Energy Crowdsourcing and Peer-to-Peer Energy Trading in Blockchain-Enabled Smart Grids

Shen Wang, *Student Member, IEEE*, Ahmad F. Taha, *Member, IEEE*, Jianhui Wang, *Senior Member, IEEE*, Karla Kvaternik, and Adam Hahn, *Member, IEEE*

***Abstract*—The power grid is rapidly transforming, and while recent grid innovations increased the utilization of advanced control methods, the next-generation grid demands technologies that enable the integration of distributed energy resources (DERs)—and consumers that both seamlessly buy and sell electricity. This paper develops an optimization model and blockchain-based architecture to manage the operation of crowdsourced energy systems (CESs), with peer-to-peer (P2P) energy trading transactions (ETTs). An operational model of CESs in distribution networks is presented considering various types of ETT and crowdsourcees. Then, a two-phase operation algorithm is presented: Phase I focuses on the day-ahead scheduling of generation and controllable DERs, whereas Phase II is developed for hour-ahead or real-time operation of distribution networks. The developed approach supports seamless P2P energy trading between individual prosumers and/or the utility. The presented operational model can also be used to operate islanded microgrids. The CES framework and the operation algorithm are then prototyped through an efficient blockchain implementation, namely, the IBM Hyperledger Fabric. This implementation allows the system operator to manage the network users to seamlessly trade energy. Case studies and prototype illustration are provided.**

***Index Terms*—Blockchain, energy crowdsourcing, energy trading, peer-to-peer (P2P) energy management.**

I. Introduction

SMART grid technologies, such as microgrids and distributed energy resources (DERs), have drastically changed the way electricity is generated and consumed in two dimensions. First, the rapid increase in energy *prosumers* introduces new grid participants and provides a more decentralized and open power grid. Second, this changes the role of a system operator or utility from a power retailer to a service provider—renting transmission/distribution lines to prosumers, rather than solely selling units of energy. This paradigm shift requires the creation of new trusted software platforms, distributed operation/control algorithms, and computational methods to enable reliable grid operations, prosumer engagement, and incentivize utility business model innovations.

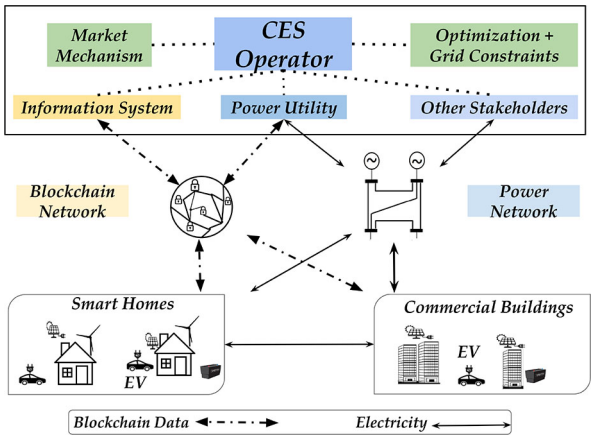


Fig. 1. Blockchain-assisted architecture of operation in CESs.

Crowdsourcing [2] is a major drive for various industries and has been utilized in various disciplines, such as medicine, cyber-physical systems, and engineering system design. The central theme in crowdsourcing is the utilization of the crowd's power to achieve system-level objectives. To see how crowdsourcing can be applied in energy systems, we provide an analogy from the most popular crowdsourcing markets, the Amazon Mechanical Turk (MTurk) [3], which enables people to post jobs with monetary rewards and expiry dates. Energy crowdsourcing offers the possibility of the transformation in energy systems, and this paper puts forth operational models of crowdsourced energy system (CES) for collaborative production and consumption in energy markets, shown in Fig. 1. The tasks in CES can be plugging in an electric vehicle, charging/discharging a battery, deferring loads, and supplying the power network with renewable energy via solar panels—with the objective of satisfying a near-real-time demand shortage/surplus. These tasks can be automated via smart inverters, plugs, and meters while interfacing with power utilities and a distributed blockchain implementation.

Manuscript received October 24, 2018; revised February 18, 2019; accepted May 4, 2019. Date of publication June 4, 2019; date of current version July 17, 2019. An earlier version of this paper was presented at the 2018 IEEE Power & Energy Society General Meeting in Portland, Oregon, August 5–9, 2018. A preprint of the conference paper can be found in [1], which shows the significant extensions and contributions of this paper in comparison with [1]. This paper was recommended by Associate Editor D. Yue. *(Corresponding author: Ahmad F. Taha.)*

S. Wang and A. F. Taha are with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78256 USA (e-mail: mvy292@my.utsa.edu; ahmad.taha@utsa.edu).

J. Wang is with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: jianhui@smu.edu).

K. Kvaternik is with Cyberphysical Systems Research, Siemens Corporate Technology, Princeton, NJ 08540 USA (e-mail: karla.kvaternik@siemens.com).

A. Hahn is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA (e-mail: ahahn@eecs.wsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2019.2916565

This transformation in sustainable energy systems, where energy management is crowdsourced by prosumers, will be supported by two key, disruptive scientific technologies.

1) New modeling and crowdsourcing-centered methods that perform real-time grid management while maintaining the grid's stability.
2) A secure cyber-infrastructure design to manage and coordinate millions of energy-trading transactions (ETTs) (*prosumer–prosumer* or *prosumer–operator trades*).

The majority of the new modeling methods are based on optimal power flow (OPF) operation models and the secure cyber-infrastructure design is implemented by the promising blockchain technology. However, both the new modeling methods and the implementation of blockchain have limitations. First, the computed OPF setpoints for DERs and controllable loads might not be eventually adopted by crowdsourcees and prosumers. Second, it is unclear how energy trading between prosumers can take place within the operational models. Third, the utilized blockchain architectures are not scalable to include millions of ETTs—especially that blockchain-based trades consume a significant amount of energy. This paper addresses these gaps, and the main contributions and organization are given as follows.

1) An operational framework and model of CESs in distribution networks is presented considering various types of ETT and crowdsourcees. The presented framework enables peer-to-peer (P2P) energy trading at the distribution level, where ubiquitous distribution-level asset owners can trade with each other. This has not done before in association with distributed OPF routines and blockchain-enabled architecture. In such a framework, an operator is needed to clear the market and ensure there is no violation of any technical constraints (e.g., distribution line limits). A distribution system operator can assume this role running the presented CES operational model (Section III). Extensions to operator-free, islanded microgrids are also showcased.
2) A two-phase, near real-time operation algorithm for CESs is explored. The first phase focusing on the day-ahead scheduling of generation and controllable DERs manages the bulk of grid-operation, while the second phase is developed to balance hour-ahead even real-time deficit/surplus in energy via monetary incentives. The developed two-phase algorithm supports arbitrary P2P energy trading between prosumers and utility, resulting in a systematic way to manage distribution networks amid P2P energy trading while incentivizing crowdsourcees to contribute to this ecosystem. The algorithm supports operation of islanded, self-autonomous microgrid (Section IV).
3) The CES framework is implemented and prototyped within IBM Hyperledger Fabric platform—an efficient blockchain implementation. This implementation allows the system operator to manage the network and supports users to log in, manage their own account, and carry on the energy trading with utilities or neighborhoods. This prototype communicates with the two-phase algorithm presented in this paper, is open source, and can be used by utilities (Section V). Finally, numerical tests on a distribution network and blockchain prototype illustration are provided (Section VI).

II. Literature Review

A. Grid Operation, OPF, and Demand Response

Recent studies have investigated integrating the operation of DERs in distribution networks. The focus of majority of these studies [4], [5] is on unit commitment, economic dispatch problems, scheduling of DERs, and maintaining the grid's frequency and voltage within acceptable ranges while given uncertainty from renewables and load forecasts.

Another branch of related work [6] studies the design of demand response signals and incentives to drive DER owners to contribute to energy production. In summary, there are three approaches to demand response: 1) reducing demand by using local DERs; 2) reducing demand through shifting controllable loads; and 3) designing efficient generator setpoints to reduce the total generation [7]. The majority of demand response schedules focus on operational timescale. Further, the need for real-time regulation and distributed dynamic pricing as a function of the grid's physical status motivates new physics-aware pricing mechanisms [8], [9]. Background on blockchain and energy trading routines is given in the next section.

B. Blockchain and Energy Trading Systems

Blockchain is a distributed ledger based on a set of communication and consensus protocols that ensure the ledger integrity through interlinked, cryptographically signed, and time-stamped blocks that define transactions [10]. The blockchain concept originated with the Bitcoin protocol, which utilized a proof of work (PoW) consensus mechanism where miners combine transactions into Merkle tree-based blocks and compete to find a random nonce that produces a hash digest within a predefined range. However, this approach has many limitations, including its significant energy consumption, scalability in the number of transactions/seconds, privacy concerns with a public ledger, and single purpose application (i.e., an exchange of the Bitcoin cryptocurrency [19]). A number of additional blockchain technologies have been introduced to address these challenges as suggested below.

1) *Efficient Consensus Mechanisms:* A consensus protocol is used to ensure the unambiguous ordering of transactions and guarantees the integrity and consistency of the blockchain across distributed nodes [20]; the annual estimated electricity consumption of Bitcoin PoW consensus is 47.1 Terawatt-hour—a staggering 0.21% of world's electricity consumption [21]. Furthermore, PoW techniques typically have limitations on the number of transactions per second, which limits the use in high performance environments. Other consensus mechanisms, such as Proof of Stake (e.g., Ethereum Casper [22]) or Redundant Byzantine Fault Tolerance (RBFT) (e.g., IBM Hyperledger Fabric [23]), can be used to reduce energy consumption.
2) *Smart Contracts:* Smart contracts provide protocols and Turing complete virtual machines that enable nodes to execute some program based on the results of new transactions and allow the blockchain to support sophisticated

TABLE I
VARIOUS IMPLEMENTATIONS OF BLOCKCHAIN. POW AND RBFT STAND FOR PROOF OF WORK AND REDUNDANT BYZANTINE FAULT TOLERANCE

	Bitcoin	*Ethereum*	*Hyperledger Fabric*
Cryptocurrency	Bitcoin	Ether	None
Network	public	public	permissioned
Transactions	anonymous	anonymous	public/confidential
Consensus	PoW	PoW	RBFT
Smart Contracts	None	Solidity	Chaincode
Language	C++	C++/Golang	Golang/Java

logic. Smart contracts and blockchain provide an excellent platform to perform ETT. In particular, Licata [24] provided a high-level description to the main merits of using cryptocurrency and blockchain in energy systems.

3) *Permissioned and Privacy Mechanisms:* Blockchain platforms can be categorized into public and private, where public implies that any miner can contribute to the consensus and block creation, while permissioned chains restrict block creation to a predefined set of parties. Therefore, permissioned chains may be preferred in applications with defined authorities or entities with management responsibilities.

Table I summarizes the attributes of different implementations of current blockchains, and Section V provides additional discussion on why the Hyperledger platform is selected to implement the proposed CES scheme.

In Table II, various focuses of recent P2P energy trading routines are compared according to focus aspects; the first three aspects are derived from [25]. These aspects reflect corresponding modules in Fig. 1, and are explained here. First, the *market mechanism*, including the participant setup, and pricing mechanism is designed to incentivize participants while maximizing the social welfare. The participant setup defines market participants and the form of energy trading, while pricing mechanism, i.e., incentive design and bidding strategy, comprises of the markets allocation and payment rules. Second, the *information system* is designed to connect all market participants, provide the market platform, offer market access, and monitor the market operations. Nowadays, blockchain is suitable to implement part of *information system*. Third, the *optimization and grid constraints* refer to scheduling of DERs while maintaining the grid in an optimal way as we discussed in Section II-A. Our corresponding implementation of the above aspects are presented in Section V-B. Finally, as for the *scenario* in these papers, we notice that most papers focus on microgrids and electric vehicles.

After comparing the typical papers in Table II, we notice the following. First, [11]–[14] and [17] focus more on approach to manage the grid with the assistance of simple negotiation, auction, or bidding mechanism and implementing the information system via thriving blockchain technology, since the security and privacy can be guaranteed. Specifically, the contribution of [12] is more about the multiagent system-based trading negotiation mechanism. Su *et al.* [13] proposed a contract-based blockchain for secure EV charging, and a reputation-based Byzantine fault-tolerance consensus algorithm is proposed. In [17], the new and hybrid charging scenario, i.e., mobile charging vehicle-to-vehicle, and grid-to-vehicle are considered. Second, Paudel *et al.* [15], Long *et al.* [16], and Hahn *et al.* [18] paid attention on designing different marketing/pricing mechanism, but the power flow model is ignored in their optimization. For example, game theoretical approaches are adopted to achieve real-time pricing in [15], [26], and [27]. Besides the typical paper listed above, the attack/threat model are explored further in energy blockchain in [28] and [29] to enhance the security and privacy. Especially in [29], Zhu *et al.* designed a special trust authority node with a veto power to prevent malicious voting. However, the marketing/pricing mechanism and platform design for P2P energy markets do not receive too much attention and still are an open research area.

Beyond research-oriented studies, companies (i.e., [30]–[32]) mainly focus on the development of business models, and the possibility of introducing those models to local energy market and design of control systems are not fully considered.

III. INTEGRATED OPERATIONAL MODEL OF CESS

In this section, we present an integrated operational model of CESs that considers a wide range of DERs, different types of crowdsourcees and ETT in distribution networks. For simplicity, we focus on radial distribution networks with a single feeder connected to traditional generation and utility-scale renewables. We consider a CES at the feeder level with n buses modeled by a tree graph $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of lines. Define the partition $\mathcal{N} = \mathcal{G} \bigcup \mathcal{C} \bigcup \mathcal{L}$, where $\mathcal{G} = \{1, \ldots, n_g\}$ collects the n_g utility-scale power generation connected to the feeder/substation; $\mathcal{C} = \{1, \ldots, n_c\}$ collects the buses containing n_c users who signed up for crowdsourcing schedules; $\mathcal{L} = \{1, \ldots, n_l\}$ collects load buses.

The *crowdsourcer*, one type of participants, here is the utility company or any other system operator, we distinguish between two types of crowdsourcees in $\mathcal{C}$. Type 1 crowdsourcees commit in the day-ahead markets (and perhaps monthly or yearly) to the crowdsourcing tasks requested by the operator. Type 1 crowdsourcees also include users who give complete control of their DERs to the operator. In return, the operator provides socio-economic incentives or discounts on the electric bill. Type 2 crowdsourcees provide near real-time adjustments or decisions based on real-time notifications and decisions from the operator. For example, the operator informs Type 2 crowdsourcees about the *crowdsourced task* (e.g., charging/discharging an electric vehicle) which depends on the users' location in the network and the physical state of the grid. Type 1 crowdsourcees provide operators with day-ahead planning flexibility, in contrast with Type 2 crowdsourcees who operate on a faster timescale. The distinction between these two types of users is needed as it resembles projected market setups [33]. We define these two types as $\mathcal{CT}_1$ and $\mathcal{CT}_2$, with $\mathcal{C} = \mathcal{CT}_1 \bigcup \mathcal{CT}_2$; this is depicted in Fig. 2.

We consider two types of ETT. *Type A*: This is akin to what takes place in today's grids, where Type 1 or 2 crowdsourcees feed the grid with power. This type of transaction is solely between crowdsourcees and the network operator. *Type B*: Crowdsourcees can trade energy with each other where the seller injects power into the grid. Fig. 3 shows the types of crowdsourcees and transactions. Since energy production

TABLE II
VARIOUS FOCUSES OF TYPICAL P2P ENERGY TRADING SYSTEMS

Reference	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]
Market Mechanism	◔	◔	◓	◔	●	◓	◔	◓
Information System	Blockchain	Blockchain	Blockchain	Blockchain	○	○	Blockchain	Blockchain
(Consensus)	PoW(Public)	Self-designed	Self-designed	NA	NA	NA	NA	PoW
Optimization + Grid Constraints	●	●	●	●	◔	●	●	○
Scenario	Microgrid	EV	EV	EV	Microgrid	Microgrid	EV	Microgrid

○ means *not considered*; ◔ means *partially considered*; ● means *fully considered*; NA means *not applicable* or the authors do not cover the aspect; *Self-designed* means that authors design a new, corresponding consensus mechanism for their own blockchain implementation;

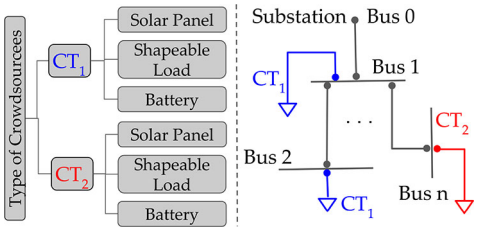


Fig. 2. Radial network with different types of crowdsourcees: $\mathcal{CT}_1$ (blue) and $\mathcal{CT}_2$ (red).

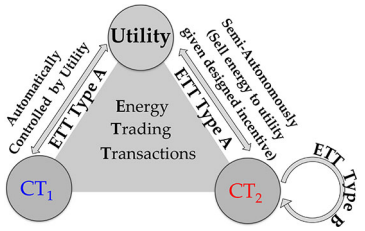


Fig. 3. Types of crowdsourcees and ETTs.

and demand response from Type 1 crowdsourcees are controlled by the operator, Type B transactions only occur among Type 2 crowdsourcees. However, Type A transactions can also take place between Type 2 crowdsourcees and the utility. The participants and the transaction types are showed in detail in Fig. 3. The Brooklyn Microgrid [34] project is an example of Type B transactions for Type 2 crowdsourcees.

A. Operational Model of Generators, Loads and DERs

Let $i \in \mathcal{N}$ denote the bus index of the distribution system and t denote the time period. We consider bulk, dispatchable generation from traditional synchronous generators, renewable energy generation from solar panels, fully controllable stationary batteries, uncontrollable loads, and shapeable loads.

1) Dispatchable Generators: Dispatchable generators are considered in this paper with a quadratic cost function. Dispatchable generation $S^g_{i,t} = P^g_{i,t} + jQ^g_{i,t}$ for $i \in \mathcal{G}$ at t are considered to have quadratic cost functions as $C_{i,t}(P^g_{i,t}) = \alpha_{i,t}(P^g_{i,t})^2 + \beta_{i,t}P^g_{i,t} + \gamma_{i,t}$, where $\alpha_{i,t}$, $\beta_{i,t}$, and $\gamma_{i,t}$ are given parameters for the cost function of the ith generator at t.

2) Solar Energy Generation: Solar panels generate real power $P^r_{i,t}$ for bus $i \in \mathcal{C}$ at t. Note that $\mathcal{CT}_1$ crowdsourcees do not control whether $P^r_{i,t}$ is fed into the grid or not (it is controlled by the utility/operator), whereas $\mathcal{CT}_2$ crowdsourcees dictate whether to use $P^r_{i,t}$ locally or sell it to the CES operator or other users.

3) Stationary Batteries: Batteries are modeled as dispatchable loads that can be controlled to withdraw or inject power. The quantity $P^b_{i,t}$ defines the output power of the batteries, where $i \in \mathcal{C}$. Negative $P^b_{i,t}$ implies that power is withdrawn. The battery operational model [11] is described as

$$E^b_{i,t} = E^b_{i,t-1} + H^b_{i,t}\eta_{i,\text{in}} - D^b_{i,t}/\eta_{i,\text{out}} \tag{1a}$$
$$P^b_{i,t} = D^b_{i,t} - H^b_{i,t} \tag{1b}$$
$$0 \le D^b_{i,t} \le P^b_{i,t,\text{dis}} \tag{1c}$$
$$0 \le H^b_{i,t} \le P^b_{i,t,\text{cha}} \tag{1d}$$
$$E^{b,\min} \le E^b_{i,t} \le E^{b,\max}. \tag{1e}$$

In the above battery model, we consider a unit time period; $\eta_{i,\text{in}}$ and $\eta_{i,\text{out}}$ represent charging and discharging efficiency constants. $H^b_{i,t}$ and $D^b_{i,t}$ is the charging and discharging power—both are optimization variables. The variable $E^b_{i,t}$, upper and lower bounded by $E^{b,\min}$ and $E^{b,\max}$, denotes the energy stored in battery at time t. The net power $P^b_{i,t}$ at t is the difference between the power of discharging and charging. $P^b_{i,t,\text{dis}}$ stands for the limitation of discharging power, $P^b_{i,t,\text{cha}}$ has a similar meaning for charging power. All of the variables related to batteries model are included in a single vector variable $\boldsymbol{x}^b_{i,t} := (E^b_{i,t}, H^b_{i,t}, D^b_{i,t}, P^b_{i,t})$.

4) Uncontrollable Loads: Uncontrollable loads (lights, plug loads, street lights, etc.) are considered to be given and are denoted by $S^u_{i,t}$ for all $i \in \mathcal{L}$ (loads can include reactive power), where $S^u_{i,t} = P^u_{i,t} + jQ^u_{i,t}$.

5) Shapeable Loads: We consider shapeable loads, defined by $S^s_{i,t} = P^s_{i,t} + jQ^s_{i,t}$ for $i \in \mathcal{L}$, such as plug-in electric vehicles and loads from appliances with flexible power profile but fixed energy demand $E^s_{i,\text{demand}}$ in 24 h. These shapeable loads must be satisfied between $t_{i,\text{start}}$ and $t_{i,\text{end}}$. The model describing the shapeable loads [11] is given next

$$E^s_{i,\text{demand}} = \sum_{t=1}^{T} S^s_{i,t}\Delta t \tag{2a}$$
$$S^s_{i,t} = 0, \text{ for } t = 1, \ldots, t_{i,\text{start}}, t_{i,\text{end}}, \ldots, T \tag{2b}$$
$$S^{s,\min}_i \le S^s_{i,t} \le S^{s,\max}_i \tag{2c}$$

where T is the length of the time-horizon and Δt is the time interval. Similarly, a single vector variable $\boldsymbol{x}^s_{i,t} := (S^s_{i,t})$ collects variables related to shapeable loads.

B. Distribution Network Model

For each bus $i \in \mathcal{N}$, denote $V_i = |V_i|e^{j\theta_i}$ as its complex voltage and $v_i = |V_i|^2$ as its magnitude squared. Let $s_i = p_i + jq_i$ be node i's net complex power injection. Also, p_i denotes net real power injection. From Section III-A, the net real power injection for each bus i at t can be expressed as

$$p_{i,t} = P^g_{i,t} + P^b_{i,t} + P^r_{i,t} - P^u_{i,t} - P^s_{i,t}. \tag{3}$$

Similarly for the net reactive power injection. For each line $i \in \mathcal{E}$, we denote bus i's parent and children buses as A_i and C_i. Let $z_i = r_i + jx_i$ be its complex impedance, I_i be the complex branch current from bus i to A_i, and $l_i = |I_i|^2$ be its magnitude squared. The variable $S_i = P_i + jQ_i$ denotes the branch power flow from bus i to A_i. For all buses in the network, define $\boldsymbol{x}_t \coloneqq (\boldsymbol{x}^b, \boldsymbol{x}^s)_t$ as a variable vector collecting the variables related to batteries and shapeable loads. Since two types of crowdsourcees are defined, $\boldsymbol{x}_t$ is divided into two variables $\boldsymbol{x}_{1_t}$ and $\boldsymbol{x}_{2_t}$, which stands for the variables belong to Type 1 and Type 2 crowdsourcees and hence $\boldsymbol{x}_t = (\boldsymbol{x_1}, \boldsymbol{x_2})_t$. Let $\boldsymbol{y}_t \coloneqq (P^u_{i,t}, P^r_{i,t})$ be a variable vector collecting the variables related to uncontrollable loads and solar energy. The preferences and setting parameters of crowdsourcees, including the willingness to sell energy, constants related to batteries, solar panel, or loads are communicated with the utility or the operator are denoted by $\mathcal{X}_t$.

To model power flow in distribution networks, we use the branch flow model [35], [36]. This model eliminates the phase angles of V_i and I_i and uses only (v_i, l_i, s_i, S_i)

$$v_{A_i} = v_i - 2(r_iP_i + x_iQ_i) + \ell_i\left(r_i^2 + x_i^2\right) \quad i \in \mathcal{E} \tag{4a}$$

$$\sum_{j \in C_i}\left(P_j - \ell_j r_j\right) + p_i = P_i \quad i \in \mathcal{N} \tag{4b}$$

$$\sum_{j \in C_i}\left(Q_j - l_j x_j\right) + q_i = Q_i \quad i \in \mathcal{E} \tag{4c}$$

$$P_i^2 + Q_i^2 = v_i\ell_i \quad i \in \mathcal{E}. \tag{4d}$$

Due to (4d), the branch flow model is not convex. However, the model can be convexified using the second-order cone program (SOCP) relaxation [37] and rewritten as

$$\left\| \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} P_i \\ Q_i \\ v_i \\ l_i \end{bmatrix} \right\| \leq \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_i \\ Q_i \\ v_i \\ l_i \end{bmatrix}. \tag{5}$$

The nonconvex branch flow model can be cast through convex SOCP constraints denoted by CvxFlowModel($\boldsymbol{z}_t$) that collects (4a)–(4c) and (5), and can be solved efficiently by interior-point method in polynomial time [38]. In this paper, all branch flow variables are collected in a single vector variable $\boldsymbol{z}_t \coloneqq (\boldsymbol{v}, \boldsymbol{l}, \boldsymbol{s}, \boldsymbol{S})_t$ at time t. Table III lists all variables introduced in this paper. The next section introduces the CES OPF formulation and incentive design.

TABLE III
NOTATION FOR VARIOUS DERS IN CES*

Symbols	*Description*
$S^g_{i,t}$	Dispatchable generation
$P^r_{i,t}$	Real power generated from solar panel
$P^b_{i,t}$	Output power of the battery
$S^u_{i,t}$	Apparent power of uncontrollable load
$S^s_{i,t}$	Apparent power of shapeable load
$p_{i,t}$	Net real power injection at each bus
$\boldsymbol{x}^b_{i,t}$	A variable collecting all of the variables in battery model
$\boldsymbol{x}^s_{i,t}$	A variable collecting all of the variables in shapeable model
$\boldsymbol{x}_t$	A variable collecting variables in battery and shapeable model
$\boldsymbol{y}_t$	A variable collecting the variables of uncontrollable loads and solar energy
$\boldsymbol{z}_t$	A variable collecting all of the branch flow variables
$\mathcal{X}_t$	Preferences and setting parameters of crowdsourcees

*Symbols with or without subscript i, t have the same meaning for simplicity.

TABLE IV
ETT TYPES AND THE CORRESPONDING IN RELEVANCE TO THE TWO-PHASE ALGORITHM

	Seller	*Buyer*	*Pricing Mechanism*	*Optimization Phase*
ETT Type A	$\mathcal{CT}_1$	Utility	Contract pricing	Phase I
ETT Type A	$\mathcal{CT}_2$	Utility	Incentive pricing	Phase II
ETT Type B	$\mathcal{CT}_2$	$\mathcal{CT}_2$	Negotiated pricing	Phase I

IV. CES-OPF AND INCENTIVES DESIGN

In this section, we propose a two-phase algorithm minimizing the cost of generation and thermal losses by rescheduling users' shapeable loads and DERs ahead of time. The algorithm also designs localized incentives that persuade users to participate in CES. In addition, the presented algorithm supports P2P ETT between different crowdsourcees and the utility. The developed two-phase algorithm supports arbitrary P2P energy trading between prosumers and utility, resulting in a systematic way to manage distribution networks amid P2P energy trading while incentivizing crowdsourcees to contribute to this ecosystem. The algorithm also supports the operation of islanded, self-autonomous microgrids. The algorithm is described next.

The first phase of the algorithm is akin to day-ahead scheduling given load, solar forecasts, which belongs to *optimization and grid constraints* in Section II-B. This phase takes into account the types of crowdsourcees and their day-ahead preferences as well as the prescheduled ETTs among crowdsourcees. Given the day-ahead solutions from the first phase, the second phase reflecting *market mechanism* in Section II-B performs two significant operations. First, rectifying the mismatch in the day-ahead forecasts and hence the demand shortage/surplus by: 1) obtaining more accurate, hour-ahead forecasts and 2) solving for real-time deviations in the generator and DER setpoints. Second, allowing for real-time energy transactions through the design of monetary incentives that reward crowdsourcees. Table IV summarizes the ETT types in relevance to the two-phase algorithm. For different phases and users, the pricing mechanism also changes. Contract pricing is decided by contract between $\mathcal{CT}_1$ and utility, incentive pricing for $\mathcal{CT}_2$ is further explained in Section IV-B. Negotiated pricing is determined between the crowdsourcees and their neighbors. In short, the first phase manages the larger chunk of operations, whereas the second phase deals with the mismatch in load and renewable energy generation. The next two sections present the details of the two-phase algorithm.

A. Phase I: Day-Ahead CES Operation

As discussed in Section III, the network operator completely controls $\mathcal{CT}_1$ users' DERs according to the signed contract,

while $\mathcal{CT}_2$ users decide to participate or not in the crowdsourcing schedules based on their preferences and the offered incentives, e.g., $\mathcal{CT}_2$ users can sell their surplus solar power to the utility if designed incentive is sufficient or acceptable in the hour-ahead or real-time markets. This entails—and due to the nature of $\mathcal{CT}_2$ users—that the output from solar panels $P^r_{i,t}$, batteries $P^b_{i,t}$, and shapeable loads $P^s_{i,t}$ for users $i \in \mathcal{CT}_2$ are uncontrollable by the utility. Hence, if Type 2 crowdsourcees declare that they would not trade energy with other users (Type B transactions); then, in this phase, these quantities are excluded in (3) by setting them to zero yielding

$$P^r_{i,t} = P^b_{i,t} = P^s_{i,t} = 0, i \in \mathcal{CT}_2. \tag{6}$$

Otherwise, the sellers and buyers should send the energy supply–demand requests for P2P energy trading day ahead to the utility. These requests for $\mathcal{CT}_2$ users in time period t are expressed as constraint EnergyTrading$(\boldsymbol{x}_{2_t}, \boldsymbol{y}_t)$. This constraint ultimately transforms variables $\boldsymbol{x}_{2_t}, \boldsymbol{y}_t$ to mere predefined constants since the users decide to inject (or receive) a certain amount of energy into (from) the grid. The CES-OPF is formulated as

$$\begin{aligned}\text{CES-OPF: } \min_{\substack{\boldsymbol{x}_t, \boldsymbol{z}_t \\ \boldsymbol{P}^g_t}} \quad & \sum_{t=1}^{T} J_t(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{P}^g_t) \\ \text{s.t.} \quad & (1)-(3),\ (6), \boldsymbol{y}_t = \boldsymbol{y}^{f-24\text{hr}}_t, \boldsymbol{x}_t \in \mathcal{X}_t \\ & \text{CvxFlowModel}(\boldsymbol{z}_t), \boldsymbol{z}^{\min}_t \le \boldsymbol{z}_t \le \boldsymbol{z}^{\max}_t \\ & \boldsymbol{P}^g_t \in \mathcal{P},\ \text{EnergyTrading}(\boldsymbol{x}_{2_t}, \boldsymbol{y}_t). \end{aligned} \tag{7}$$

The objective function of CES-OPF at time t is defined as

$$J_t\big(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{P}^g_t\big) = \sum_{i=1}^{n_g} C_{i,t}\big(P^g_{i,t}\big) + \sum_{i=1}^{|\mathcal{E}|} l_{i,t} r_i + \sum_{i=1}^{|\mathcal{CT}_1|} U_i(x_t).$$

The objective is to minimize the generator's cost function, given by $\sum_{i=1}^{n_g} C_{i,t}(P^g_{i,t})$, in addition to the thermal losses that are characterized by $\sum_{i=1}^{|\mathcal{E}|} l_{i,t} r_i$, and crowdsourcees' disutility function $U_i(x_t) = u_i(S^s_{i,t} - S^{s,\max}_i)^2\ \forall t \le T_{\text{set}}$ designed to compensate for the inconvenience caused by rescheduling shapeable load. The parameter $u_i \in [0, 1]$ stands for the urgency to finish a certain task before a setting time T_{set}; the $S^{s,\max}_i$ is the same parameter appearing in (2); and u_i is the parameter determined by users through preferences $\mathcal{X}_t$.

The CES-OPF captures the cost of power losses between two peers through the second term of $J_t(\cdot)$ which sums the losses for all lines $\mathcal{E}$ in a distribution network. These lines include the distribution lines between any two users/peers, including traditional energy consumers. Preferences set by users are included in $\mathcal{X}_t$ and are assumed to be linear and time dependent; $\boldsymbol{y}^{f-24\text{hr}}_t$ is the day-ahead uncontrollable load and solar energy forecasts. Constants $\boldsymbol{z}^{\min}_t$ and $\boldsymbol{z}^{\max}_t$ are lower and upper bounds on branch flow model variable $\boldsymbol{z}_t$; i.e., the voltage in p.u. at each node is in [0.95 1.05]. The linear ramp constraints and upper/lower bounds on $\boldsymbol{P}^g_t$ are denoted by $\mathcal{P}$.

The CES-OPF can be decomposed into small optimization subproblems by decoupling variables and constraints—the overall problem can be then solved through a decentralized alternating direction method of multipliers (ADMMs) algorithm (see [39]). Another approach is to simply solve CES-OPF in a centralized fashion after requesting the user's preferences $\mathcal{X}_t$ ahead of time for medium- or small-scale distribution networks and microgrids. Another way of making CES-OPF more computationally tractable is to replace the convexified branch flow model with the LinDistFlow$(\boldsymbol{z}_t)$ model [40] which is linear in $\boldsymbol{z}_t$; this transforms CES-OPF to a quadratic program that can be solved for large-scale networks.

After solving CES-OPF, we obtain the equilibrium $S^{g,\text{eq}}_{i,t} = P^{g,\text{eq}}_{i,t} + jQ^{g,\text{eq}}_{i,t}$ and $\boldsymbol{x}^{\text{eq}}_{1_t}$ which includes $P^{b,\text{eq}}_{i,t}$ and $S^{s,\text{eq}}_{i,t}$. This entails that the utility-scale generation, batteries, and shapeable loads belonging to $\mathcal{CT}_1$ users will be fixed with this equilibrium for the next 24 h. To compensate crowdsourcees for their contributions, the distributed locational marginal price (DLMP)—the time-varying electricity price for users at various buses in the network—is computed by finding the dual variables associated with the real power balance constraint in the convexified branch flow model, and denoted by $\lambda^{\text{eq}}_{i,t}$.

B. Phase II: Real-Time CES Incentives Design

As outlined in Section IV-A, we solve CES-OPF and obtain setpoints for utility-scale power plants and Type 1 crowdsourcees, knowing that some energy trading transactions will take place between crowdsourcees. In this section, the presented crowdsourcing incentive design performs the two key functions: 1) incentivizes Type 2 crowdsourcees to sell excess solar power to the utility and 2) mitigates and balances the unexpected load and solar output fluctuations due to the forecast error in the grid. The formulation presented in this section is solved every hour or less, depending on the availability of hour-ahead forecasts and the operator's preference.

Here, we outline the design of crowdsourcing incentives that provide near real-time ancillary services to relieve real-time demand shortage or surplus—and hence, the additional incentives which based on the amount of energy provided to the grid are offered for $\mathcal{CT}_2$. For $i \in \mathcal{CT}_2$, the amount of energy provided to the grid is depicted by the net injection power $P^{\text{ni}}_{i,t}$ and computed as

$$P^{\text{ni}}_{i,t} = P^r_{i,t} - P^s_{i,t} + P^b_{i,t}, \quad i \in \mathcal{CT}_2. \tag{8}$$

This indicates when solar panels produce more power, and the shapeable load reduces, more net injected power can be sold to the utility or other crowdsourcees through energy trading. Here, for $i \in \mathcal{CT}_2$, shapeable loads and batteries cannot be scheduled 24 h ahead since no contract exists between Type 2 crowdsourcees and the utility. Hence, $P^s_{i,t}$ and $P^b_{i,t}$ belonging to variable $\boldsymbol{x}_{2_t}$ are treated now as uncontrollable loads for $\mathcal{CT}_2$ in Phase II. In addition, solar energy is also known ahead of time. Hence, $P^{\text{ni}}_{i,t}$ is known and not an optimization variable for Type 2 crowdsourcees from (8). The crowdsourcing incentive design routine for crowdsourcees i at time t is formulated as

$$\begin{aligned}\text{CES-ID: } \min_{\substack{\boldsymbol{x}_t, \boldsymbol{z}_t \\ \boldsymbol{P}^g_t, \boldsymbol{\lambda}^a_t \\ \boldsymbol{b}_t}} \quad & \sum_{i=1}^{n_g} C_{i,t}\big(P^g_{i,t} - P^{g,\text{eq}}_{i,t}\big) + \sum_{i=1}^{|\mathcal{E}|} l_{i,t} r_i + \sum_{i=1}^{|\mathcal{CT}_2|} b_{i,t} \\ \text{s.t.} \quad & (1)-(3),\ (8), \boldsymbol{x}_{1_t} = \boldsymbol{x}^{\text{eq}}_{1_t}, \boldsymbol{x}_{2_t} \in \mathcal{X}_{2_t} \\ & \boldsymbol{y}_t = \boldsymbol{y}^{f-1\text{hr}}_t, \boldsymbol{z}^{\min}_t \le \boldsymbol{z}_t \le \boldsymbol{z}^{\max}_t \end{aligned}$$

$$\begin{aligned}&\text{CvxFlowModel}(\boldsymbol{z}_t), P_t^g \in \mathcal{P}\\&b_{i,t} = P_{i,t}^{\text{ni}}\left(\lambda_{i,t}^{\text{eq}} + \lambda_{i,t}^{a}\right), b_{i,t} \geq 0, i \in \mathcal{CT}_2\\&\sum_{i=1}^{|\mathcal{CT}_2|} b_{i,t} \geq b_t^{\text{total}}, i \in \mathcal{CT}_2.\end{aligned} \tag{9}$$

In CES-ID, the objective is to minimize: 1) the deviation in the cost of generation from the day-ahead operating point; 2) the network's thermal losses; and 3) the budget $\sum_{i=1}^{|\mathcal{CT}_2|} b_{i,t}$ (in \$) which the operator has allocated to spend on the real-time incentives at the feeder level. The constraints are explained as follows. We set variables $P_{i,t}^{b}, S_{i,t}^{s} \in \boldsymbol{x}_{1_t}$ to the equilibrium $P_{i,t}^{b,\text{eq}}, S_{i,t}^{s,\text{eq}} \in \boldsymbol{x}_{1_t}^{\text{eq}}$ which is obtained by CES-OPF to schedule DERs that are controlled by the utility. For $i \in \mathcal{CT}_2$, the willingness to sell energy to the utility is set in preference $\mathcal{X}_{2_t}$ which sent to system operator. The constraints on $\boldsymbol{y}_t, \boldsymbol{z}_t, P_t^g$ are the same as CES-OPF (7) except that $\boldsymbol{y}_t$ is set to the hour-ahead (or shorter) available forecast $\boldsymbol{y}_t^{f-1\text{hr}}$.

Besides the optimization variables mentioned above, we consider that Type 2 crowdsourcees receive the final incentive price $\lambda_{i,t}^{\text{eq}} + \lambda_{i,t}^{a}$ where $\lambda_{i,t}^{a}$, additional variable, is an adjustment price which varies with the net energy injected to grid and location of $\mathcal{CT}_2$; $\lambda_{i,t}^{\text{eq}}$ is DLMP computed by CES-OPF. The variable budget $b_{i,t}$ for $i \in \mathcal{CT}_2$ at t is equal to $P_{i,t}^{\text{ni}}(\lambda_{i,t}^{\text{eq}} + \lambda_{i,t}^{a})$, which is always greater than 0. As mentioned $P_{i,t}^{\text{ni}}$, $\lambda_{i,t}^{\text{eq}}$ are known. When the crowdsourcee i has no energy to sell to utility ($P_{i,t}^{\text{ni}} \leq 0$), variable $\lambda_{i,t}^{a}$ is forced to approach $-\lambda_{i,t}^{\text{eq}}$ to make $b_{i,t}$ as 0^+ (a small positive value which is approximately close to zero). Hence, no incentive is offered to those who inject no power into the grid. When $P_{i,t}^{\text{ni}} > 0$ which means crowdsourcee i at t has excess energy to sell, variable $\lambda_{i,t}^{a}$ is forced to be small while also minimizing the final incentive price $\lambda_{i,t}^{\text{eq}} + \lambda_{i,t}^{a}$ and budget $b_{i,t}$ for all Type 2 crowdsourcees. At time t, the total budget for $\mathcal{CT}_2$ is b_t^{total}, which can be set as a reasonable value. For example, this can be set to the cost for dispatchable generation to produce $\sum_{i=1}^{|\mathcal{CT}_2|} P_{i,t}^{\text{ni}}$. Further explanations and examples are presented in Section VI-B.

Notice that both CES-OPF and CES-ID are based on branch flow model which is convex, and can be solved with great efficiency in polynomial time by interior-point optimizer. The CES-ID is solved hourly, and the computed incentives are sent to users at the end of the day. Thus, the energy trading (Type A transactions) between $\mathcal{CT}_2$ users and the utility is finished. The transactions are done by the assist of blockchain, which is explained in the next section.

V. Blockchain and Smart Contracts Implementation for CESs

In this section, we discuss an implementation for blockchain that is scalable to accommodate millions of crowdsourcees and ETT. An algorithm to integrate the optimization models in Section IV with this blockchain implementation is also presented.

A. Blockchain and Smart Contracts Implementation for CESs

While Table I summarizes the attributes of different blockchain platforms, this section identifies the properties most applicable for the proposed CES model and algorithms introduced in Sections III and IV. Specifically, the blockchain platform must adequately address the goals to incorporate a precise set of CES users, the computational requirements of the CES algorithms, the performance of the consensus algorithms, and the privacy demands of the users. The CES requirements and blockchain properties for each of these domains are identified in Table V. Based on this analysis, the Hyperledger is selected to meet the required CES requirements and necessary blockchain features. As previously mentioned, Hyperledger uses RBFT for consensus, which should minimize the energy required for each transaction. Furthermore, Hyperledger's permissioned model ensures that the participants are restricted to those within the distribution grids service region, and also prevents the exposure of privacy data from crowdsourcees. Finally, the smart contracts can be implemented through the `chaincode` mechanisms, which does not require the per-operation execution costs that are enforced on other public blockchains.

TABLE V
CES Requirements Mapping to Blockchain Features

	CES Requirements	Blockchain Features
Participants	The CES will be operated for a distribution grid, so users will be confined to a geographic area users	Permissioned chain as users should be restricted to those currently within that distribution area
Computation	CES must require performing non-linear optimizations such as solving power flow and economic dispatch	Efficient smart contracts requiring the ability to execute Turing complete programs on large quantities of data without heavy cost
Consensus	Minimal energy usage to ensure energy sustainability goals of CES	Avoid computationally expensive PoW consensus algorithms
Privacy	Crowdsourcee preferences and usages likely exposes privacy data	Permissioned model that protects crowdsourcee data from external observers

This, unlike other blockchain applications, still requires a central authority—the utility company or the system operator to manage the grid, provide technical supports for each small-scale energy trading, clear the market, and ensure there is no violation of any technical constraints (e.g., distribution line limits). Small-scale energy trading without a central authority can take place (see [41]), yet the scaling of these transactions to include thousands of people and millions of daily energy transactions without the utility coordinating the communication among small-scale energy trading systems is remote in today's markets. To this end, the presented architecture in this paper requires a central authority to manage the grid but can also autonomously be run in islanded microgrids as we showcase in the case studies section.

B. Blockchain Implementation Using Hyperledger Fabric

We integrate and implement blockchain and smart contracts with the optimization models given in Section IV. This is shown in Fig. 4. The presented CES implementation consists of three modules—surrounded by the dotted lines in Fig. 4. Module I, corresponding to *optimization and grid constraints* in Section II-B, includes the optimization problems in Section IV which are coded by `CVXPY` [42]. Module II is

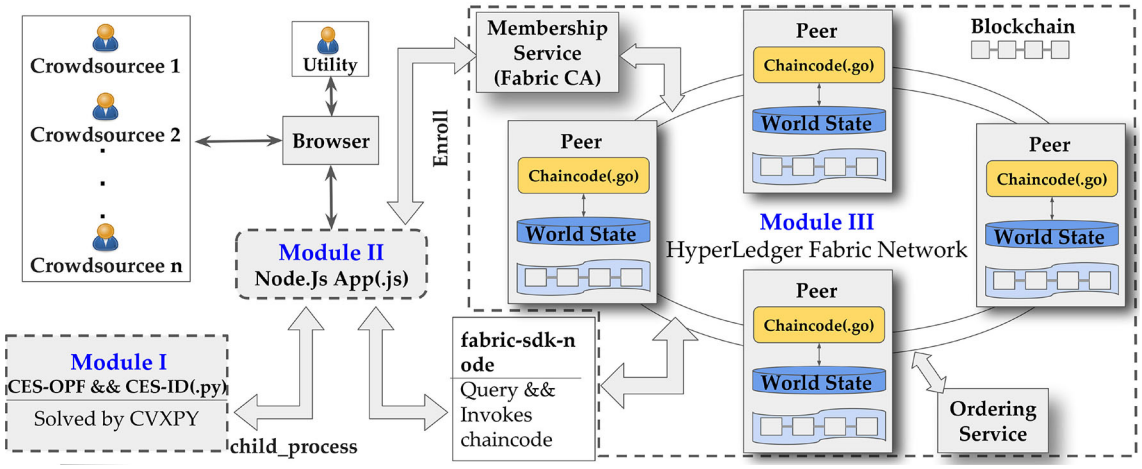


Fig. 4. Architecture of combining blockchain and smart contract with the optimization formulations presented in this paper.

a `Node.js` application, also take care of the communication between Python-written optimization problem and Module III. This process is finished by the `child_process` standard library which generates a Python process and computes the solutions to CES-OPF (7), CES-ID (9) while returning results back to `Node.js` program.

Module III, the *information system* in Section II-B, is implemented by the IBM Hyperledger Fabric network deployed in cloud to provide the blockchain service. The network consists of many peers that communicate with each other, runs smart contracts called `chaincode` which is written by Go language, holds state and ledger data. Peers in the Hyperledger Fabric Network are different from the ones in the other blockchain implementations. The roles of peers relate to the life-cycle of transactions which is one key difference between Hyperledger Fabric and many other blockchain platforms. The life-cycle of a transaction in other blockchain platforms is usually order-execute, which means that transactions are added to the ledger in a specific order and executed sequentially. But in Hyperledger Fabric, it is a three-step process: execute–order–validate. First, transactions are executed in parallel considering any order. Second, they are ordered by an ordering service. Third, each peer validates and applies the transactions in sequence. The roles of peers also have a strong relationship with robust privacy and permission support, the reader is referred to [43] for more information.

The crowdsourcees shown in Fig. 4 are the end-users in the distribution network and can perform energy trading. Thousands of crowdsourcees are allowed to connect and sign up to the Fabric network via a browser after receiving a code from the operator. The operator also can log in via browser to manage the overall system—screen shots are given in the next section showing the graphical user interface. After enrolling in the network via `Fabric-CA` [44], a certificate needed for enrollment through a software development kit (SDK), crowdsourcees can communicate with the network through `fabric-sdk-node` [45], update their preferences to blockchain and store it in World State [46] which is the database. Peers in Hyperledger are used to commit transactions, maintain the world state, and a copy of the ledger (consists of blocks). The `chaincode` in Hyperledger Fabric is deployed into peers and is executed as a user satisfies

Algorithm 1 Blockchain-Assisted CES Operation

Phase I:
Obtain crowdsourcees preferences $\mathcal{X}_t$
Request/obtain day-ahead P2P ETT requests via blockchain implementation developed (Fig. 4)
Estimate day-ahead forecasts $\boldsymbol{y}_t^{f-24hr}$
Solve CES-OPF (7) and obtain generator and DER schedules
Establish Type A ETTs smart contracts for users $i \in \mathcal{G} \bigcup \mathcal{CT}_1$
Establish Type B ETTs smart contracts for users $i \in \mathcal{CT}_2$
Phase II:
while $t \in 1, \ldots, 24$ hrs **do**
  Select Type 2 crowdsourcees willing to sell solar power to the utility at time t according to the preferences $\mathcal{X}_{2_t}$
  Obtain hour-ahead forecasts $\boldsymbol{y}_t^{f-1hr}$
  Solve CES-ID (9) at time t
  Communicate to crowdsourcees $i \in \mathcal{CT}_2$ incentives $\lambda_{i,t}^{\text{eq}} + \lambda_{i,t}^{a}$
  Establish Type A ETTs smart contracts for users $i \in \mathcal{CT}_2$
end while
Reconcile payments weekly or monthly

their commitments. Then, *ordering service*, akin to mining in Bitcoin, generates new blocks in Fabric. Every peer updates their local blockchain after receiving ordered state updates in the form of blocks from the ordering service. In this way, the order and number of blocks, a form of blockchain, are maintained and synchronized for all peers. The ETTs records are included in blockchain stored at each peer's repository and protected by this mechanism.

This specific implementation is endowed with the following characteristics: 1) scalable to million of crowdsourcees; 2) requires little understanding of the blockchain technology from the users' side; 3) communicates seamlessly with any optimization-based formulation; and 4) requires very little energy to run blockchain. Algorithm 1 illustrates how the developed optimization routines are implemented with blockchain and smart contracts.

VI. Case Studies

A. Simulation Setup

The numerical tests are simulated in Ubuntu 16.04.4 LTS with an Intel Xeon CPU E5-1620 v3 @ 3.50 GHz. We use the

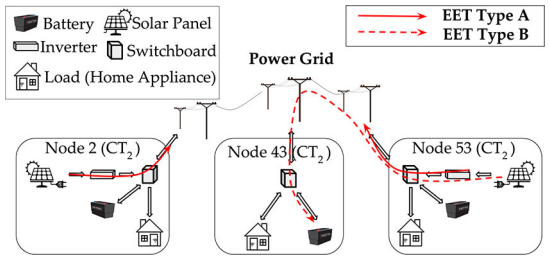


Fig. 5. Scenarios of ETT.

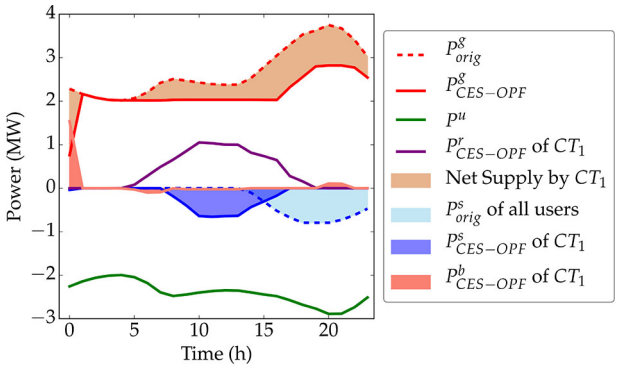


Fig. 6. Aggregate load profile and generation after solving CES-OPF (7).

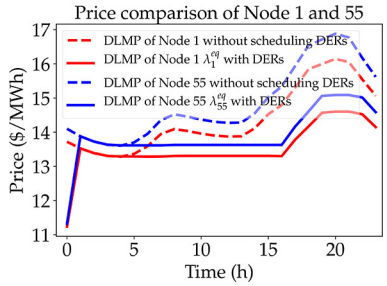


Fig. 7. Price comparison of Nodes 1 and 55 before and after CES-OPF.

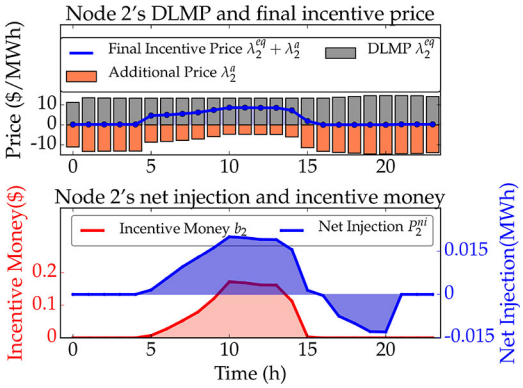


Fig. 8. Final incentive price, net injection, and incentive money for Node 2.

Southern California Edison (SCE) 56-bus test feeder [47] as a distribution network. Reasonable uncontrollable load profile $\boldsymbol{P}^u$ is generated for $T = 24$ h from California Independent System Operator (CAISO) [48] and normalized to ensure that the optimization problems have feasible sets for different time periods. We modify SCE 56-bus test feeder as shown in Fig. 2 and place stationary batteries, solar panels, and uncontrollable and shapeable loads at each bus in the network (see Fig. 5). Similar to [11], batteries are set up with a power capacity of 80% of the peak uncontrollable load at the bus, a 4-h energy storage capacity with 20% initial energy storage. We assume that the solar generation power profile is given and contributes to 50% of the uncontrollable load at peak for each bus. Shapeable loads have net energy demand that is up to 20% the peak power consumption of the uncontrollable loads and can be charged for 4–8 h. The scheduling time of shapeable loads is from 8 A.M. to 11 P.M.

We also assume that each bus is connected to a crowdsourcee of Type 1 ($\mathcal{CT}_1$) or Type 2 ($\mathcal{CT}_2$). We make the following assignment: if the number of a bus is a prime number, then the user belongs to $\mathcal{CT}_2$, otherwise they belong to $\mathcal{CT}_1$ (we have $|\mathcal{CT}_1| = 40$ and $|\mathcal{CT}_2| = 16$). From the above setup, Nodes 2, 43, and 53 belong to $\mathcal{CT}_2$ in Fig. 5. As we present in Table IV, two types of ETT take place in CESs. Type A transactions occur between $\mathcal{CT}_1$ or $\mathcal{CT}_2$ with utilities, while the trading transactions among $\mathcal{CT}_2$ users are Type B transactions. In Fig. 5, we present two scenarios of ETT for further explanation: 1) ETT Type A where Node 2 decides to sell excess solar energy to the utility and 2) ETT Type B where Node 43 chooses to buy energy from Node 53. The next section presents the outcome of the two-phase optimization discussed in Section IV.

B. Results and Discussions

In order to present the effectiveness of our algorithm, we compare the cases with and without considering the energy trading among crowdsourcees based on the modified SCE 56 bus test feeder illustrated in Section VI-A.

1) Phase I (Day-Ahead CES Operation): Solving CES-OPF (7). Fig. 6 shows P^u, P^s_{orig}, and P^g_{orig} (the aggregate uncontrollable load, shapeable load, and the output of generator) when our algorithm is not applied—in the absence of energy crowdsourcing or trading between crowdsourcing. Fig. 6 also shows the aggregate load profile and generation after solving the CES-OPF for $T = 24$ h. The figure shows that battery variable $P^b_{\text{CES-OPF}}$ charges when the solar panel produces and injects power $P^r_{\text{CES-OPF}}$ into network. The reason why the curve of $P^b_{\text{CES-OPF}}$ does not change significantly is that the solar panels do not generate enough energy in this setup. Hence, the algorithm is less inclined to store energy into batteries. As for the scenarios when the solar panel produces enough energy, please refer to Fig. 11 in Section VI-B3. Fig. 6 indicates that shapeable loads of $\mathcal{CT}_1$ are rescheduled to $P^s_{\text{CES-OPF}}$. The updated power generation $P^g_{\text{CES-OPF}}$ is smaller than P^g_{orig} due to the injections of solar power, scheduling of batteries, and shapeable loads from crowdsourcees $\mathcal{CT}_1$. Fig. 7 presents the changes in the DMLPs with and without scheduling DERs in the distribution network through CES-OPF (7) for Nodes 1 and 55. The DLMPs for both nodes are smaller due to the net injection from Type 1 crowdsourcees (shaded orange area in Fig. 6). This illustrates how the DLMP price becomes lower when rescheduling DERs and injecting renewable energy into the grid.

2) Phase II (Real-Time CES Incentives Design): CES-ID is solved once every hour but it can also be solved

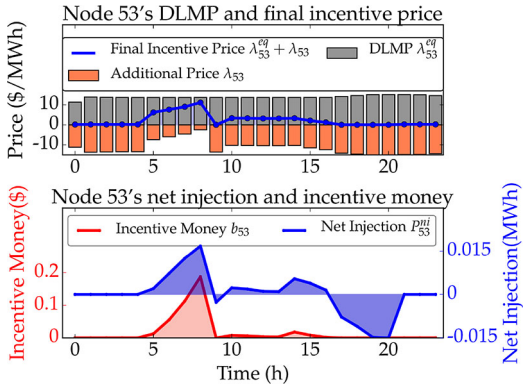


Fig. 9. Final incentive price, net injection, and incentive money for Node 53.

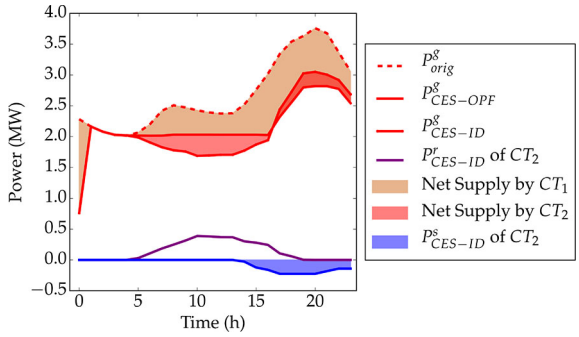


Fig. 10. Aggregate load profiles and generation after Algorithm 1 terminates and the incentives are designed.

TABLE VI
TRANSACTION DETAILS FOR NODE 53

Time	*Seller*	*Buyer*	*Energy*	*ETT Type*	*Phase*
6 am–9 am	Node 53	Utility	0.0385 MWh	Type A	Phase II
9 am–2 pm	Node 53	Node 43	0.119 MWh	Type B	Phase I
14 pm–5 pm	Node 53	Utility	0.062 MWh	Type A	Phase II

every 5–15 min depending on the availability of accurate weather/load forecasts. The monetary rewards offered to Type 2 crowdsourcees are obtained from CES-ID. We assume that the crowdsourcees of Type 2 at Nodes 2, 43, and 53 accept the designed incentives.

Fig. 8 shows the final incentive price, net injection, and overall incentive money for Node 2. The time-varying nature of the final incentive price of a node is due to variations of its DLMP and its net injection. We assume that the solar panel produces energy between 6 A.M. and 7 P.M. The solar panel of Node 2 produces solar power and incentives are earned by the customer between 6 A.M. and 2 P.M. as shown in Fig. 8. However, the load at Node 2 starts to consume energy at 5 P.M. making the net injection of Node 2 is 0 MWh. Hence, no monetary incentives are offered from 7 P.M. to 11 P.M. Fig. 9 presents the results for Type B transactions for $\mathcal{CT}_2$ user at Node 53. The user at Node 43 decides to charge the battery at a constant charging rate between 9 A.M. and 2 P.M., and the excess solar energy produced from Node 53's solar power can satisfy this demand shortage. Notice that Node 43 only consumes energy while Node 53 earns incentive rewards from the utility and negotiated money from Node 43 during different time periods. The transaction details between these crowdsourcees are summarized in Table VI.

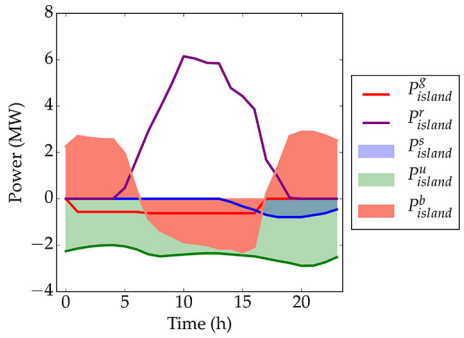


Fig. 11. Results for an islanded, autonomous microgrid operation.

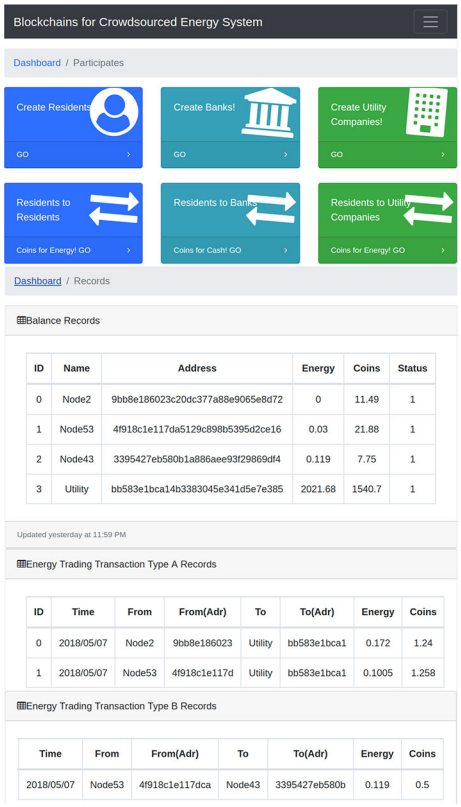


Fig. 12. Web-based user interface for CESs with Hyperledger Fabric.

Fig. 10 depicts the aggregate load profile and generation after Algorithm 1 terminates. More renewable energy is injected into the grid and traded via the designed incentives for $\mathcal{CT}_2$ crowdsourcees. The net contribution of $\mathcal{CT}_2$ crowdsourcees is shaded in red. It is noteworthy to mention that the utility cannot schedule the shapeable loads of $\mathcal{CT}_2$ crowdsourcees. The blue area in Fig. 10 displays the unexpected load demand of $\mathcal{CT}_2$ crowdsourcees. The generator at the substation covers this demand shortage (see Fig. 10), where $P_{\text{CES-ID}}^{g}$ is greater than $P_{\text{CES-OPF}}^{g}$ from 3 P.M. to 11 P.M.

3) Islanded Microgrid Test: After implementing P2P energy trading, we simulate a scenario of a small islanded, autonomous microgrid. In this microgrid, we assume the following. First, all users have: 1) enough solar power to produce enough energy to supply the grid and 2) the microgrid has a battery with sufficient capacity to store excess solar energy. Second, each user agrees to participate in the program and

their DERs would be fully controlled by the microgrid management algorithm akin to Algorithm 1. The simulation setup remains the same as in Section VI-A except the solar panels produce more energy and the capacities of batteries are enlarged. Fig. 11 shows the outcome of the autonomous microgrid operation. Between 6 A.M. and 7 P.M., the solar panel on each crowdsourcees' roof not only produces enough energy to meet the real-time load demand but also stores excess energy into batteries for night use. At night, batteries start to discharge energy to cover the demand shortage facilitating ETT with crowdsourcees in need for energy using blockchain and smart contracts.

4) Blockchain and ETT GUI: Fig. 12 shows a Web-based user prototype that we implemented using Hyperledger Fabric as described in Section V. The Web application shows the system operation which includes creating crowdsourcees, selling energy to the utility or neighborhood, and listing all ETT with information about the prices and the users. This Web-based prototype interacts with the optimization solvers and algorithms that generate forecasts, as well as the crowdsourcees.

VII. Paper Summary, Limitations and Future Work

This paper introduced the notion of blockchain-assisted CESs with a specific implementation and prototype of blockchain that scales to include millions of crowdsourcees and P2P ETTs. A thorough review of the blockchain technology for energy systems is given. Various types of crowdsourcees and ETTs are introduced to mimic current and projected energy market setups. Then, an operational OPF-based model of CESs with batteries, shapeable loads, and other DERs is introduced for distribution networks—considering ETT and crowdsourcees preferences—yielding a day-ahead market equilibrium. Monetary incentives are designed to attract crowdsourcees in hour-ahead and real-time markets to the computed equilibrium while satisfying a demand shortage or surplus. Furthermore, an implementation of blockchain through the IBM Hyperledger Fabric is discussed with its coupling with the optimization models. This implementation allows the system operator to manage the network users to seamlessly trade energy. Finally, case studies are given to illustrate the practicality of the presented methods for classical distribution networks, as well as self-sufficient and islanded microgrids.

There is still an uncontrollable risk in blockchain-based energy trading system, i.e., the attack from malicious market operator, stakeholders, or outsider.

1) A malicious market operator will attempt to modify the operation of the market algorithms in order to produce results that provide an output (market price, load demand) providing them with a financial advantage over the authentic price or demand outputs.
2) A malicious stakeholder might try to produce a false clearing price offering them with reduced energy costs. However, because Hyperledger messages are digitally signed, the consensus results will not be manipulated as long as there are $2f + 1$ total operators, where f is the number of malicious operators.
3) A malicious outsider will try to remotely tamper with all messages communicated between the crowdsourcees and market operators. Their goal is to manipulate the resulting market operations in order to either manipulate: a) the load demand bids submitted by the crowdsourcees or b) the market clearing prices.

In future work, we plan to extend the presented research to address distributed consensus mechanisms for blockchain in CES, and threats from malicious crowdsourcees, market operators, and outsiders.

References

[1] S. Wang, A. Taha, and J. Wang, "Blockchain-assisted crowdsourced energy systems," in *Proc. IEEE PES Gen. Meeting*, Portland, OR, USA, Aug. 2018, pp. 1–5. [Online]. Available: https://arxiv.org/abs/1802.03099

[2] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.

[3] P. G. Ipeirotis, "Analyzing the Amazon mechanical turk marketplace," *XRDS Crossroads ACM Mag. Students*, vol. 17, no. 2, pp. 16–21, 2010.

[4] X. He, D. W. Ho, T. Huang, J. Yu, H. Abu-Rub, and C. Li, "Second-order continuous-time algorithms for economic power dispatch in smart grids," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1482–1492, Sep. 2018.

[5] X. He, T. Huang, J. Yu, C. Li, and Y. Zhang, "A continuous-time algorithm for distributed optimization based on multiagent networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[6] M. H. Hajiesmaili, M. Chen, E. Mallada, and C.-K. Chau, "Crowdsourced storage-assisted demand response in microgrids," in *Proc. ACM 8th Int. Conf. Future Energy Syst.*, 2017, pp. 91–100.

[7] R. Deng, Z. Yang, M. Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 570–582, Jun. 2015.

[8] C. Langbort, "On real-time pricing for strategic agents," in *Proc. Workshop Multi Agent Coordination Estimation*, 2010, pp. 1–7.

[9] T. Namerikawa, N. Okubo, R. Sato, Y. Okawa, and M. Ono, "Real-time pricing mechanism for electricity market with built-in incentive for participation," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2714–2724, Nov. 2015.

[10] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[11] E. Münsing, J. Mather, and S. Moura, "Blockchains for decentralized optimization of energy resources in microgrid networks," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, 2017, pp. 2164–2171.

[12] F. Luo, Z. Y. Dong, G. Liang, J. Murata, and Z. Xu, "A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain," *IEEE Trans. Power Syst.*, to be published.

[13] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things J.*, to be published.

[14] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.

[15] A. Paudel, K. Chaudhari, C. Long, and H. B. Gooi, "Peer-to-peer energy trading in a prosumer based community microgrid: A game-theoretic model," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6087–6097, Aug. 2019.

[16] C. Long, J. Wu, Y. Zhou, and N. Jenkins, "Peer-to-peer energy sharing through a two-stage aggregated battery control in a community microgrid," *Appl. Energy*, vol. 226, pp. 261–276, 2018.

[17] X. Huang, Y. Zhang, D. Li, and L. Han, "An optimal scheduling algorithm for hybrid EV charging scenario using consortium blockchains," *Future Gener. Comput. Syst.*, vol. 91, pp. 555–562, Feb. 2019.

[18] A. Hahn, R. Singh, C. Liu, and S. Chen, "Smart contract-based campus demonstration of decentralized transactive energy auctions," in *Proc. IEEE PES Innov. Smart Grid Technol.*, 2017, pp. 1–5.

[19] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[20] A. Baliga, "Understanding blockchain consensus models," Persistent Syst. Ltd., Pune, India, Rep., 2017.

[21] *Digiconomist*. Accessed: Feb. 7, 2019. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption

[22] V. Zamfir. (2015). *Introducing Casper 'The Friendly Ghost'*. Accessed: Aug. 2015. [Online]. Available: https://blog.ethereum.org

[23] *Hyperledger—Open Source Blockchain Technologies*. Accessed: Feb. 7, 2019. [Online]. Available: https://www.hyperledger.org

[24] J. Licata. (Mar. 2017). *500 Billion Reasons Blockchain Can Be a Power Play*. Accessed: Feb. 7, 2019. [Online]. Available: https://www.greenbiz.com/article/500-billion-reasons-blockchain-can-be-power-play

[25] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: A case study: The Brooklyn microgrid," *Appl. Energy*, vol. 210, pp. 870–880, Jan. 2018.

[26] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, "Reinforcement learning in energy trading game among smart microgrids," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5109–5119, Aug. 2016.

[27] S. Park, J. Lee, S. Bae, G. Hwang, and J. K. Choi, "Contribution-based energy-trading mechanism in microgrids for future smart grid: A game theoretic approach," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4255–4265, Jul. 2016.

[28] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informat.*, to be published.

[29] L. Zhu, Y. Wu, K. Gai, and K.-K. R. Choo, "Controllable and trustworthy blockchain-based cloud data management," *Future Gener. Comput. Syst.*, vol. 91, pp. 527–535, Feb. 2019.

[30] *DAISEE*. Accessed: Feb. 8, 2019. [Online]. Available: https://daisee.org

[31] *Grid+*. Accessed: Feb. 8, 2019. [Online]. Available: https://gridplus.io/

[32] *Bankymoon*. Accessed: Feb. 7, 2019. [Online]. Available: http://bankymoon.co.za/

[33] P. Terium. *The Global Energy System Is Changing, and Customers Must Take Center Stage*. Accessed: Feb. 17, 2019. [Online]. Available: https://www.weforum.org/agenda/2017/03/how-customers-are-taking-centre-stage-to-transform-the-energy-system/

[34] *Brooklynmicrogrid.com*. Accessed: Feb. 8, 2019. [Online]. Available: http://brooklynmicrogrid.com/

[35] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification—Part I," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2554–2564, Aug. 2013.

[36] S. H. Low, "Convex relaxation of optimal power flow—Part II: Exactness," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 177–189, Jun. 2014.

[37] M. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Del.*, vol. PWRD-4, no. 1, pp. 735–743, Jan. 1989.

[38] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, nos. 1–3, pp. 193–228, 1998.

[39] Q. Peng and S. H. Low, "Distributed algorithm for optimal power flow on a radial network," in *Proc. IEEE 53rd Annu. Conf. Decis. Control (CDC)*, 2014, pp. 167–172.

[40] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov, "Options for control of reactive power by distributed photovoltaic generators," *Proc. IEEE*, vol. 99, no. 6, pp. 1063–1073, Jun. 2011.

[41] J. Cottrell and M. Basden, *How Utilities Are Using Blockchain to Modernize the Grid*, Harvard Bus. Rev., Boston, MA, USA, Mar. 2017.

[42] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[43] *Peers—Hyperledger-Fabricdocs Master Documentation*. Accessed: Feb. 8, 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/peers/peers.html

[44] *Hyperledger/Fabric-CA*. Accessed: Feb. 7, 2019. [Online]. Available: https://github.com/hyperledger/fabric-ca

[45] *Hyperledger/Fabric-SDK-Node*. Accessed: Feb. 7, 2019. [Online]. Available: https://github.com/hyperledger/fabric-sdk-node

[46] *Ledger—Hyperledger-Fabricdocs Master Documentation*. Accessed: Feb. 7, 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/ledger/ledger.html

[47] Q. Peng, Y. Tang, and S. H. Low, "Feeder reconfiguration in distribution networks based on convex relaxation of OPF," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1793–1804, 2014.

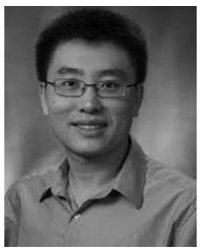
[48] CAISO. *Real Time Demand Curves*. Accessed: Jul. 2017. [Online]. Available: http://www.caiso.com/outlook/SystemStatus.html

Shen Wang (S'18) received the master's degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Texas at San Antonio, San Antonio, TX, USA.

His current research interests include optimal control in cyber-physical systems with special focus on energy and water systems.

Ahmad F. Taha (S'07–M'15) received the B.E. degree in electrical and computer engineering from the American University of Beirut, Beirut, Lebanon, in 2011, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2015.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX, USA. He is interested in understanding how complex cyber-physical systems (CPS) operate, behave, and *misbehave*. His current research interest includes optimization, control, and security of CPSs with applications to power, water, and transportation networks.



Jianhui Wang (M'07–SM'12) received the Ph.D. degree in electrical engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2007.

He had a 11-year stint with Argonne National Laboratory, Lemont, IL, USA, with the last appointment as a Section Lead Advanced Grid Modeling. He is currently an Associate Professor with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX, USA. He has held visiting positions in Europe, Australia, and Hong Kong, including a VELUX Visiting Professorship with the Technical University of Denmark, Lyngby, Denmark.

Dr. Wang was a recipient of the Highly Cited Researcher Award from Clarivate Analytics for 2018. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON SMART GRID and an IEEE Power & Energy Society (PES) Distinguished Lecturer. He is the Secretary of the IEEE PES Power System Operations, Planning & Economics Committee.

Karla Kvaternik received the Ph.D. degree in systems and control theory from the University of Toronto, Toronto, ON, Canada, in 2015, focusing on the synthesis of decentralized optimization algorithms, networked multiagent coordination control methods, and stability analysis techniques.

She is a Research Scientist with the Predictive Analytics Research Group, Siemens Corporate Technology, Princeton, NJ, USA, where she addresses a variety of data-driven decision support problems and prototyping transactive energy applications. She was a Research Associate with Princeton University, Princeton, NJ, USA, where she studied multiagent reinforcement learning algorithms applied to multiarmed bandit settings.

Dr. Kvaternik was a recipient of the Best Student Paper Award at the IEEE Multiconference on Systems and Control, for her M.Sc. research on multivariable output feedback for nonlinear systems.

Adam Hahn received the M.S. and Ph.D. degrees in computer engineering from the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA, in 2006 and 2013, respectively.

He was a Senior Information Security Engineer with the MITRE Corporation, McLean, VA, USA, supporting numerous cybersecurity assessments within the federal government and leading research projects in cyber-physical system (CPS) security. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA. His current research interests include cybersecurity of the smart grid and CPSs, including intrusion detection, risk modeling, vulnerability assessment, and secure system architectures.