stand for the highest and the lowest prices, respectively, $\omega^j$ is the price preference that returns a higher value for a lower price and vice versa.

The scheduling of (1) is a 0-1 ILP problem, which can be solved by conventional LP algorithms. The solution is denoted as a binary decision matrix $\mathbf{S}$, and only the current decisions, $s_n^k$, $\forall n \in \psi^k$, will be actually performed. Instead of solving the problem of (1), this paper develops a search-swapping algorithm based on the recursive method to obtain near-optimal solutions, where the objectives will be separately achieved in the following stages:

1) Initialization: by the first come first serve (FCFS) strategy, initialize $\mathbf{S}$ without considering power limits
2) Cost minimization: by the recursive method, move high-priced decisions to low-priced periods to minimize charging cost.

3) CCR maximization: by the recursive search, transfer the decisions violating power limits to the time steps that have available power capacity.

4) Overload control: by a strategy such as the largest SOC first, turn off on-decisions for each power-violating time step until meeting the power limit.

The first three stages can be performed at the beginning of each time step, while the last stage can be repeated at a short time interval such as a few seconds to handle overload in time. At current time step $k$, the decision matrix is denoted as

$$
\mathbf{S}(k) = \begin{bmatrix}
s_1^k & s_1^{k+1} & \ldots & s_1^J \\
s_2^k & s_2^{k+1} & \ldots & s_2^J \\
\vdots & \vdots & \ddots & \vdots \\
s_N^k & s_N^{k+1} & \ldots & s_N^J
\end{bmatrix}
$$

where $N^k$ denotes the total number of EVs at the current step.

The initializing process with the FCFS strategy is denoted as

$$
\begin{align*}
& s^j_i = 1 \quad P_{\text{max}}^j \leq P_{\text{max}}^n, \forall n \in \psi^j, \forall j \in [k, k_{n_f}^j] \\
& k_{n_f}^j = \min \left( k_{\text{def}}^n, k + \left[ \frac{E_n(r_n - p_n) - E_n(r_n - p_n)}{p_n \eta_n} \right] \right) - 1
\end{align*}
$$

where $k_{\text{def}}^n$ denotes the time step in which EV $n$ will leave.

One example of initial solutions, including 6 EVs and 10 time steps, is illustrated in Fig. 2, where the time step marked by a cross is outside of the parking time range. Evidently, this scheme may cause power violations and high cost, thus it needs to be optimized in the next steps.

B. BASIC IDEA OF THE SSA
At the beginning of current time step, the FCFS solution is generated with (3). After that, decision swapping is used to minimize cost or maximize CCR. In Fig. 3(a), in order to reduce the cost, $s_1^k$ can be postponed to a low-priced time step that can be found by a horizontal search (HS):

$$
\begin{align*}
& s^j_i = 0 \quad j = k + 1, \ldots, k + 8 \\
& \theta_i < \theta^k \quad p_1 \leq P_{\text{max}}^1
\end{align*}
$$

When the power capacity of each low-priced period is not limited, each EV can transfer its high-priced on-decisions. In this case, the scheduling problem has the non-aftereffect property, i.e., Markov property, thus the HS can find a global optimal solution. On the contrary, when the power capacity is limited, a vertical search (VS) can be used to approximate the optimal solution. As shown in Fig. 3(b), the capacities at both $k + 5$ and $k + 6$ are limited, thus the HS cannot find a route to swap $s_6^k$, while the VS denoted by the green arrow can find a feasible route, i.e., $s_4^{k+5} \Rightarrow s_4^{k+9}$ and $s_6^k \Rightarrow s_6^{k+5}$.

By the combination of HS and VS, the search scope can be significantly expanded, as illustrated in Fig. 4.

C. DESIGN OF THE SSA
For power-violating on-decisions, HS&VS will try to find a swapping route to transfer them, instead of turning them off, to maximize CCR. While for reducing charging cost, a route

FIGURE 2. Schematic illustration of the FCFS solution.

FIGURE 3. Methodology of the SSA: (a) HS and (b) HS&VS.

FIGURE 4. Schematic illustration of a HS&VS swapping route.
for swapping must be profitable. Define the root node as $s^y_x$. The steps of cost-saving optimization are as follows:

1) Define $\alpha$ as the swapping profit, and initialize it to zero.
2) Use the HS to traverse all decisions of EV $x$ for an off-decision that meets

$$\begin{align*}
    s^j_y &= 0 \\
    p_x &\leq P^j_{\text{max}}, \quad j = k, \ldots, k^f_x \\
    \alpha + p_x(\theta^y - \theta^j) &> 0
\end{align*}$$

If find and the profit is positive (i.e.,), swap $s^y_x$ and $s^j_x$, and return true; otherwise, go to the next step.

3) Use the HS to look for an off-decision (i.e.,, $s^j_x = 0$, $j = k, \ldots, k^f_x$). If not find, return false; otherwise, use the VS to look for an on-decision:

$$\begin{align*}
    s^j_n &= 1 \\
    P^j_{\text{max}} + p_n &\geq p_x
\end{align*}$$

If find, let $s^o_n$ as the child node to create a subbranch:

$$\begin{align*}
    x &\leftarrow n \\
    y &\leftarrow j \\
    \alpha &\leftarrow \alpha + p_x(\theta^y - \theta^j), \quad \forall n \in \psi^k
\end{align*}$$

Go to Step 2 to start a new HS of (5); otherwise, resume the previous HS of Step 3 until $j$ equals $k^f_x$.

Define $\mathcal{F}^+$ and $\mathcal{F}^-$ as two taboo lists to respectively record traversed EVs and time steps. Define $\text{add}(s^y_x, R)$ as the operator to add a node to the route of $R$. Define $\text{ins}(\mathcal{L}, i)$ as the operator to insert one item into the list $\mathcal{L}$. Define a flag $\lambda$ to indicate the scheduling purpose: cost minimizing ($\lambda = 0$) or load shifting ($\lambda = 1$). The recursive function, $\Lambda(s^y_x, \alpha, \lambda)$, is formally written as Algorithm 1.

Using the taboo lists, one node can only be accessed once, hence the time complexity of Algorithm 1 should be $O(JN)$. To be specific, since the total number of time steps, $J$, is much larger than the number of time steps for parking, the time cost of Algorithm 1 satisfies:

$$\tau_A(J, N) < JN[\Delta + \tau_A(J - 1, N - 1)]$$

where $\Delta$ denotes the time cost of Line 10 in Algorithm 1, and $N$ is the number of EVs. Obviously, (8a) satisfies:

$$\tau_A(JN) < JN \cdot \tau_A(JN - 1)$$

Accordingly, the time complexity of Algorithm 1 is $O(JN)$. For Stage 2 of Algorithm 2, due to the price constraint ($\theta^j < \alpha$), one node can only be swapped less than $J$ times. Its time complexity is $O(J^3N^2)$. Thus, SSA is a polynomial-time algorithm. In fact, its time cost is far less than the maximum boundary of (8). First, the number of participating time steps is much smaller than $J$ since the parking duration is much shorter than the scheduling period. Second, the HS with less time cost, instead of HS&VS, will find most feasible routes. On a personal computer that spends 15s computing the math constant $\pi$ to one million digits using SuperPI 1.9, SSA takes less than 0.3s on the coordination of $J$ equals 96 (i.e., 15min interval) and $N$ equals 1000.

Algorithm 2: SSA($k$), where $k$ stands for current time step.

1: Stage 1: Initialize the FCFS solution
2: Stage 2: minimize charging cost
3: $\nabla(k)$;
4: $\text{clr}(\mathcal{F}^+) \cap \text{clr}(\mathcal{F}^-); \beta := 0$;
5: for each $x$ in $\psi^k$
6: for each $y \in [k, k^f]$ where $s^y_x = 1$
7: $\text{clr}(R)$;
8: if $\Lambda(s^y_x, \theta^y, 0)$
9: $\text{sup}(s^y_x, R)$;
10: $\beta := 1$;
11: if ($\beta = 1$) goto Line 4
12: Stage 3: maximize CCR
13: $\text{clr}(\mathcal{F}^+) \cap \text{clr}(\mathcal{F}^-); \beta := 0$;
14: for each $y \in [k, J]$ where $P^y_{\text{max}} < 0$
15: for each $x$ in $\psi^k$ where $s^y_x = 1$
16: if $\Lambda(s^y_x, \theta^y, 1)$
17: $\text{sup}(s^y_x, R)$;
18: $\beta := 1$;
19: if ($\beta = 1$) goto Line 13
20: Stage 4: overload control
21: $\delta^k := 0$;
22: while $P^y_{\text{max}} < 0$ do
23: $i := \nabla(k)$;
24: $\delta^k := 0$;
25: $P^y_{\text{max}} := P^y_{\text{max}} + p_i$;
26: $\delta^k := \delta^k + p_i$;

It should be noted that Algorithm 1 utilizes single recursion rather than multiple recursion. The single recursion requires
only one on-decision for swapping, i.e., to select a single node for branching, while multiple recursion can combine multiple on-decisions to provide the required power capacity for the parent node. Accordingly, multiple recursion can give a tree-structure map instead of a single route. To achieve multiple recursion, the constraints of (6) shall be modified to

\[
\begin{align*}
\mathcal{S}_n &= 1 \\
\mathcal{P}_{\text{max}} + \sum_{y_{n} \in \phi} \mathcal{P}_n &\geq \mathcal{P}_x, \quad \phi \in \psi^k
\end{align*}
\]  

(9)

where \( \phi \) is a subset of \( \psi^k \).

According to (9), the optimization is a knapsack problem, classified as NP-hard, which will take an exponential time to program multiple charging rates. Consequently, the diversity of charging rates is one of the critical factors that lead to high computation cost. In fact, multiple recursion can only show its advantage when the remaining capacity is very small, less than most charging rates. Consequently, using (6) instead of (9) will only result in slight loss of optimality but can significantly reduce computation cost.

Reducing charging interruption can improve the reliability of coordinated systems as well as prolong battery lifespan. For conventional algorithms, however, achieving this goal without impairing the optimality is very challenging. In Stage 2 of Algorithm 2, the SSA schedules the charging tasks in the way of one EV after another. As a result, the SSA prefers to transfer a whole charging process for an EV with the least number of interruptions. Moreover, the SSA can transfer the high-SOC EVs first since it sorts the EVs in descending order of SOC, as shown in Stage 2. It also suspends the high-SOC EVs first, as shown in Stage 4. Consequently, the difference between final SOCs can be significantly reduced.

### D. OPTIMALITY ANALYSIS

According to (1) and (2), the scheduling problem has the same structure at each time step, and its only state variable is SOC. Therefore, this problem has non-aftereffect property, which can be solved by dynamic programming (DP) [26]. The SSA runs in the DP fashion through a recursive method. In theory, it could obtain the global optimal solution, however selecting single recursion rather than multiple recursion will result in a tiny loss of optimality.

When power capacity is larger than charging demand at any time step, the HS can transfer an on-decision to any time step, and its only state variable is SOC. Otherwise, the scheduling needs to program the proper charging sequence for each EV. When EVs have the same charging rate and the same departure time, the SSA can obtain the global optimal solution, as proved in Appendix.

According to Algorithm 1, the SSA sorts EVs in the SOC descending order before scheduling. Each SOC will randomly increases from its random initial value. Thus, the selection of branching nodes is strongly random. Since the departure time is widely various and independently random, its diversity will not affect the optimality significantly when a large number of EVs participate. For convenience of analysis, the scheduling can be approximated to the specific case where EVs have the same departure time. In this case, the charging-rate diversity is the only factor that affects optimality. By programming various charging rates, the LP algorithm can maximize the capacity utilization. For instance, it can combine 2kW and 3kW charging tasks, to fully use the remaining capacity of 5kW. As for the SSA, using single recursion might leave more capacity unused, impairing optimality. For instance, when the remaining capacity is 5kW, it randomly arranges a 4kW task, which may reduce power capacity utilization. Logically, the unutilized capacity is less than the minimum rate since the SSA can transfer any on-decision to any time step due to the same parking time range. Even if the departure diversity has an effect due to the limited number of EVs, the unutilized capacity is usually less than the maximum rate. Therefore, the optimality gap of SSA and LPs is very small.

Compared with the FCFS, the HS greatly improves CCR as well as reduce charging cost. This provides a basic guarantee of optimality. Furthermore, the combination of HS and VS can expand the search scope to approximate the optimal solution. Although using random selection rather than programming the various charging rates may cause a slight loss of optimality, the computational cost can be greatly reduced.

### IV. INTER-LEVEL COORDINATED PROTOCOL

Hierarchical charging coordination should realize the equity of power capacity allocation. In addition to the station level, the power dispatch should take into account the uniformity of final SOC. Thus, the SC with a larger DTC should take more responsibility for curtailing charging demand.

For the charging operator \( i \), define \( P_{\text{de},i} \) as the charging demand under the local power limit \( \mathcal{P}_{\text{max},i} \). Define \( P_{\text{ab},i} \) as the abandoned demand of when the initial \( P_{\text{max},i} \) equals zero (i.e., no charging is allowed). \( P_{\text{ab},i} \) is the portion that cannot be shifted to other time steps, which can be computed by \( \sum \delta \) as shown in Algorithm 2. Define \( P_{\text{dte},i} \) as the DTC. For the upper level, total charging demand and DTC are denoted as

\[
\begin{align*}
\mathcal{P}_{\text{de},\Sigma} &= \sum_{i=1}^{L} \mathcal{P}_{\text{de},i} \\
\mathcal{P}_{\text{dte},\Sigma} &= \sum_{i=1}^{L} \max(\mathcal{P}_{\text{de},i} - P_{\text{ab},i}, 0)
\end{align*}
\]  

(10)

where \( L \) denotes the total number of charging operators.

Define \( P_{\text{av}} \) as the upper-level available capacity, which can be computed according to base load, voltage deviation, transmission threshold, and DR requests. When satisfying \( P_{\text{av}} \geq \mathcal{P}_{\text{de},\Sigma} \), the optimal schemes submitted by all charging operators will be approved without modifications. Otherwise, the charging demand curtailment is

\[
\mathcal{P}_{\text{ct}} = \mathcal{P}_{\text{de},\Sigma} - P_{\text{av}}, \quad (i\text{f} \quad P_{\text{av}} < \mathcal{P}_{\text{de},\Sigma})
\]  

(11)

The demand curtailment of charging operator \( i \) is computed by

\[
\mathcal{P}_{\text{ct},i} = \begin{cases} 
\mathcal{P}_{\text{dte},\Sigma} \cdot \mathcal{P}_{\text{ct}}, & (i\text{f} \quad \mathcal{P}_{\text{ct}} \leq \mathcal{P}_{\text{dte},\Sigma}) \\
\mathcal{P}_{\text{dte},\Sigma} \cdot \mathcal{P}_{\text{ct}}, & (i\text{f} \quad \mathcal{P}_{\text{ct}} > \mathcal{P}_{\text{dte},\Sigma})
\end{cases}
\]  

(12)

\[\text{This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/}\]
Last, the upper controller sends $P_{ct,i}$ to charging operator $i$. Once receiving $P_{ct,i}$, the SC adjusts the local power constraint to

$$P_{\text{max},i} = P_{\text{de},i} - P_{ct,i} \quad (13)$$

The proposed protocol of (10) – (13) can be achieved by only one round interaction, as shown in Fig. 5. The interaction requires only three parameters therefore the communication overhead is very small. The final scheme can fully meet the power constraints of upper and station levels. Although SSA runs two or three times at each time step, the overall time cost of real-time hierarchical coordination is fully affordable by using the efficient SSA.

![FIGURE 5. One-round negotiation between upper and station levels.](image)

The illustrated bi-level system is a base module. In this way, a multilevel tree-structure framework can be built to cope with large-scale EV charging, as shown in Fig. 6. The charging load will be aggregated via a bottom-up method (i.e., from base level to top level), while the power dispatch is achieved via a top-down method. Only the station controllers need to run the scheduling algorithm a few times, as well as need to know the detailed information of EVs. Due to low communication overhead, the charging coordination can be implemented via the Internet even mobile Internet.

![FIGURE 6. Illustration of the tree-structure framework for large-scale charging coordination.](image)

### V. STATION-LEVEL ALGORITHM VERIFICATION

#### A. SIMULATION SETTINGS

Take a residential charging station as an example. The total number of participating EVs during 24 hours, $M$, is set to 50, 100, 150, 200, 250, and 300. Table I shows the settings of EV mobility behaviors, while Table II presents the specifications, where $\mathcal{N}$ and $\mathcal{U}$ denote the normal and uniform distributions. As shown in Fig. 7, the power limits and electricity prices are respectively set based on the actual load from RTE-France [27] and the day-ahead trading records on the EPEX-SPOT [28]. Private vehicles usually arrive home in afternoon and leave in morning, hence the time range is set to [12:00 pm, 12:00 am]. The scheduling time interval is set to 15 min, i.e., the algorithm runs 96 times during a day.

![TABLE I THE SETTINGS OF EV MOBILITY PATTERNS](image)

<table>
<thead>
<tr>
<th>Group</th>
<th>Arrival</th>
<th>Departure</th>
<th>Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mathcal{N}(18, 2^2)$</td>
<td>$\mathcal{N}(6, 1^2)$</td>
<td>70%</td>
</tr>
<tr>
<td>2</td>
<td>$\mathcal{U}(12, 24)$</td>
<td>$\mathcal{U}(0, 12)$</td>
<td>30%</td>
</tr>
</tbody>
</table>

![TABLE II EV SPECIFICATIONS AND PROPORTIONS](image)

<table>
<thead>
<tr>
<th>Type</th>
<th>$E_{\text{cap}}$ [kWh]</th>
<th>$p$ [kW]</th>
<th>$\gamma^0$</th>
<th>$\gamma^{\text{max}}$</th>
<th>$\eta$</th>
<th>Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td>3.3</td>
<td>$\mathcal{U}(0.1, 0.6)$</td>
<td>1.00</td>
<td>0.9</td>
<td>50%</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>6.0</td>
<td>$\mathcal{U}(0.1, 0.6)$</td>
<td>1.00</td>
<td>0.9</td>
<td>30%</td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>9.0</td>
<td>$\mathcal{U}(0.1, 0.6)$</td>
<td>1.00</td>
<td>0.9</td>
<td>20%</td>
</tr>
</tbody>
</table>

Five algorithms are realized through Visual Studio 2013 for comparisons: FCFS, SSA, B&C without the limit of $\gamma_{\text{min}}$, B&C with $\gamma_{\text{min}}$ of 0.6 (B&C-L), and the two-stage convex relaxation method (LP-R) [9], where the LP algorithms use the objective function of (1) and call the library function of CPLEX 12.6 for implementation. In addition, FCFS and SSA adopt the overload control of Stage 4 in Algorithm 2. A personal computer with Intel Core i7-6700 CPU (3.4 GHz) and 8GB RAM is used to run the programs. For each scenario of $M$, the algorithms will run 10 rounds (days).

Notably, CPLEX’s gap tolerance and time limit will affect precision and computational cost. To be specific, a small tolerance can improve precision however lead to high time cost. In the following simulations, the gap tolerance and time limit are set to 0.0005 and 60s, respectively.

![FIGURE 7. Power limits and electricity prices for simulations.](image)

#### B. OPTIMALITY TEST

Competitive ratio (CR) is usually employed to evaluate the optimality and the boundary of the worst-case performance [29]. The offline B&C, which knows the full information of EV behaviors, is as the benchmark. For each algorithm, the CR is computed by
where $f$ is the objective function of (1), and $S$ and $S_0$ stand for the online and offline schemes, respectively. That is, the closer the CR is to 1, the better the optimality is.

Table III gives the simulation results of average as well as the worst cases. Overall, two B&C algorithms show the best performance, while the optimality gap of SSA to B&C is less than 2%. In the 200-EV scenario, the B&C algorithms output more than 20% of non-optimal (feasible) solutions therefore the optimality slightly decreases. As for LP-R, the optimality greatly decreases in the heavy-demand scenarios, since it does not schedule charging from an overall perspective. In the other words, its two separate stages cannot interwork to deal with heavy charging demand.

### TABLE III COMPARISON OF COMPETITION RATIO RESULTS

<table>
<thead>
<tr>
<th>$M$</th>
<th>B&amp;C</th>
<th>B&amp;C-L</th>
<th>LP-R</th>
<th>SSA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg. worst</td>
<td>avg. worst</td>
<td>avg. worst</td>
<td>avg. worst</td>
</tr>
<tr>
<td>50</td>
<td>1.00 0.94</td>
<td>1.00 0.94</td>
<td>1.00 0.94</td>
<td>1.00 0.94</td>
</tr>
<tr>
<td>100</td>
<td>1.00 0.97</td>
<td>1.00 0.97</td>
<td>0.99 0.96</td>
<td>0.99 0.97</td>
</tr>
<tr>
<td>150</td>
<td>0.99 0.98</td>
<td>0.99 0.98</td>
<td>0.98 0.96</td>
<td>0.99 0.97</td>
</tr>
<tr>
<td>200</td>
<td>0.92 0.91</td>
<td>0.92 0.91</td>
<td>0.84 0.19</td>
<td>0.93 0.92</td>
</tr>
<tr>
<td>250</td>
<td>0.90 0.89</td>
<td>0.90 0.89</td>
<td>0.83 0.36</td>
<td>0.90 0.89</td>
</tr>
<tr>
<td>300</td>
<td>0.90 0.89</td>
<td>0.90 0.89</td>
<td>0.72 0.28</td>
<td>0.90 0.89</td>
</tr>
</tbody>
</table>

For the purpose of error analysis, Fig. 8 shows the charging load profiles in the 300-EV scenario. In the low-priced range of [21:00 pm, 7:00 am], the charging load profiles of SSA and B&C almost coincide. The three maximum gaps, occurred at 1:00 am, 3:15 am and 6:00 am, are 8.0kW, 5.7kW and 5.7kW, respectively, which are less than the maximum rate of 9kW.

In order to comprehensively verify the SSA’s optimality, the 31-day load data of October, 2017 is extracted from RTE-France and converted by a ratio to fit a distribution system with the capacity threshold of 800kW, as shown in Fig. 9. The load data is used as regular power to obtain power limits. The day-ahead electricity prices are from EPEX-SPOT, as shown in Fig. 10. Other settings remain the same. Simulation results of the 300-EV scenario are shown in Fig. 11, where the largest gap of the SSA to B&C is 1.30% on the 29th.

Simulations show that the time cost of B&C will take much more time to program all charging rates to obtain a delicate scheme, thus it may output more non-optimal solutions. In this case, the SSA performs better, such as the first and the last two days.

In theory, under the same scheduling goals, the inaccurate predictions of base load, electricity price, and power capacity should affect all algorithms in a similar way. To be specific, the effects on optimality, caused by the forecast errors and the EV mobility uncertainty, are very complex. For instance, the unexpected change in price will cause a variation in optimality, and the better the algorithm performs, the greater the variation is. To tackle these uncertainties, a short time interval must be set to respond quickly to random changes in the future information. In this regard, the efficient SSA has outstanding advantages.
C. FINAL SOC TEST

Table IV presents the final SOC results. In all scenarios, SSA shows the best performance. In the heavy-demand scenarios, it improves the average final SOC by up to 0.04. Apart from that, SSA can improve the uniformity of final SOC.

Fig. 12 shows the histograms of final SOC in the 300-EV scenario. For B&C, the final SOC is widely distributed within the range of [0.2, 1.0] and the probability peak is at the SOC of 1.0. For B&C-L, the final SOC is within the range of [0.6, 1.0] and the peak is at the SOC of 0.64. While for SSA, the peak is at the SOC of 0.80 and the final SOC approximates a triangular distribution within the range of [0.7, 1.0]. It shows better performance in terms of final-SOC equalization.

Raising the lower bound, $\gamma_{\text{min}}$, can narrow the final-SOC range for B&C-L. However, it may cause the solving failure. For instance, when the parking duration of an EV is too short to be fully charged, its final SOC cannot reach the lower bound. This often occurs in non-overnight charging. In order to find a proper bound, the scheduling must be performed iteratively at each time step. It is less practical for real-time coordination. The standard deviation is computed to evaluate the overall final-SOC equalization. As can be seen from Table V, SSA presents the best results. It can reduce the deviation by up to 36% in the heavy-demand scenarios.

D. COMPUTATIONAL COST

Stage 2 of Algorithm 2 is performed one hundred times to analyze SSA’s convergence property. As shown in Fig. 13(a), the optimization can be done in less than 20 goto-loops, and the number of the swaps drops rapidly. Fig. 13(b) shows the probability histogram of swaps. It can be seen that most optimizations can be completed in less than 1,000 swaps. Accordingly, SSA’s convergence rate is excellent.

Raising the lower bound, $\gamma_{\text{min}}$, can narrow the final-SOC range for B&C-L. However, it may cause the solving failure. For instance, when the parking duration of an EV is too short to be fully charged, its final SOC cannot reach the lower bound. This often occurs in non-overnight charging. In order to find a proper bound, the scheduling must be performed iteratively at each time step. It is less practical for real-time coordination. The standard deviation is computed to evaluate the overall final-SOC equalization. As can be seen from Table V, SSA presents the best results. It can reduce the deviation by up to 36% in the heavy-demand scenarios.

![Figure 12](image_url)

**FIGURE 12.** Comparison of probability distributions of the final SOCs in the 300-EV case.

![Figure 13](image_url)

**FIGURE 13.** Convergence of SSA in Stage 2 of Algorithm 2 for the 300-EV scenario.
algorithms take exponential time. Relaxing the gap tolerance to 0.001 can improve the ratio to about 98% however lead to worse optimality than SSA.

LP-R still needs to solve a small-size ILP problem in the second stage. As the number of participating EVs increases, the time cost of the second stage increases. Thus, the total time cost is instead higher than the delicate algorithms.

By contrast, SSA can provide near-optimal solutions in less than 0.1 s in all scenarios. The outstanding advantage in efficiency makes it possible to realize large-scale coordination on low-cost microcontrollers.

### E. CHARGING INTERRUPTION

In practice, charging interruptions lead to battery degradation and cause the failures of charging devices. Consequently, reducing the charging interruptions is very meaningful in reality. Table VII shows the results of charging interruption. Compared with B&C, SSA can reduce the interruptions by more than 40% in heavy-demand scenarios.

<table>
<thead>
<tr>
<th>( M )</th>
<th>B&amp;C</th>
<th>B&amp;C-L</th>
<th>LP-R</th>
<th>SSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.54</td>
<td>0.54</td>
<td>0.51</td>
<td>0.10</td>
</tr>
<tr>
<td>100</td>
<td>2.06</td>
<td>2.06</td>
<td>1.38</td>
<td>0.97</td>
</tr>
<tr>
<td>150</td>
<td>5.53</td>
<td>5.59</td>
<td>5.06</td>
<td>3.24</td>
</tr>
<tr>
<td>200</td>
<td>6.02</td>
<td>6.06</td>
<td>5.14</td>
<td>3.40</td>
</tr>
<tr>
<td>250</td>
<td>5.65</td>
<td>5.84</td>
<td>5.15</td>
<td>3.36</td>
</tr>
<tr>
<td>300</td>
<td>5.32</td>
<td>5.32</td>
<td>3.64</td>
<td>3.11</td>
</tr>
</tbody>
</table>

### V. INTER-LEVEL PROTOCOL VERIFICATION

To verify the hierarchical protocol, UC and SC simulators are developed. The framework includes one UC and five SCs. Six computers with the same specifications as the previous one are used to run the simulators, which communicate with each other over the transmission control protocol (TCP) on a 10Mbps local network. The UC runs as a server, while each SC runs as a client. Once all SCs are connected, the UC sends signals to start the simulation in a distributed fashion. In reality, the control can be synchronized with the signal of the global positioning system (GPS).

The load demand and power limits are shown in Fig. 14 and Table VIII, where load ratio is the ratio of power load to capacity threshold. The capacity threshold of the UC is set to 2500kVA, which is 80% of the sum of the local capacity limits (i.e., the coincidence factor is set to 0.8). The settings of EV mobility behaviors remain the same in the previous section. In the following, three scenarios are set for multi-angle verification, and the simulations of 10 days are made for each one.

<table>
<thead>
<tr>
<th>SC</th>
<th>( M )</th>
<th>Cap. limit [kVA]</th>
<th>Base load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>400</td>
<td>(a)</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>500</td>
<td>(a)</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>630</td>
<td>(a)</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>800</td>
<td>(b)</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>800</td>
<td>(b)</td>
</tr>
</tbody>
</table>

Scenario 1: Fig. 15 shows charging load profiles of SC 1, 2, and 3. As can be seen, the charging of each SC is constrained by the upper limits. For the five SCs, the averages of final SOC are 0.919, 0.921, 0.920, 0.922, and 0.920, respectively. All load profiles are almost coincident. This means that the upper-level capacity can be reasonably dispatched to all SCs. At each time step, the average overhead of UC is 139 bytes (TCP headers included), and the average time cost of hierarchical coordination is only 0.6 s.
Scenario 2: The real-time instruction of demand response from the UC is considered. The charging load profiles are shown in Fig. 16. The averages of final SOC are 0.843, 0.844, 0.844, 0.845, and 0.843, respectively. It can be seen that these five SCs can proportionally share the responsibility of load curtailing.

FIGURE 16. The samples of charging load profiles in Scenario 2: (a) upper and (b) station levels

Scenario 3: The EV numbers of SC 1 and 2 are raised to 200 and 250, respectively, and the other settings are the same as Scenario 1. Since the schemes of SC 1 and SC 2 are limited by local and upper limits, the average final SOCs should be lower than the other SCs. The averages of five SCs are 0.747, 0.748, 0.871, 0.872, and 0.872, respectively. As shown in Fig. 17, the load profiles of SC 1 and 2 are almost coincident, significantly different from the load profile of SC 3.

FIGURE 17. The samples of charging load profiles in Scenario 3: station levels

V. CONCLUSION
In charging coordination, relaxation and approximation ways are widely used for improving computation efficiency to meet the real-time coordination need. However, when dealing with heavy charging demand, they have some disadvantages that cannot be ignored, such as large differences in final SOC and frequent charging interruption.

Based on the recursive method, this paper proposes a novel search-swapping algorithm. In essence, this approach utilizes the non-aftereffect property of EV charging coordination to approximate the optimal solution. This property comes from the wide diversity of charging requirements. Since SSA gives a near-optimal solution, it still has a small optimality gap to the delicate algorithms employed by commercial solvers such as CPLEX. Numerous simulations show that the gap is less than 2%, which is acceptable in real applications. As a return, the SSA can provide much higher efficiency, less charging interruption, and smaller deviation in final SOC.

With the excellent efficiency, even the microcontrollers can run the SSA to cope with large-scale coordination, reducing overall costs. This will encourage more charging operators to participate. Furthermore, the proposed work can be extended to address other similar ILP problems as long as a cost factor is built to play the role of electricity price.

APPENDIX
The assumptions are as follows:
• Define $S_0$ as the global optimal scheme.
• Define $S$ as the near-optimal scheme given by the SSA.
• All EVs will leave at the time step $k^{dep}$.
• All EVs are charged at the charging rate $p$.

Take the cost-saving optimization as an example. At the lowest-price time step $j$, it is assumed that the charging load of $S_0$ is denoted $P_0$ while the charging load of $S$ is $(P_0 - mP)$ where $m$ is an integer larger than zero. At least one EV $n$ is not charged at this time step, expressed as

$$\begin{cases} (s_n^j = 1, s_n^j = 0) |_{S_0} \\
(s_n^j = 0, s_n^j = 1) |_{S}, j \in [k, k_2^j], y \in [k, k_2^j] \\
(\theta^j < \theta^y) |_{(S_0, S)}
\end{cases}$$

According to (5), the HS can swap $s_n^j$ and $s_n^j$ of $S$ to reduce the cost. As a result, the schemes of $S_0$ and $S$ should be the same at the time step $j$.

The rest time steps, such as the second and the third lowest-price ones, can be analyzed in a similar fashion. At last, it can be deduced that the schemes of $S_0$ and $S$ are equal in the charging cost.

REFERENCES
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2929626, IEEE Access