

A Two-stage Autonomous EV Charging Coordination Method Enabled by Blockchain

Jian Ping, Zheng Yan, and Sijie Chen

Abstract—Increasing electric vehicle (EV) penetration in distribution networks necessitate EV charging coordination. This paper proposes a two-stage EV charging coordination mechanism that frees the distribution system operator (DSO) from extra burdens of EV charging coordination. The first stage ensures that the total charging demand meets facility constraints, and the second stage ensures fair charging welfare allocation while maximizing the total charging welfare via Nash-bargaining trading. A decentralized algorithm based on the alternating direction method of multipliers (ADMM) is proposed to protect individual privacy. The proposed mechanism is implemented on the blockchain to enable trustworthy EV charging coordination in case a third-party coordinator is absent. Simulation results demonstrate the effectiveness and efficiency of the proposed approach.

Index Terms—Electric vehicle (EV) charging coordination, Nash bargaining, alternating direction method of multipliers (ADMM), blockchain.

I. INTRODUCTION

INCREASING electric vehicle (EV) penetration in distribution networks triggers overload risks to distribution network facilities and the need for EV charging coordination. Traditionally, EV charging is coordinated by a distribution system operator (DSO). The optimal charging schedule is derived by centralized optimization or decentralized iterations between a DSO and local controllers, e.g., charging stations, EVs, or smart buildings. However, two challenges arise from such coordination approaches:

1) It brings an extra computation burden to the DSO. The DSO needs to either solve an optimization problem including the details of all equipment or communicate with local controllers frequently.

2) It is vulnerable to single-point failures. A cyber attack to the DSO can fail the coordination.

This paper aims to address the following issues:

1) How to design an EV charging coordination mechanism that neither brings extra computational burdens to a

Manuscript received: November 5, 2019; accepted: July 23, 2020. Date of CrossCheck: July 23, 2020. Date of online publication: October 22, 2020.

This work was supported by National Natural Science Foundation of China (No. U1866206).

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

J. Ping, Z. Yan, and S. Chen (corresponding author) are with the Key Laboratory of Control of Power Transmission and Conversion, Ministry of Education (Shanghai Jiao Tong University), Shanghai, China (e-mail: ppjj1994@sjtu.edu.cn; yanz@sjtu.edu.cn; Sijie.chen@sjtu.edu.cn).

DOI: 10.35833/MPCE.2019.000139

DSO nor overloads facilities?

2) Given that local controllers are independent and profit-seeking decision-makers, how to derive a global optimal charging schedule while ensuring the fairness of welfare allocation?

3) How can local controllers coordinate charging schedules autonomously without a DSO?

To ensure that the charging demand meets facility constraints, remarkable studies have been done on EV charging coordination. In [1]-[5], a central operator collects the charging power demands of all EVs and derives the optimal charging schedule while considering system constraints. In [6], [7], the optimal charging schedule is derived by iterations between a DSO and individual EVs, which protects the privacy of EVs. In [8]-[10], EVs in a local area are controlled by an EV aggregator, who plays as a local controller. As an intermediary between a DSO and its on-site EVs, an EV aggregator always optimizes the local charging schedules and communicates with the DSO. Such a hierarchical structure eases the computational burden of a DSO. However, a DSO still needs to communicate with aggregators frequently for the EV charging power allocation, which may be beyond the duty of the DSO. In the above studies, a DSO faces extra computational burdens for EV charging coordination. It is also vulnerable to single-point failures.

Our first contribution is the proposition of a two-stage EV charging coordination mechanism. Charging stations (CSs) are set as local controllers of their on-site EVs. To mitigate facility overload in distribution networks, the charging power of a CS can not exceed an allowable limit which is defined as its charging power quota (CPQ). At the CPQ pre-allocation stage, CPQs are fairly allocated to CSs while meeting facility constraints. At the CPQ trading stage, CSs are allowed to trade CPQs each other. CSs with inelastic demands have stronger motivation for charging, who can buy CPQs from CSs with elastic charging demands. Given that CSs are independent and profit-seeking decision-makers in CPQ trading, the trading among CSs is realized by using Nash bargaining theory [11].

In the proposed mechanism, a DSO only needs to provide the total permissible charging load, which frees the DSO from managing charging coordination. CPQ allocations among CSs are optimized via CPQ trading. The Nash bargaining-based CPQ trading maximizes the total welfare while the benefits of trading are fairly allocated to CSs [12].

Moreover, for computation and privacy concerns, it is



more practical to solve the Nash bargaining problem in a decentralized manner. The alternating direction method of multipliers (ADMM) is introduced in [13]. By using ADMM, a Nash bargaining problem is decomposed into an upper problem and some sub-problems, and can be solved through iterations. However, it still requires a third-party coordinator to solve the upper problem and communicate with sub-problem solvers. Letting a DSO be this third-party coordinator still brings extra burdens to the DSO. If not letting the DSO be the coordinator, there are few other third-party candidates suitable for the role.

Blockchain, as an emerging technology, paves the way for autonomous operation for independent decision-makers without a trusted third party. With an increasing number of independent participants in power systems, especially in distribution networks, blockchain applications in distribution systems have been explored in these years. In [14], blockchain enables trusted and secure settlement of electricity trading in a distribution network. In [15], [16], blockchain is utilized for trusted peer-to-peer (P2P) trading among prosumers. In [17], EVs in a CS can trade electricity with each other on a blockchain-based platform. In [18], EVs can audit and share the optimal trading records between EVs and a CS on a blockchain. In [19], blockchain is utilized to ensure the security of charging process management. In [20], blockchain is used to ensure the security and privacy between EVs and CSs. These studies show the potential of blockchain for enabling trusted operation in energy systems. However, existing studies on blockchain-based EV charging mainly focus on the trading among individual EVs or between EVs and CSs, whereas the coordination of CSs is rarely concerned.

Our second contribution is to enable fully autonomous EV charging coordination based on ADMM and blockchain. Firstly, the CPQ trading problem is solved in a decentralized way by using ADMM, which significantly protects individual privacy. Then, the coordination mechanism is implemented via blockchain. The blockchain-based implementation can coordinate EV charging while removing the need for a third-party coordinator, enabling fully autonomous EV charging coordination.

The rest of this paper is organized as follows. Section II presents a two-stage EV charging coordination mechanism. Section III proposes a decentralized EV charging coordination method. Section IV gives simulation results. Section V concludes the paper.

II. TWO-STAGE EV CHARGING COORDINATION

A. Proposed Timeline

In the proposed EV charging coordination mechanism, a day is split into a number of equal time intervals. In each time interval, the EV charging schedule is coordinated through two sequential stages: the CPQ pre-allocation stage and the CPQ trading stage.

In the CPQ pre-allocation stage, the DSO calculates the total permissible charging load in this time interval based on the forecast of conventional loads, while each CS submits its charging power demand according to its on-site EV informa-

tion. If the total charging power demand does not exceed the total permissible charging load, the charging power demands of CSs are met and the coordination is finished. Otherwise, CPQs would be allocated to CSs fairly, which then triggers the CPQ trading stage.

In the CPQ trading stage, CSs can trade their CPQs with each other. CSs with inelastic demands have stronger motivation for charging, who can buy CPQs from CSs with elastic demands. Through this stage, CSs can either meet their inelastic charging power demands or receive payments by selling CPQs. Hence this stage yields Pareto improvement.

In general, the CPQ pre-allocation stage provides an initial CPQ allocation scheme before the CPQ trading. Based on the CPQ pre-allocation stage, the CPQ trading stage improves the charging welfare of CSs through the CPQ trading. The timeline of EV charging coordination in a time interval is illustrated in Fig. 1, where P_t^{TL} denotes the total permissible charging load in time interval t , $Q_{i,t}^A$ denotes the charging power demand of CS i in time interval t , and \mathcal{M}^{CS} is the set of CSs in a distribution network.

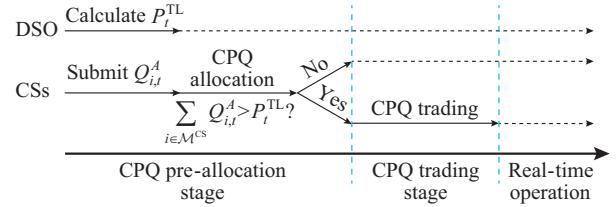


Fig. 1. Timeline of two-stage EV charging coordination in a time interval.

B. CPQ Pre-allocation Stage

At the beginning of the CPQ pre-allocation stage, each CS submits its charging power demand by solving a convex optimization problem P_0 , as shown in (1)-(3), according to the information collected from its on-site EVs.

$$\max W_{i,t}(Q_{i,t}^A, \mathbf{P}_{i,t}) \quad (1)$$

s.t.

$$\mathbf{F}_i(Q_{i,t}^A, \mathbf{P}_{i,t}) \leq \mathbf{0} \quad (2)$$

$$\mathbf{G}_i(Q_{i,t}^A, \mathbf{P}_{i,t}) = \mathbf{0} \quad (3)$$

where $\mathbf{P}_{i,t}$ is the charging power vector of EVs parked at CS i in time interval t ; and function $W_{i,t}(\cdot)$ represents the charging welfare of CS i in time interval t . Constraints (2) and (3) represent the inequality constraints and equality constraints in CS i , respectively.

Then, CPQs are allocated by (4).

$$\mathbf{Q}_t^A = \begin{cases} \mathbf{Q}_t^A & \sum_{i \in \mathcal{M}^{\text{CS}}} Q_{i,t}^A \leq P_t^{\text{TL}} \\ \frac{P_t^{\text{TL}} \mathbf{P}^{\text{CS}, \text{max}}}{\sum_{i \in \mathcal{M}^{\text{CS}}} P_i^{\text{CS}, \text{max}}} & \sum_{i \in \mathcal{M}^{\text{CS}}} Q_{i,t}^A > P_t^{\text{TL}} \end{cases} \quad (4)$$

where $\mathbf{Q}_t^A = (Q_{i,t}^A)_{i \in \mathcal{M}^{\text{CS}}}$, and $Q_{i,t}^A$ is the allocated CPQs of CS i ; $\mathbf{P}^{\text{CS}, \text{max}} = (P_i^{\text{CS}, \text{max}})_{i \in \mathcal{M}^{\text{CS}}}$, and $P_i^{\text{CS}, \text{max}}$ is the rated capacity of CS i ; and $\mathbf{Q}_t^A = (Q_{i,t}^A)_{i \in \mathcal{M}^{\text{CS}}}$.

According to (4), if the total charging power demand does not exceed the total permissible charging load, the charging

power demands of CSs are not curtailed and the coordination process is finished. Otherwise, the CPQs are allocated to CSs according to the proportion of their rated capacities in the sum of the capacities. This is because: ① a CS with a larger capacity usually needs to pay a higher capacity fee in reality; ② all CSs have the same contributions to the load of the transformer they are affiliated with. Such an allocation scheme ensures that the total charging load does not exceed the permissible level. Moreover, this scheme can prevent CSs from submitting an exaggerated charging power demand. Assuming that CS i submits an exaggerated demand $Q_{i,t}^{A'}$, if the total demand does not exceed the limit, CS i would be punished in the settlement stage because its actual charging load deviates from $Q_{i,t}^{A'}$. If the total demand exceeds the limit, the allocated CPQs of CS i is independent of $Q_{i,t}^{A'}$. Hence, a rational CS would submit its actual charging power demand.

If the charging power demands of CSs are curtailed, each CS reschedules the charging power of its on-site EVs according to the reduced CPQs, i.e., each CS resolves problem P0, with an additional constraint (5). The solution $W_{i,t}^0$ denotes the maximum charging welfare of CS i before CPQ trading, which also named as the disagreement point in the bargaining literature [21].

$$Q_{i,t}^A = Q_{i,t}^{A,*} \quad (5)$$

C. CPQ Trading Stage

In the CPQ trading stage, CSs trade CPQs with other CSs to achieve higher charging welfare. Let $Q_{i,t}^B$ and $\pi_{i,t}^B$ denote the amount and the unit price of CPQ bought by CS i , respectively, and the increase of the charging welfare of CS i through CPQ trading is a function of $Q_{i,t}^B$, $\pi_{i,t}^B$, and $\mathbf{P}_{i,t}$, as shown in (6).

$$B_{i,t}(Q_{i,t}^B, \pi_{i,t}^B, \mathbf{P}_{i,t}) = W_{i,t}(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) - \pi_{i,t}^B Q_{i,t}^B \Delta t - W_{i,t}^0 \quad (6)$$

where $B_{i,t}(Q_{i,t}^B, \pi_{i,t}^B, \mathbf{P}_{i,t})$ is the increase of the charging welfare of CS i through CPQ trading; Δt is the duration of a time interval; and a negative $Q_{i,t}^B$ denotes the amount of sold CPQs.

In (6), the first term represents the charging welfare considering the bought/sold CPQs; the second term denotes the cost/revenue of buying/selling CPQs; and the third term is the charging welfare before the CPQ trading.

The CPQ trading process is realized by using Nash bargaining to achieve a fair allocation of trading benefits. The Nash bargaining solution for CPQ trading can be derived by solving the optimization problem defined by (7)-(12).

$$\max \prod_{i \in \mathcal{M}^{\text{CS}}} B_{i,t}(Q_{i,t}^B, \pi_{i,t}^B, \mathbf{P}_{i,t}) \quad (7)$$

s.t.

$$\mathbf{F}_i(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) \leq \mathbf{0} \quad i \in \mathcal{M}^{\text{CS}} \quad (8)$$

$$\mathbf{G}_i(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) = \mathbf{0} \quad i \in \mathcal{M}^{\text{CS}} \quad (9)$$

$$\sum_{i \in \mathcal{M}^{\text{CS}}} Q_{i,t}^B = 0 \quad (10)$$

$$\sum_{i \in \mathcal{M}^{\text{CS}}} \pi_{i,t}^B Q_{i,t}^B = 0 \quad (11)$$

$$B_{i,t}(Q_{i,t}^B, \pi_{i,t}^B, \mathbf{P}_{i,t}) \geq 0 \quad i \in \mathcal{M}^{\text{CS}} \quad (12)$$

Constraints (8) and (9) represent the inequality constraints and equality constraints in CS i , respectively; constraints (10) and (11) represent the CPQ balance constraint and payment balance constraint, respectively; and constraint (12) ensures that CSs are better off in CPQ trading.

The Nash bargaining solution $\{Q_{i,t}^{B,*}, \pi_{i,t}^{B,*}, \mathbf{P}_{i,t}^*\}$, $i \in \mathcal{M}^{\text{CS}}$, derived by solving (7)-(12) has the following advantages:

1) Individual rationality. The Nash bargaining solution ensures that all CSs are better off. Hence, CSs have incentives to participate in CPQ trading.

2) Pareto optimality. There is no alternative solution where a CS is better off while the interests of other CSs are not compromised.

3) Fair welfare allocation. The increase of charging welfare through CPQ trading is equally allocated to all CSs, i.e., $B_{i,t}(Q_{i,t}^{B,*}, \pi_{i,t}^{B,*}, \mathbf{P}_{i,t}^*) = B_{j,t}(Q_{j,t}^{B,*}, \pi_{j,t}^{B,*}, \mathbf{P}_{j,t}^*)$, $\forall i, j \in \mathcal{M}^{\text{CS}}$.

The Nash bargaining problem is non-convex because of the objective function (7) and constraint (11), and it is hard to solve. Inspired by [13], this paper decomposes the Nash bargaining problem in the CPQ trading stage into two sequential problems, P1 and P2, where problem P1 figures out the optimal transferred CPQs and problem P2 determines the optimal CPQ prices.

Problem P1 denotes a total charging welfare maximization problem, as shown in (13).

$$\begin{cases} \max \sum_{i \in \mathcal{M}^{\text{CS}}} W_{i,t}(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) \\ \text{s.t. (8)-(10)} \end{cases} \quad (13)$$

Solving problem P1 indicated by (13) can yield the optimal transferred CPQs of CSs $Q_{i,t}^{B,*}$. Yet, the information about $\pi_{i,t}^B$ is missing because the payments of CSs cancel out each other. $\pi_{i,t}^B$ is derived via problem P2.

Problem P2 formulates a CPQ price bargaining problem, as shown in (14)-(16).

$$\max \prod_{i \in \mathcal{M}^{\text{CS}}} B_{i,t}(Q_{i,t}^{B,*}, \pi_{i,t}^B, \mathbf{P}_{i,t}^*) \quad (14)$$

s.t.

$$\sum_{i \in \mathcal{M}^{\text{CS}}} \pi_{i,t}^B Q_{i,t}^{B,*} = 0 \quad (15)$$

$$B_{i,t}(Q_{i,t}^{B,*}, \pi_{i,t}^B, \mathbf{P}_{i,t}^*) \geq 0 \quad i \in \mathcal{M}^{\text{CS}} \quad (16)$$

Equation (14) can be rewritten in a logarithmic function form, as shown in (17).

$$\max \sum_{i \in \mathcal{M}^{\text{CS}}} \ln B_{i,t}(Q_{i,t}^{B,*}, \pi_{i,t}^B, \mathbf{P}_{i,t}^*) \quad (17)$$

By solving problem P2 indicated by (14)-(16), the Nash bargaining solution of CPQ prices can be derived.

The CPQ trading based on Nash bargaining is shown in Fig. 2. For simplicity, only two charging stations are considered. In Fig. 2, the blue dashed line and the red dashed line represent the disagreement points of CS 1 and CS 2, i.e., $W_{1,t}^0$ and $W_{2,t}^0$, respectively. The orange area includes all feasible solutions of the CPQ trading, i.e., CPQ trading results satisfying constraints (8)-(12). The green line represents the Pareto frontier of the CPQ trading, i.e., the trading results derived by solving problem P1. The Nash bargaining solution is the tangency point of the objective function of problem

P2, i.e., $f(B_{1,t}, B_{2,t})=B_{1,t}B_{2,t}$, and the Pareto frontier, i.e., the trading results derived by solving problem P2. It can be seen that the Nash bargaining solution not only yields Pareto efficiency but also allocates the trading welfare equally.

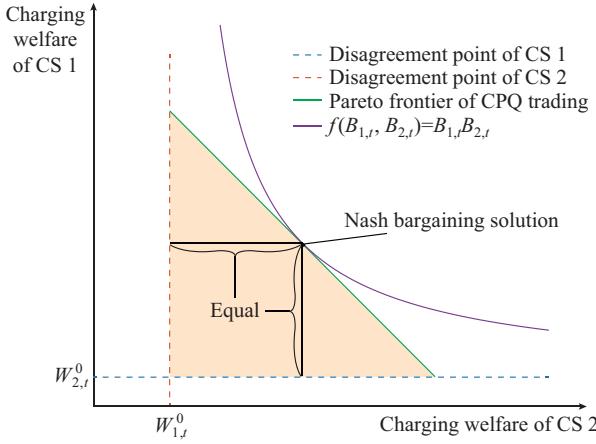


Fig. 2. Schematic diagram of CPQ trading based on Nash bargaining.

Problems P1 and P2 are both convex and can be easily solved centrally. However, the centralized solution of problems P1 and P2 may be impractical in reality because of the following concerns:

- 1) A central coordinator needs to have all the information about CSs and their on-site EVs, which may violate individual privacy.
- 2) There may not exist a central coordinator who is willing to organize the CPQ trading in many local markets.

III. DECENTRALIZED EV CHARGING COORDINATION

In this section, a decentralized implementation of the two-stage coordination mechanism is proposed. First, ADMM-based decentralized algorithms are proposed to solve optimization problems P1 and P2 in the CPQ trading stage. In this way, a CS only needs to disclose limited information, i.e., $Q_{i,t}^B$ and $\pi_{i,t}^B$. Then, the proposed two-stage coordination mechanism is implemented on a blockchain. A set of CSs plays the role of a central coordinator, enabling autonomous CPQ trading without the need of a central coordinator.

A. ADMM-based Decentralized Algorithms

1) ADMM-based Algorithm for Problem P1

Problem P1 can be decomposed into several sub-problems and an upper problem, and be solved through iterations. In problem P1, the objective function in (13) and constraints (8) and (9) are separated by each CS, whereas constraint (10) couples CSs. In order to decompose problem P1, auxiliary variable $\hat{Q}_{i,t}^B$ ($i \in \mathcal{M}^{\text{CS}}$) is introduced, and constraint (10) is replaced as follows:

$$\hat{Q}_{i,t}^B = Q_{i,t}^B \quad i \in \mathcal{M}^{\text{CS}} \quad (18)$$

$$\sum_{i \in \mathcal{M}^{\text{CS}}} \hat{Q}_{i,t}^B = 0 \quad (19)$$

where $\hat{Q}_{i,t}^B$ ($i \in \mathcal{M}^{\text{CS}}$) can be interpreted as the optimal CPQs of CSs calculated by the upper problem.

By using ADMM [22], $Q_{i,t}^B$ and $\hat{Q}_{i,t}^B$ reach an agreement through iterations so that problem P1 is solved.

Problem sub-P1_{*i*} is formulated as:

$$\max \left[W_{i,t}(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) - \frac{\rho_1(k)}{2} (\hat{Q}_{i,t}^B(k) - Q_{i,t}^B)^2 + \lambda_{i,t}(k) Q_{i,t}^B \right] \quad (20)$$

s.t.

$$\mathbf{F}_i(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) \leq \mathbf{0} \quad (21)$$

$$\mathbf{G}_i(Q_{i,t}^{A,*} + Q_{i,t}^B, \mathbf{P}_{i,t}) = \mathbf{0} \quad (22)$$

where $\rho_1(k) > 0$ is a penalty parameter in iteration k ; and $\lambda_{i,t}(k)$ is the Lagrangian multiplier of constraint (18) in iteration k .

Problem upper-P1 is formulated as:

$$\begin{cases} \max \sum_{i \in \mathcal{M}^{\text{CS}}} \left[-\frac{\rho_1(k)}{2} (\hat{Q}_{i,t}^B(k+1) - Q_{i,t}^B(k+1))^2 - \lambda_{i,t}(k) \hat{Q}_{i,t}^B \right] \\ \text{s.t. (19)} \end{cases} \quad (23)$$

The Lagrangian multiplier $\lambda_{i,t}$ is updated according to (24).

$$\lambda_{i,t}(k+1) = \lambda_{i,t}(k) + \rho_1(k) (\hat{Q}_{i,t}^B(k+1) - Q_{i,t}^B(k+1)) \quad (24)$$

In iteration k , each CS calculates $Q_{i,t}^B(k+1)$ and $\mathbf{P}_{i,t}(k+1)$ by solving sub-P1_{*i*} with $\lambda_{i,t}(k)$ and $\hat{Q}_{i,t}^B(k)$ derived by the upper problem upper-P1 in iteration $k-1$. Then, the upper problem upper-P1 calculates $\lambda_{i,t}(k+1)$ and $\hat{Q}_{i,t}^B(k+1)$ according to $Q_{i,t}^B(k+1)$ ($i \in \mathcal{M}^{\text{CS}}$) submitted by CSs. The iteration stops when the following terminal condition is satisfied.

$$r_1(k) = \sum_{i \in \mathcal{M}^{\text{CS}}} |\hat{Q}_{i,t}^B(k+1) - Q_{i,t}^B(k+1)| \leq \epsilon_1 \quad (25)$$

$$s_1(k) = \rho_1(k) \sum_{i \in \mathcal{M}^{\text{CS}}} |Q_{i,t}^B(k+1) - Q_{i,t}^B(k)| \leq \epsilon_2 \quad (26)$$

where $r_1(k)$ and $s_1(k)$ are the primal residual and dual residual of problem P1 in iteration k , respectively; and ϵ_1 and ϵ_2 are the small positive numbers.

The self-adaptive rule of the penalty parameter proposed in [23] is introduced to ensure efficient convergence, as shown in (27).

$$\rho_1(k+1) = \begin{cases} \frac{\rho_1(k)}{1 + \zeta(k)} & r_1(k) < h s_1(k) \\ \rho_1(k)(1 + \zeta(k)) & h r_1(k) \geq s_1(k) \\ \rho_1(k) & \text{otherwise} \end{cases} \quad (27)$$

where $h \in (0, 1)$; and the non-negative sequence $\{\zeta(k)\}$ satisfies $\sum_{k=1}^{+\infty} \zeta(k) < \infty$.

2) ADMM-based Algorithm for Problem P2

Similar to problem P1, problem P2 can be decomposed into several sub-problems and an upper problem, and be solved through iterations. In problem P2, objective function (14) and constraint (16) are separated by CSs, whereas constraint (15) couples CSs. In order to decompose problem P2, auxiliary variables $\hat{\pi}_{i,t}^B = (\hat{\pi}_{i,i,t}^B)_{i \in \mathcal{M}^{\text{CS}}}$ are introduced, and constraint (15) is replaced as follows:

$$\hat{\pi}_{i,t}^B = \pi_{i,t}^B \quad i \in \mathcal{M}^{\text{CS}} \quad (28)$$

$$\sum_{i \in \mathcal{M}^{\text{CS}}} \hat{\pi}_{i,t}^B Q_{i,t}^{B,*} = 0 \quad (29)$$

Problem sub-P2_i is defined as:

$$\begin{cases} \max \left[\ln B_{i,t}(Q_{i,t}^{B,*}, \pi_{i,t}^B, P_{i,t}^*) - \frac{\rho_2(k)}{2} (\hat{\pi}_{i,t}^B(k) - \pi_{i,t}^B)^2 + \gamma_{i,t}(k) \pi_{i,t}^B \right] \\ \text{s.t. (16)} \end{cases} \quad (30)$$

where $\rho_2(k) > 0$ is a penalty parameter in iteration k ; and $\gamma_{i,t}(k)$ is the Lagrangian multiplier of constraint (28) in iteration k .

Problem upper-P2 is defined as:

$$\begin{cases} \max \sum_{i \in \mathcal{M}^{\text{CS}}} \left[-\frac{\rho_2(k)}{2} (\hat{\pi}_{i,t}^B - \pi_{i,t}^B(k+1))^2 - \gamma_{i,t}(k) \hat{\pi}_{i,t}^B \right] \\ \text{s.t. (29)} \end{cases} \quad (31)$$

The Lagrangian multiplier $\gamma_{i,t}$ is updated according to (32).

$$\gamma_{i,t}(k+1) = \gamma_{i,t}(k) + \rho_2(k) (\hat{\pi}_{i,t}^B(k+1) - \pi_{i,t}^B(k+1)) \quad (32)$$

In iteration k , each CS calculates $\pi_{i,t}^B$ by solving problem sub-P2_i with $\gamma_{i,t}(k)$ and $\hat{\pi}_{i,t}^B(k)$ derived by the upper problem upper-P2. Then, the upper problem upper-P2 figures out $\gamma_{i,t}(k+1)$ and $\hat{\pi}_{i,t}^B(k+1)$ according to $\pi_{i,t}^B(k+1)$ ($i \in \mathcal{M}^{\text{CS}}$) submitted by CSs. The iteration stops when the following terminal condition is satisfied.

$$r_2(k) = \sum_{i \in \mathcal{M}^{\text{CS}}} |\hat{\pi}_{i,t}^B(k+1) - \pi_{i,t}^B(k+1)| \leq \epsilon_3 \quad (33)$$

$$s_2(k) = \rho_2(k) \sum_{i \in \mathcal{M}^{\text{CS}}} |\pi_{i,t}^B(k+1) - \pi_{i,t}^B(k)| \leq \epsilon_4 \quad (34)$$

where $r_2(k)$ and $s_2(k)$ are the primal residual and dual residual of problem P2 in iteration k , respectively; ϵ_3 and ϵ_4 are the small positive numbers; and the selection method of penalty parameter $\rho_2(k)$ is similar to $\rho_1(k)$ and is omitted here.

Unlike the centralized coordination, a CS in the proposed ADMM-based algorithms does not need to expose those sensitive information about itself and on-site EVs to a central coordinator, but only limited information. This protects individual privacy during the CPQ trading.

B. Blockchain-based Implementation

The above decentralized coordination approach still relies on a center for coordinating CSs, i.e., allocating CPQs in the CPQ pre-allocation stage and solving problems upper-P1 and upper-P2 in the CPQ trading stage. Given that there may not exist such a center, a blockchain-based implementation of the EV charging coordination is proposed, which enables fully autonomous EV charging coordination. The two-level structure of the blockchain-based system is shown in Fig. 3, where a set of CSs (named as delegates) plays the role of a central coordinator. As demonstrated in Fig. 3, on the upper level, delegates solve upper problems and communicate with CSs and a DSO; on the lower level, CSs communicate with delegates and optimize the local charging schedules, and a DSO provides the total permissible charging load. A simplified delegated Byzantine fault tolerance (SDBFT) consensus algorithm is proposed to ensure that CSs can trust the coordination results proposed by delegates.

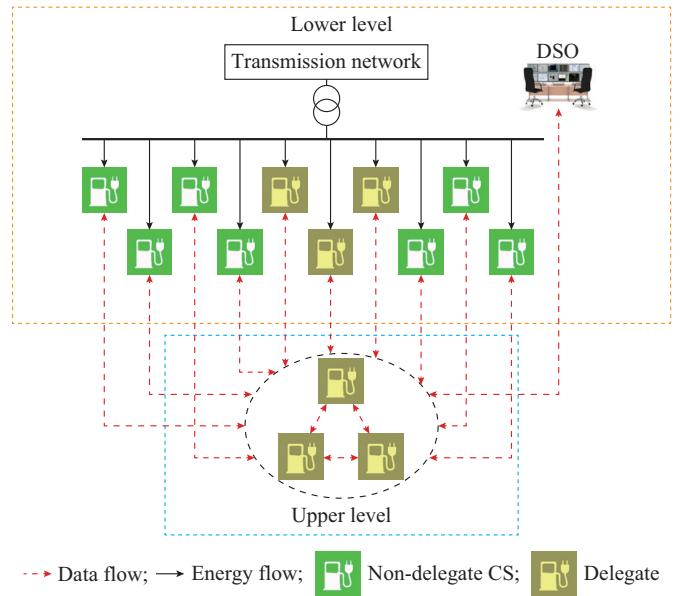


Fig. 3. Structure of blockchain-based system.

In decentralized EV charging coordination approaches, it is usually assumed that the communication network is synchronous, i.e., a message sent by a node can be received by all nodes within a fixed and known time. This is because: ① it may not converge in time if the network is not synchronous; ② it is convenient to establish a synchronous communication network among charging stations in a distribution network in practice [24]. The blockchain-based implementation in this paper is also under this assumption.

1) Definitions of Key Concepts

Definition 1: round. Rounds can be deemed as the basic units of consensus processes and are numbered consecutively. In a round, all delegates select a leader, who leads the rest of delegates to provide the coordinator functions. In round V , delegate l is selected as the leader of this round, where $l = V \bmod |\mathcal{M}^d|$, and $|\mathcal{M}^d|$ is the number of delegates. In the following text, “round” is represented by “view”, which is commonly used in the blockchain literature.

Definition 2: message. Messages transferred among CSs have a standard format. A message m has a fixed set of fields, defined as $\langle m^D, m^V, m^P, m^S, m^C \rangle$, where m^D is the digest of a message, containing the hash of its parent block and the hash of $\langle m^V, m^P, m^S, m^C \rangle$. m^V , m^P and m^S record the view, the phase and the stage in which the message is being sent, respectively. $m^S \in \{\text{requireCPQ}, \text{solveP1}, \text{solveP2}\}$, in which requireCPQ, solveP1, solveP2 represent the CPQ pre-allocation stage, solving problem P1 in the CPQ trading stage and solving problem P2 in the CPQ stage, respectively. m^C records the detailed message.

Definition 3: digital signature. When sending a message, one should sign the message with its private key. A message $\langle m \rangle$ signed by node i is represented by $\langle m \rangle^{\sigma}$. Anyone can verify a signed message with the sender’s public key, ensuring that the message is not tampered by others [25].

Definition 4: smart contract. A smart contract is a set of software codes which set the rule for delegates about how to

derive the coordination result according to the messages sent by CSs. Smart contracts are approved by all CSs before the coordination. In the CPQ pre-allocation stage, i.e., $m^s \leftarrow \text{requireCPQ}$, the contract refers to solving (4). In solving problem P1 in the CPQ trading stage, i.e., $m^s \leftarrow \text{solveP1}$, the contract refers to solving (23) and (24). In solving problem P2 in the CPQ trading stage, i.e., $m^s \leftarrow \text{solveP2}$, the contract refers to solving (31) and (32).

2) Structure

The structure of the blockchain-based system can be divided into two layers, as illustrated in Fig. 3. In the upper layer, the delegates play the role of the coordinator besides their own roles as CSs, i.e., solving the upper problems and communicating with CSs and a DSO. In the lower layer, CSs communicate with the delegates and optimize the charging schedules of on-site EVs. As a special participant, a DSO is also in the lower layer, which provides P_t^{TL} to the delegates. Instead of communicating with a central coordinator, the CSs and DSO communicate with the delegates in the SDBFT consensus algorithm. In the CPQ pre-allocation stage, each CS or the DSO sends a request message (containing $Q_{i,t}^A$ or P_t^{TL}) to the delegates. Then, the delegates execute the smart contract in the CPQ pre-allocation stage and send a block (containing $Q_t^{A,*}$) to the CSs and DSO. In the CPQ trading stage, each CS sends request messages (containing the updated $Q_{i,t}^B$ or $\pi_{i,t}^B$) to the delegates in each iteration of the ADMM algorithms. Then, the delegates execute the smart contracts of solving problem P1 or P2 in the CPQ trading stage and send blocks (containing $\lambda_{i,t}(k)$ and $\hat{Q}_{i,t}^B(k)$ or $\gamma_{i,t}(k)$ and $\hat{\pi}_{i,t}^B(k)$) to the CSs and DSO. The iteration between the CSs and delegates stops when the terminal condition of problem P1 or problem P2 is satisfied.

3) Blockchain-based EV Charging Coordination

In each view, the SDBFT consensus algorithm needs to go through 4 phases of Request, Pre-prepare, Prepare, and Reply in normal-case operation and an additional phase of View-change in faulty-leader operation, i.e., $m^P \in \{\text{Request, Pre-prepare, Prepare, Reply, View-change}\}$. Algorithm 1 demonstrates the SDBFT consensus algorithm in normal-case operation.

In the Request phase, CS i sends a signed request message $m_i = \langle m_i^D, m_i^V, m_i^P, m_i^S, m_i^C \rangle^{\sigma_i}$ to a leader, where m_i^V , m_i^P and m_i^S are the current view, phase and stage recorded by CS i itself, respectively. If $m^s \leftarrow \text{requireCPQ}$, m_i^C refers to $Q_{i,t}^A$; if $m^s \leftarrow \text{solveP1}$, m_i^C refers to $Q_{i,t}^B$; and if $m^s \leftarrow \text{solveP2}$, m_i^C refers to $\pi_{i,t}^B$. A leader verifies m_i by using the $\text{verifyM}(m_i)$ function. Function $\text{verifyM}(m_i)$ returns true only if: ① the hash values in m_i^d are correct; ② m_i^V , m_i^P and m_i^S are the current view, phase view and stage view recorded by the leader, respectively; ③ σ_i is signed by the private key of CS i .

In the Pre-prepare phase, a leader l , after collecting the request messages from all CSs, broadcasts a pre-prepare message $m_l = \langle m_l^D, m_l^V, m_l^P, m_l^S, \langle m_l^R, m_l^A \rangle \rangle^{\sigma_l}$ to all delegates, where m_l^R consists of all request messages, $m_l^A = C_l^S(m_l^R)$ is the leader's proposal of the coordination result by executing the smart contract C_l^S in the current stage. A delegate verifies a pre-prepare message m_l by calling the $\text{verifyM}(m_l)$ function.

Algorithm 1 SDBFT consensus algorithm

```

1: //Request
2: for  $i \in \mathcal{M}^{\text{CS}}$  do
3:   CS  $i$  sends  $m_i$  to leader  $l = m_i^V \bmod |\mathcal{M}^{\text{d}}|$ 
4:   if  $\text{verifyM}(m_i) = \text{true}$  then
5:     leader  $l$  adds request:  $m_l^R \leftarrow m_i$ 
6:   end if
7: end for
8: //Pre-prepare
9: Leader  $l$  executes  $C_l^S$  to obtain  $m_l^A; m_l^A \leftarrow C_l^S(m_l^R)$ 
10: Leader  $l$  broadcasts  $m_l$  to all delegates
11: for  $r \in \mathcal{M}^{\text{d}}$  do
12:   if  $\text{verifyM}(m_l) = \text{true}$  and  $\text{verifyM}(m_r) = \text{true}, i \in \mathcal{M}^{\text{CS}}$  then
13:     //Prepare
14:     if  $m_l^A = C_r^S(m_l^R)$  then
15:       Delegate  $r$  sends  $m_r$  to leader  $l$ 
16:     end if
17:   end if
18: end for
19: for  $r \in \mathcal{M}^{\text{d}}$  do
20:   if  $\text{verifyM}(m_r) = \text{true}$  then
21:     Leader  $l$  combines signature:  $\sigma \leftarrow \sigma_r$ 
22:   end if
23: //Reply
24: Leader  $l$  broadcasts  $B^V$  to all CSs
25: end for
26: for  $i \in \mathcal{M}^{\text{CS}}$  do
27:   if  $\text{verifyM}(B^V) = \text{true}$  then
28:     CS  $i$  accepts  $B^V$  and  $m_i^V \leftarrow m_i^V + 1$ 
29:   else
30:     //View-change
31:     Go to Step 1 in Algorithm 2
32:   end if
33: end for

```

In the Prepare phase, each delegate verifies m_l^A in the pre-prepare message by executing C_l^S . If it is correct, delegate r generates a prepare message $m_r = \langle m_l^D, m_l^V, m_l^P, m_l^S, \langle m_l^R, m_l^A \rangle \rangle^{\sigma_r}$ by signing the pre-prepare message and then sends it back to a leader. A leader verifies a prepare message m_r by calling the $\text{verifyM}(m_r)$ function.

In the Reply phase, only if a leader collects more than $|\mathcal{M}^{\text{d}}|/2$ delegates' prepare messages, it generates the final block in this view $B^V = \langle m_l^D, m_l^V, m_l^P, m_l^S, \langle m_l^R, m_l^A \rangle \rangle^{\sigma}$, where σ contains more than $|\mathcal{M}^{\text{d}}|/2$ signatures (one from the leader and others from non-leader delegates). Then, the leader broadcasts the final block to all CSs (including delegates). Once receiving a correct block, i.e., a final block with more than $|\mathcal{M}^{\text{d}}|/2$ signatures, a CS adds it to its local blockchain and considers m_l^A in the reply message as the coordination result in this view. Let $|\mathcal{M}^{\text{f}}|$ be the number of faulty delegates, the proposed algorithm can work as long as $|\mathcal{M}^{\text{d}}| \geq 2|\mathcal{M}^{\text{f}}| + 1$ [26].

If CS i does not receive a correct final block by a pre-set time since it sends a request message to a leader, it means that the leader may be faulty. In such a faulty-leader case, CS i would ask all CSs (including non-delegate CSs and delegate CSs) for changing the view so that the next delegate becomes the new leader, triggering a View-change phase. CS i sends a signed view-change message $m_i = \langle m_i^D, m_i^V, m_i^P, m_i^S, m_i^C \rangle^{s_i}$ to all CSs, where m_i^V , m_i^P and m_i^S are the current view, current phase (View-change) and current stage recorded by CS i , respectively, and m_i^C refers to a view-change request. Once a CS receives and verifies a view-change message, the CS abandons any message or block received in this view, changes the view number $V \leftarrow V + 1$, and then restarts Algorithm 1. Algorithm 2 illustrates the proposed view-change algorithm.

Algorithm 2 View-change in SDBFT consensus algorithm

```

1: if CS  $i$  does not receive correct  $B^V$  in a pre-set time then
2:   CS  $i$  sends a view-change message  $m_i$  to CS  $j \in \mathcal{M}^{\text{CS}}$ 
3:    $m_i^V \leftarrow m_i^V + 1$ 
4: end if
5: for  $j \in \mathcal{M}^{\text{CS}}$  do
6:   if  $\text{verifyM}(m_i) = \text{true}$  then
7:      $m_j^V \leftarrow m_j^V + 1$ 
8:   end if
9: end for
10: Back to Step 1 in Algorithm 1

```

In the SDBFT consensus algorithm, once a CS receives a correct final block and does not receive view-change messages from other CSs, it could confirm that all CSs receives the same coordination result, and the result is valid because it has been verified by the majority of delegates. Hence, based on the SDBFT consensus algorithm, the EV charging coordination can be implemented on a blockchain. The blockchain-based EV charging coordination algorithm is shown in Algorithm 3. Figure 4 illustrates the communication pattern on the blockchain-based implementation when solving problem P2 in the CPQ trading stage. In Fig. 4, C1 and C2 are non-delegate CSs, D0-D3 are delegates, and D0 is the leader in this view.

The blockchain-based implementation has the following advantages:

1) It enables fully autonomous EV charging coordination. Given that the proposed ADMM-based algorithms still rely on a coordinator, the blockchain-based implementation enables the CPQ trading in case a third-party coordinator is absent.

2) It is fair, trustworthy, and transparent. The smart contracts are designed based on Nash bargaining theory and are pre-approved by all CSs before coordination, ensuring fairness. The SDBFT consensus algorithm ensures that the coordination cannot be tampered with as long as faulty delegates are the minority, enabling trusted coordination. Once being verified and recorded on a blockchain, all CSs follow the same result and the result is tamper-resistant, guaranteeing transparency.

3) It has good scalability. The SDBFT consensus algorithm has linear communication complexity, i.e., the communication complexity is a linear function of the number of CSs. This ensures efficient EV charging coordination for a large number of CSs.

Algorithm 3 Blockchain-based EV charging coordination

```

1: for  $T \leftarrow 1, 2, 3, \dots$  do
2:   //CPQ pre-allocation stage
3:   DSO  $s$  generates  $m_s$ , where  $m_s^C \leftarrow P_t^{\text{TL}}$ 
4:   for  $i \in \mathcal{M}^{\text{CS}}$  do
5:     CS  $i$  solves local optimization problem  $P0_i$ 
6:     CS  $i$  generates  $m_i$ , where  $m_i^C \leftarrow Q_{i,t}^A$ 
7:   end for
8:   Running Step 1 to Step 32 in Algorithm 1
9: end for
10: CS  $i \in \mathcal{M}^{\text{CS}}$  gets  $Q_{i,t}^{A,*}$  according to  $B^V$ 
11: if  $Q_{i,t}^{A,*} = Q_t^A$  then
12:   CS  $i \in \mathcal{M}^{\text{CS}}$  waits for the next time interval
13: else
14:   //Solving problem P1 in CPQ trading stage
15:   Initialization:  $k = 1$ ,  $\hat{Q}_{i,t}^B(k) = \lambda_{i,t}(k) = 0$ ,  $i \in \mathcal{M}^{\text{CS}}$ 
16:   repeat
17:     for  $i \in \mathcal{M}^{\text{CS}}$  do
18:       CS  $i$  solves local optimization problem sub-P1 $_i$ 
19:       CS  $i$  generates  $m_i$ , where  $m_i^C \leftarrow Q_{i,t}^B(k+1)$ 
20:     end for
21:     Running Step 1 to Step 32 in Algorithm 1
22:     CS  $i \in \mathcal{M}^{\text{CS}}$  gets  $\hat{Q}_{i,t}^B(k+1)$  and  $\lambda_{i,t}(k+1)$  according to  $B^V$ 
23:      $k \leftarrow k + 1$ 
24:   until  $r_1(k) \leq \epsilon_1$  and  $s_1(k) \leq \epsilon_2$ 
25:   //Solving problem P2 in CPQ trading stage
26:   Initialization:  $k = 1$ ,  $\hat{\pi}_{i,t}^B(k) = \gamma_{i,t}(k) = 0$ ,  $i \in \mathcal{M}^{\text{CS}}$ 
27:   repeat
28:     for  $i \in \mathcal{M}^{\text{CS}}$  do
29:       CS  $i$  solves local optimization problem sub-P2 $_i$ 
30:       CS  $i$  generates  $m_i$ , where  $m_i^C \leftarrow \pi_{i,t}^B(k+1)$ 
31:     end for
32:     Running Step 1 to Step 32 in Algorithm 1
33:     CS  $i \in \mathcal{M}^{\text{CS}}$  gets  $\hat{\pi}_{i,t}^B(k+1)$  and  $\gamma_{i,t}(k+1)$  according to  $B^V$ 
34:      $k \leftarrow k + 1$ 
35:   until  $r_2(k) \leq \epsilon_3$  and  $s_2(k) \leq \epsilon_4$ 
36: end if

```

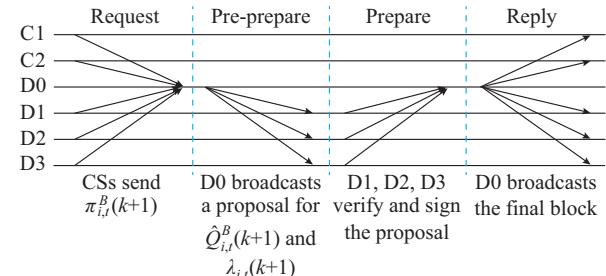


Fig. 4. Communication pattern on blockchain-based implementation when solving problem P2 in CPQ trading stage.

IV. SIMULATION RESULTS

A. Data

A simulation is carried out on a distribution network with 780 houses [27], where each house has one EV, mimicking a scenario with high EV penetration. All 780 EVs are equally assigned to 20 CSs. We consider 96 time intervals in each 24-hour day. The total permissible charging load in each time interval is determined by the capacity of the transformer at the root of the network and the conventional load curve provided by Fig. 2 in [27]. The charging welfare function (the charging service income minus the charging power reduction compensation) and charging constraints of a CS, and the parameters of EVs, CSs, and the distribution network are provided in our previous work [28]. Five CSs are randomly selected as the delegates in the simulations.

B. Coordination Results

Figure 5 illustrates the total load curves with or without the proposed EV charging coordination approach. In an uncoordinated scenario, the total load in the network exceeds the limit in peak hours. With the proposed EV charging coordination approach, the charging load in peak hours would be shifted to neighboring hours, avoiding the overload of the facility.

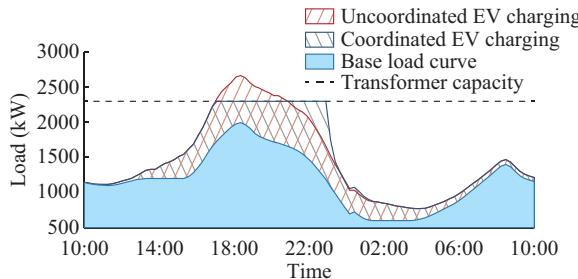


Fig. 5. Total load curve with or without EV charging coordination.

The total charging welfare before and after CPQ trading during peak hours is illustrated in Fig. 6. As shown in Fig. 6, the CPQ trading mechanism improves the total charging welfare. In addition, the charging schedule derived by the proposed mechanism is the same as the one derived by the centralized optimization.

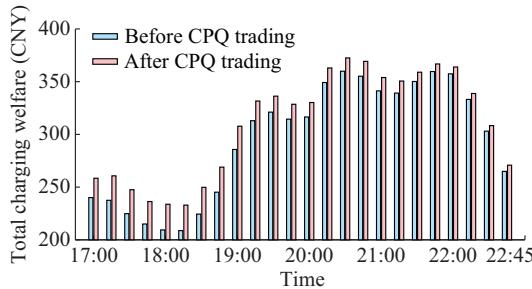


Fig. 6. Total charging welfare before and after CPQ trading.

Figure 7 shows the convergence performance of the ADMM algorithms in the time interval which has the longest iteration process. Solving problem P1 takes about 50 itera-

tions, and solving problem P2 takes about 140 iterations. The algorithms converge to the global optimum rapidly.

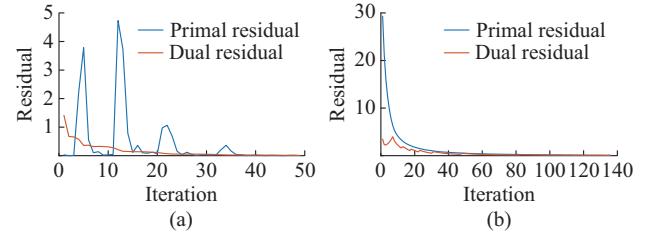


Fig. 7. Convergence performance of ADMM algorithms. (a) Convergence performance in problem P1. (b) Convergence performance in problem P2.

The CPQ trading results in a time interval, i.e., 19:00-19:15, are demonstrated in Fig. 8. As shown in Fig. 8, all rational CSs participate in CPQ trading. CSs buy/sell CPQs from/to other CSs according to their local charging power demands. And the Nash bargaining solution ensures that all CSs equally share the increase of the charging welfare.

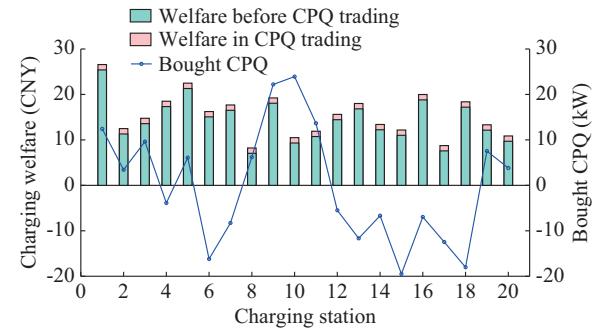


Fig. 8. CPQ trading result in a time interval.

C. Effectiveness of SDBFT Consensus Algorithm

Figure 9 illustrates the consensus process of the SDBFT consensus algorithm, where C1 and C2 represent a DSO and all non-delegate CSs, respectively; D0-D4 are the delegates; and D0 is the leader in this view.

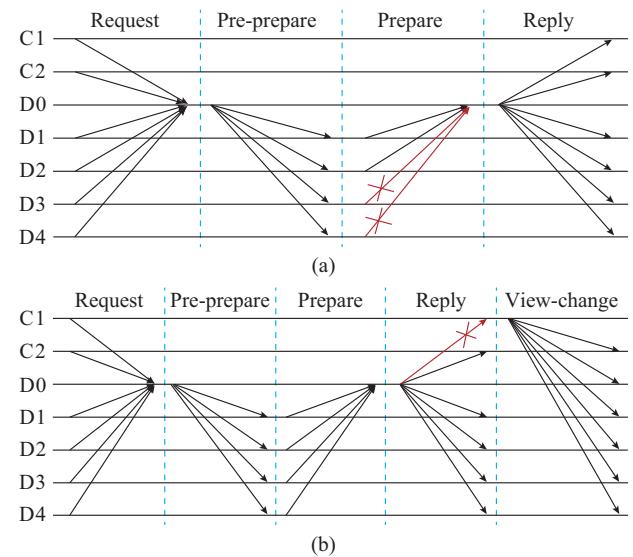


Fig. 9. Consensus processes within a view for normal-case operation and faulty-leader operation. (a) Normal-case operation. (b) Faulty-leader operation.

In Fig. 9(a), D3 and D4 are faulty delegates who do not send the prepare messages to D0 on time (denoted by the red lines). But D0 can still generate a final block with more than $|\mathcal{M}^d|/2$ signatures (provided by D0-D2). Consequently, all CSs can reach an agreement for the coordination result as usual. In Fig. 9(b), D0 is a faulty leader who does not send the reply message to C1 in time. Then, C1 sends a view-change message to all CSs, triggering the change of the leader. The next delegate, i.e., D1, becomes the new leader and a new view starts. Hence, the SDBFT consensus algorithm ensures the safety of the coordination process as long as the faulty delegates are the minority.

V. CONCLUSION

This paper proposes a fully autonomous two-stage EV charging coordination approach. The CPQ pre-allocation stage guarantees the security of the facility. The Nash bargaining-based CPQ trading stage makes the Pareto improvement. ADMM algorithms and a blockchain-based implementation are proposed to enable decentralized coordination. From the DSO perspective, the proposed approach can derive the optimal charging schedule while satisfying facility constraints. The proposed decentralized algorithm can work without a central coordinator, enabling fully autonomous coordination. From an individual perspective, the Nash bargaining solution ensures the fairness of charging welfare allocation. The ADMM-based decentralized algorithms protect individual privacy. The blockchain implementation provides trusted coordination.

REFERENCES

- [1] Z. Li, Q. Guo, H. Sun *et al.*, “A new real-time smart-charging method considering expected electric vehicle fleet connections,” *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 3114-3115, Nov. 2014.
- [2] L. Hua, J. Wang, and C. Zhou, “Adaptive electric vehicle charging coordination on distribution network,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2666-2675, Nov. 2014.
- [3] B. Sun, Z. Huang, X. Tan *et al.*, “Optimal scheduling for electric vehicle charging with discrete charging levels in distribution grid,” *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 624-634, Mar. 2018.
- [4] C. S. Antúnez, J. F. Franco, M. J. Rider *et al.*, “A new methodology for the optimal charging coordination of electric vehicles considering vehicle-to-grid technology,” *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 596-607, Apr. 2016.
- [5] Z. Liu, Q. Wu, S. S. Oren *et al.*, “Distribution locational marginal pricing for optimal electric vehicle charging through chance constrained mixed-integer programming,” *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 644-654, Mar. 2018.
- [6] L. Gan, U. Topcu, and S. H. Low, “Optimal decentralized protocols for electric vehicle charging,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940-951, May 2013.
- [7] Z. Tan, P. Yang, and A. Nehorai, “An optimal and distributed demand response strategy with electric vehicles in the smart grid,” *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 861-869, Mar. 2014.
- [8] Y. He, B. Venkatesh, and L. Guan, “Optimal scheduling for charging and discharging of electric vehicles,” *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095-1105, Sept. 2012.
- [9] C. Shao, X. Wang, M. Shahidehpour *et al.*, “Partial decomposition for distributed electric vehicle charging control considering electric power grid congestion,” *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 75-83, Jan. 2017.
- [10] H. Fan, C. Duan, C.-K. Zhang *et al.*, “ADMM-based multiperiod optimal power flow considering plug-in electric vehicles charging,” *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3886-3897, Jul. 2018.
- [11] J. Li, C. Zhang, Z. Xu *et al.*, “Distributed transactive energy trading framework in distribution networks,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7215-7227, Nov. 2018.
- [12] S. Fan, Q. Ai, and L. Piao, “Bargaining-based cooperative energy trading for distribution company and demand response,” *Applied Energy*, vol. 226, pp. 469-482, Sept. 2018.
- [13] H. Wang and J. Huang, “Incentivizing energy trading for interconnected microgrids,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2647-2657, Jul. 2018.
- [14] F. Luo, Z. Y. Dong, G. Liang *et al.*, “A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain,” *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 4097-4108, Sept. 2019.
- [15] S. Noor, W. Yang, M. Guo *et al.*, “Energy demand side management within micro-grid networks enhanced by blockchain,” *Applied Energy*, vol. 228, pp. 1385-1398, Oct. 2018.
- [16] Z. Li, J. Kang, R. Yu *et al.*, “Consortium blockchain for secure energy trading in industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690-3700, Aug. 2018.
- [17] J. Kang, R. Yu, X. Huang *et al.*, “Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154-3164, Dec. 2017.
- [18] Z. Su, Y. Wang, Q. Xu *et al.*, “A secure charging scheme for electric vehicles with smart communities in energy blockchain,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4601-4613, Jun. 2019.
- [19] X. Huang, C. Xu, P. Wang *et al.*, “LNSC: a security model for electric vehicle and charging pile management based on blockchain ecosystem,” *IEEE Access*, vol. 6, pp. 13565-13574, Mar. 2018.
- [20] X. Huang, Y. Zhang, D. Li *et al.*, “An optimal scheduling algorithm for hybrid EV charging scenario using consortium blockchains,” *Future Generation Computer Systems*, vol. 91, pp. 555-562, Feb. 2019.
- [21] J. F. Nash, “The bargaining problem,” *Econometrica*, vol. 18, no. 2, pp. 155-162, Apr. 1950.
- [22] S. Boyd, N. Parikh, E. Chu *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” in *Foundations and Trends in Machine Learning*, vol. 3. New York: Now Foundations and Trends, 2010, pp. 1-122.
- [23] S. L. Wang and L. Z. Liao, “Decomposition method with a variable parameter for a class of monotone variational inequality problems,” *Journal of Optimization Theory and Applications*, vol. 109, no. 2, pp. 415-429, May 2001.
- [24] W. Sun, F. Bränström, and E. G. Ström, “Network synchronization for mobile device-to-device systems,” *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1193-1206, Mar. 2017.
- [25] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.
- [26] M. Castro and B. Liskov, “Practical byzantine fault tolerance,” in *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*, New Orleans, USA, Feb. 1999, pp. 173-186.
- [27] C.-K. Wen, J.-C. Chen, J.-H. Teng *et al.*, “Decentralized plug-in electric vehicle charging selection algorithm in power systems,” *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1779-1789, Dec. 2012.
- [28] J. Ping, Z. Yan, S. Chen *et al.*, “Coordinating EV charging via blockchain,” *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 3, pp. 573-581, May 2020.

Jian Ping received the B.E. and Ph.D. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2015 and 2020, respectively. He is currently a Postdoctoral Fellow with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include application of blockchain in power systems, transactive energy systems and electric vehicle charging management.

Zheng Yan received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1984, and the M.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1987 and 1991, respectively, all in electrical engineering. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include application of optimization theory in power systems and power markets and dynamic security assessment.

Sijie Chen received the B.E. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Electronic Infor-

mation and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He is a recipient of 2017 Young Elite Scientists Sponsorship Program by China Association for Science and Technology and a recipient of

2017 Shanghai Pujiang Talent Award. His research interests include energy blockchain, demand response, transactive energy system, and electricity market.