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, Member, IEEE, 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 (DER)—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 (CES), with peer-to-peer (P2P) energy trading transactions. An operational model of CESs in distribution networks is presented considering various types of energy trading transactions and crowdsources. 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 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—Energy Crowdsourcing, Blockchain, Energy Trading, Peer-to-Peer 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.

Energy crowdsourcing offers the possibility of this transformation in energy systems. The central theme in crowdsourcing is the utilization of the crowd’s power to achieve system-level objectives. Crowdsourcing has been utilized in various disciplines such as medicine, engineering system design, and cyber-physical systems. This paper puts forth operational models for Crowdsourced Energy System (CES), shown in Fig. 1, for collaborative production and consumption in energy markets. 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.

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 (prosumer-prosumer or prosumer-operator trades). This paper develops a blockchain-assisted architecture with an optimization model to manage the day-ahead and hour-ahead operation of CESs, with peer-to-peer (P2P) energy trading. Related literature and the paper’s contributions are given next.

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 the majority of
these studies is on unit commitment and economic dispatch problems as well as scheduling of DERs, given uncertainty from renewables and load forecasts. The work in [2] investigates security-constrained unit commitment with volatile wind power. The authors in [3] present computationally tractable optimization routines to manage the operation of distribution networks with high photovoltaic penetration. Other studies explore augmenting classical optimal power flow (OPF) routines with other types of DERs in large-scale networks and microgrids using novel optimization methods [4]–[7]. An exhaustive list of this work is not provided here, but the majority of this research aims to maintain the grid’s frequency and voltage within acceptable ranges, while managing DERs and grid operations.

Another branch of related work studies the design of demand response signals and incentives to drive DER owners to contribute to energy production. The authors in [8] investigate the problem of utilizing heterogeneous, crowdsourced energy storage systems in microgrids to perform demand response. In summary, there are three approaches to demand response: (a) Reducing the total demand by using their local DERs as the grid requires. (b) Reducing demand through shifting controllable loads. (c) Designing efficient generator setpoints to reduce the total generation [9]. The majority of demand response schedules focus on the 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 [10]–[12]. In short, the majority of the above studies reconstruct the price of regulation from the dual variables in a static or dynamic optimization problem. Background on blockchain is given next.

B. Blockchain and Energy 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 [13]. The main motivation behind blockchain is the need to have a distributed, secure system that minimizes the need for trusted third parties or central authorities to organize and authorize transactions. This innovation is increasingly important for a transactive system where miners validate new transactions and record them on the global blockchain ledger. The annual estimated electricity consumption of Bitcoin PoW consensus is 73.12 Terawatt-hour—a staggering 0.33% of world electricity consumption [16]. Furthermore, PoW techniques typically have limitations on the number of transactions per second, which limits use in high performance environments. Other consensus mechanisms, such as Proof of Stake (e.g., Ethereum Casper [17]) which requires betting a cryptocurrency-based deposit on block validation correctness or Redundant Byzantine Fault Tolerance (RBFT) (e.g., using the IBM Hyperledger Fabric [18], [19]), can be used to reduce energy consumption.

- **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 [15]; various consensus protocols have been developed for blockchain implementations. An important component of blockchain is the mining process where miners validate new transactions and record them on the global blockchain ledger. The majority of this research aims to maintain the grid’s frequency and voltage within acceptable ranges, while managing DERs and grid operations.

- **Smart contracts:** Smart contracts provide protocols andTouring complete virtual machines that enable nodes to execute some program based on the results of new transactions and enable the blockchain to support a wide range of functions. Smart contracts and blockchain provide an excellent platform to perform energy trading transactions. In particular, the authors in [20], [21] provide a high-level description to the main merits of using cryptocurrencies and blockchain in energy systems.

- **Permissioned and privacy mechanisms:** Blockchains 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. Furthermore, public chains typically do not enforce confidentiality of the ledger, which presents challenges in applications where transactions involve personal or proprietary data.

Tab. I summarizes the attributes of different implementations of current blockchains, which each have some applicability to future energy systems, the Hyperledger platform is used to implement the proposed CES scheme. Section V provides additional discussion on why this platform was selected.

The authors in [22] design smart contracts to enable energy producers to sell excess energy to the highest bidder through honest bidding. Through the Ethereum blockchain, this design is tested at Washington State University campus. The study in [23] investigates a paradigm for providing demand-side

| Year Released | Bitcoin | Ethereum | Hyperledger Fabric |
|---------------|---------|---------|-------------------|
| 2009          |         |         |                   |
| 2015          |         |         |                   |
| 2017          |         |         |                   |
| Cryptocurrency | Bitcoin | Ether   | None              |
| Network       | public  | public  | permissioned      |
| Transactions  | anonymous| anonymous| public/confidential|
| Consensus     | PoW     | PoW     |                   |
| Smart Contracts| None    | Solidity| RBFT              |
| Language      | C++     | C++/Golang| Chaincode        |

| Table I | VARIOUS IMPLEMENTATIONS OF BLOCKCHAIN. POW AND RBFT STAND FOR PROOF OF WORK AND REDUNDANT BYZANTINE FAULT TOLERANCE. |
management services where households can manage their surplus/shortage of energy through self-enforcing smart contracts via blockchain. The authors in [24] develop a novel blockchain-based architecture which includes a decentralized OPF formulation. Recently, a blockchain-based transactive energy systems implementation that preserves the privacy of users is presented in [25]. In order to manage billions of general IoT devices, the author in [26] presents an architecture to scalable access management in IoT via blockchain. Unlike solutions that target millions/billions of IoT devices, power grids and distribution systems are interconnected through the physical components. The need to have a blockchain-solution that considers the grid’s physical constraints, the distribution network topology and parameters, user’s preferences, and energy trading transactions—while managing the grid—imposes its own set of challenges which are addressed in this paper through CES operational model and the scalable blockchain implementation and prototype.

Beyond research-oriented studies, companies are using blockchain to provide services in actual energy systems. Daisee [27], a French startup bootstrapped in 2016, proposes Internets of Energy and focuses on P2P energy trading. Grid+ [28] that is a follow up of Brooklyn Microgrid project [29], offers wholesale price for users using Ethereum blockchain and are starting to test their system in Texas. Bankymoon [30] develops Bitcoin-funded smart meters and allows customers to top up smart meters using Bitcoin.

C. Paper Contributions and Organization

The majority of the aforementioned blockchain implementations and OPF-based operation models have three limitations. First, the blockchain architectures are not scalable to include millions of energy trading transactions—especially that blockchain-based trades consume a significant amount of energy. Second, it is unclear how energy trading between prosumers can take place within the operational models. Third, the computed OPF setpoints for DERs and controllable loads might not be eventually adopted by crowdsourceses and prosumers. The paper addresses these literature gaps and the contributions of this work are given as follows.

- An operational framework and model of CESs in distribution networks is presented considering various types of energy trading transactions and crowdsources. The presented framework enables P2P energy trading at the distribution level, where ubiquitous distribution-level asset owners can trade with each other. This has not been 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).

- A two-phase, near real-time operation algorithm for CESs is explored. The first phase focuses on the day-ahead scheduling of generation and controllable DERs, whereas the second phase is developed for hour-ahead or even real-time operation of distribution networks. While the first phase manages the bulk of grid-operation, the second phase is used to balance hour-ahead and real-time deficit/surplus in energy via monetary crowdsourcing 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 crowdsourceses to contribute to this ecosystem. The algorithm supports the operation of islanded, self-autonomous microgrid (Section IV).

- 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 and prototype illustration are provided (Section VI).

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 crowdsources and energy trading transactions 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 $\mathcal{N} = \mathcal{G} \cup \mathcal{C} \cup \mathcal{L}$, where $\mathcal{G} = \{1, \ldots, n_\text{g}\}$ collects the $n_\text{g}$ utility-scale power generation connected to the feeder/substation; $\mathcal{C} = \{1, \ldots, n_\text{c}\}$ collects the buses containing $n_\text{c}$ users who signed up for crowdsourcing schedules; $\mathcal{L} = \{1, \ldots, n_\text{l}\}$ collects load buses.

The crowdsourcer here is the utility company or any other system operator. We distinguish between two types of crowdsources in $\mathcal{C}$. Type 1 crowdsources commit in the day-ahead markets (and perhaps monthly or yearly) to the crowdsourcing tasks requested by the operator. Type 1 crowdsources 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 crowdsources provide near real-time adjustments or decisions based on real-time notifications and decisions from the operator. For example, the operator informs Type 2 crowdsources 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 crowdsources provide operators with day-ahead planning flexibility, in contrast with Type 2 crowdsources who operate on a faster timescale. The distinction between these two types of users is needed as it resembles projected market setups [31]. We define these two types as CT$_1$ and CT$_2$, with $C = CT_1 \cup CT_2$; this is depicted in Fig. 2.

We consider two types of energy trading transactions (ETT). Type A: This is akin to what takes place in today’s grids, where Type 1 or 2 crowdsources feed the grid with power. This type of ETT is solely between crowdsources and the network operator. Type B: Crowdsources can trade energy with each other where the seller injects power into the grid. Fig. 3 shows the types of crowdsources and ETTS. Since energy production and demand response from Type 1 crowdsources are controlled by the operator, Type B ETTS only occur among Type 2 crowdsources. However, Type A ETTS can also take place between Type 2 crowdsources and the utility. The participants and the transaction types are showed in detail in Fig. 3. The Brooklyn Microgrid [29] project is an example of Type B ETT for Type 2 crowdsources.

A. Operational Model of Generators, Loads and DERs

Let $i \in N$ denote the bus index of the distribution system and $t$ denote the time-period. In this paper, 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 $S^q_{i,t} = P^q_{i,t} + jQ^q_{i,t}$ for $i \in G$ at $t$ are considered to have quadratic cost functions as $C_{i,t}(P^q_{i,t}) = \alpha_{i,t}(P^q_{i,t})^2 + \beta_{i,t}P^q_{i,t} + \gamma_{i,t}$ where $\alpha_{i,t}$, $\beta_{i,t}$, and $\gamma_{i,t}$ are given parameters for the cost function of the $i$-th generator at $t$.

2) Solar Energy Generation: Solar panels generate real power $P^s_{i,t}$ for bus $i \in C$ at $t$. Note that CT$_1$ crowdsources do not control whether $P^s_{i,t}$ is fed into the grid or not (it is controlled by the utility/operator), whereas CT$_2$ crowdsources dictate whether to use $P^s_{i,t}$ 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 $i \in C$. Negative $P^b_{i,t}$ implies that power is withdrawn. The battery operational model [32] is described as:

$$E^b_{i,t} = E^b_{i,t-1} + H^b_{i,t}\eta_{i,in} - \frac{D^b_{i,t}}{\eta_{i,out}} \quad (1a)$$
$$P^b_{i,t} = D^b_{i,t} - H^b_{i,t} \quad (1b)$$
$$0 \leq D^b_{i,t} \leq P^b_{i,t,dis} \quad (1c)$$
$$0 \leq H^b_{i,t} \leq P^b_{i,t,cha} \quad (1d)$$
$$E^b_{i,min} \leq E^b_{i,t} \leq E^b_{i,max}. \quad (1e)$$

In the above battery model, we consider a unit time-period; $\eta_{i,in}$ and $\eta_{i,out}$ 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 $E^b_{i,min}$ and $E^b_{i,max}$ 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. $P^b_{i,t,dis}$ stands for the limitation of discharging power, $P^b_{i,t,cha}$ has a similar meaning for charging power. All of variables related to batteries model are included in a single vector variable $x^b_{i,t} := (E^b_{i,t}, H^b_{i,t}, P^b_{i,t,dis}, P^b_{i,t,cha})$.

4) Uncontrollable Loads: Uncontrollable loads (lights, plug loads, street lights, et cetera) are considered to be given and are denoted by $S^u_{i,t}$ for all $i \in L$ (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 $i \in L$, such as plug-in electric vehicles and loads from appliances with flexible power profile but fixed energy demand $E^s_{i,demand}$ in 24 hours. These shapeable loads must be satisfied between $t_{start}$ and $t_{end}$. The model describing the shapeable loads [32] is given next.

$$E^s_{i,demand} = \sum_{t=1}^{T} S^s_{i,t}\Delta t \quad (2a)$$
$$S^s_{i,t} = 0, \text{ for } t = 1, \ldots, t_{start}, t_{end}, \ldots, T \quad (2b)$$
$$S^s_{i,min} \leq S^s_{i,t} \leq S^s_{i,max} \quad (2c)$$

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

B. Distribution Network Model

For each bus $i \in N$, 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

$$p_{i,t} = P^s_{i,t} + P^b_{i,t} + P^r_{i,t} - P^a_{i,t} - P^s_{i,t}. \quad (3)$$

Similarly for the net reactive power injection. For each line $i \in E$, 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 $|I_i|^2$ 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 $x_i := (x^b, x^s)_i$ as a variable vector collecting the variables related to batteries and shapeable loads. Since two types of crowdsources are defined, $x_i$ is divided into two variables $x_{i1}$ and $x_{i2}$, which stands for the variables belong to Type 1 and Type 2 crowdsources and hence $x_i = (x_{i1}, x_{i2})$. Let $y_i :=
(\(P_{i,t}^b, P_{i,t}^r\)) 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 [33]. This model eliminates the phase angles of \(V_i\) and \(I_i\) and uses only \((v_i, l_i, s_i, S_i)\).

\[
v_{A_i} = v_i - 2(r_i P_i + x_i Q_i) + \ell_i (r_i^2 + x_i^2) \quad i \in \mathcal{E} \tag{4a}
\]

\[
\sum_{j \in \mathcal{C}_i} (P_j - \ell_j r_j) + p_i = P_i \quad i \in \mathcal{N} \tag{4b}
\]

\[
\sum_{j \in \mathcal{C}_i} (Q_j - \ell_j x_j) + q_i = Q_i \quad i \in \mathcal{E} \tag{4c}
\]

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

Due to (4d), the branch flow model is not convex. However, the model can be convexified using the second order cone program (SOCP) relaxation [34] and rewritten as

\[
\begin{bmatrix}
2 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix}
P_i \\
Q_i \\
v_i \\
l_i
\end{bmatrix}
\leq
\begin{bmatrix}
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
P_i \\
Q_i \\
v_i \\
l_i
\end{bmatrix}
\tag{5}
\]

The nonconvex branch flow model can be cast with SOCP constraints, denoted by \(\text{CvxBranchFlowModel}(z_t)\). It is worthwhile noting that \(\text{CvxBranchFlowModel}(z_t)\) includes equations (4a)–(4c) and (5). In this paper, all of the branch flow equations are included in a single vector variable \(z_t := (v, l, s, S)_t\). Tab. II lists the variables introduced in CES. The next section introduces the CES optimal power flow formulation and incentive design.

IV. CES-OPF AND INCENTIVES DESIGN

In this section, we propose a two-phase algorithm which minimizes 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 CESs. In addition, the presented algorithm supports P2P energy trading transactions (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. This phase takes into account the types of crowdsourcees and their day-ahead preferences as well as the pre-scheduled ETTs between any two crowdsourcees. Given the day-ahead solutions from the first phase, the second phase of the algorithm performs two significant operations. First, rectifying the mismatch in the day-ahead forecasts and hence the demand shortage/surplus by (a) obtaining more accurate, hour-ahead forecasts and (b) 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. Tab. III 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 \(CT_1\) and utility, incentive pricing for \(CT_2\) 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 \(CT_1\) users’ DERs according to the signed contract, while \(CT_2\) users decide to participate or not in the crowdsourcing schedules based on their preferences and the offered incentives. For example, \(CT_2\) users can sell their excess solar power to the utility if the designed incentive is sufficient or acceptable in the hour-ahead or real-time markets. This entails—and due to the nature of \(CT_2\) users—that the output from the solar panels \(P_{i,t}^s\), batteries \(P_{i,t}^b\), and shapeable loads \(P_{i,t}^s\) for users \(i \in CT_2\) are uncontrollable by the utility. Hence, and if Type 2 crowdsourcees declare that they would not trade energy with other users (ETT Type B), then in this phase these quantities are excluded in (3) by setting them to zero yielding

\[
P_{i,t}^s = P_{i,t}^b = P_{i,t}^s = 0, i \in CT_2. \tag{6}
\]

Otherwise, the sellers and buyers should send the energy supply-demand requests for P2P energy trading day ahead

| TABLE II |
| --- |
| **NOTATION FOR VARIOUS DERs IN CES. SYMBOLS WITH OR WITHOUT SUBSCRIPT i, t HAVE THE SAME MEANING FOR SIMPLICITY.** |
| Symbols | Description |
| --- | --- |
| \(S_{i,t}^f\) | Dispatchable generation |
| \(P_{i,t}^b\) | Real power generated from solar panel |
| \(P_{i,t}^r\) | Output power of the battery |
| \(S_{i,t}^r\) | Apparent power of uncontrollable load |
| \(S_{i,t}^r\) | Apparent power of shapeable load |
| \(p_{i,t}\) | Net real power injection at each bus |
| \(#\{i,t\}\) | A variable collecting all of the variables in battery model |
| \(#\{i,t\}\) | A variable collecting all of the variables in shapeable model |
| \(#\{i,t\}\) | A variable collecting variables in battery and shapeable model |
| \(y_t\) | A variable collecting the variables of uncontrollable loads and solar energy |
| \(#\{i,t\}\) | A variable collecting all of the branch flow variables |
| \(X_t\) | Preferences and setting parameters of crowdsourcees |

| TABLE III |
| --- |
| **ETT TYPES AND THE CORRESPONDING IN RELEVANCE TO THE TWO-PHASE ALGORITHM.** |
| ETT Type | Seller | Buyer | Pricing Mechanism | Optimization Phase |
| --- | --- | --- | --- | --- |
| Type A | \(CT_1\) | Utility | Contract pricing | Phase I |
| Type A | \(CT_2\) | Utility | Incentive pricing | Phase II |
| Type B | \(CT_2\) | \(CT_2\) | Negotiated pricing | Phase I |
to the utility. These requests for $CT_2$ users in time-period $t$ are expressed as constraint $\text{EnergyTrading}(x_{2i}, y_{t})$. This constraint ultimately transforms variables $x_{2i}, y_{t}$ to mere pre-defined constants since the users decide to inject (or receive) a certain amount of energy into (from) the grid. The CES Optimal Power Flow (CES-OPF) is formulated as

**CES-OPF**: $\min_{P_t^g, x_{t}, z_t} \sum_{t=1}^{T} J_t(x_{t}, z_t, P_t^g)$

s.t. (1) – (3), (6), $y_{t} = y_{t}^{1-24hr}, x_{t} \in \chi_t$ (7)

$\text{CvxBranchFlowModel}(z_t), z_t^{\min} \leq z_t \leq z_t^{\max}$

$P_t^g \in \mathcal{P}$, $\text{EnergyTrading}(x_{2i}, y_t)$.

The objective function of CES-OPF is defined as

$$J_t(x_{t}, z_t, P_t^g) = \sum_{i=1}^{n} C_{i,t}(P_t^g) + \sum_{i=1}^{|E|} l_{i,t} r_{i} + \sum_{i=1}^{|CT_1|} U_{i}(x_{i}).$$

The objective here in the distribution network operation is to minimize the generator’s cost function, given by $\sum_{i=1}^{n} C_{i,t}(P_t^g)$, in addition to the thermal losses that are characterized by $\sum_{i=1}^{|E|} l_{i,t} r_{i}$, and crowdsources’ disutility function $\sum_{i=1}^{|CT_1|} U_{i}(x_{i})$. The formulation CES-OPF captures the cost of power losses between two peers in the distribution network through the second term of $J_t(\cdot)$ which sums the losses for all lines $E$ in a distribution network. These lines include the distribution lines between any two users/peers, including traditional energy consumers. The convexified branch flow model is used, making CES-OPF (7) a convex quadratic program that can be solved for large-scale networks. LinDistFlow$(\cdot)$, a centralized fashion. For medium- or small-scale distribution networks and microgrids, it is plausible to solve CES-OPF in a centralized fashion after requesting the user’s preferences $\chi_t$ ahead of time—as a simpler alternative to implementing the decentralized ADMM algorithm. Another way of making CES-OPF more computationally tractable is to replace the convexified branch flow model with the LinDistFlow($z_t$) model [36] which is linear in $z_t$; this transforms CES-OPF to a quadratic program that can be solved for large-scale networks.

After solving CES-OPF, we obtain $S_{i,t}^g, x_{2i}^{eq}, P_{t,i}^{g,eq}$ and $x_{1i}^{eq}$ which includes $P_{t,i}^{b,eq}$ and $S_{i,t}^{s,eq}$. This entails that the utility-scale generation, batteries and shapeable loads belonging to $CT_1$ users will be fixed with this equilibrium for the next 24 hours. To compensate crowdsources 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 $\lambda_{t,i}^{eq}$.

**B. Phase II: Real-Time CES Incentives Design**

As outlined in Section IV-A, we solve CES-OPF and obtain the setpoints for utility-scale power plants and Type 1 crowdsources, knowing that some ETTs will take place between crowdsources. In this section, the presented crowdsourcing incentive design performs the two key functions: (a) Incentivizes Type 2 crowdsources to sell excess solar power to the utility; (b) 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 $CT_2$.

For $i \in CT_2$, the amount of energy provided to the grid is depicted by the net injection power $P_{t,i}^{eq}$ and computed as

$$P_{t,i}^{eq} = P_{t,i}^{eq} - P_{t,i}^{s} + P_{t,i}^{b}, \quad i \in CT_2.$$ (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 crowdsources through energy trading. Here, for $i \in CT_2$, shapeable loads and batteries cannot be scheduled 24 hours ahead since no contract exists between Type 2 crowdsources and the utility. Hence, $P_{t,i}^{s}$ and $P_{t,i}^{b}$ which belong to variable $x_{2i}$, are treated now as uncontrollable loads for $CT_2$ in Phase II. In addition, solar energy is also known ahead of time. Hence, $P_{t,i}^{eq}$ is known and not an optimization variable for Type 2 crowdsources from (8). The crowdsourcing incentive design routine for crowdsources $i$ at time $t$ is formulated as

**CES-ID**: $\min_{x_{1i}, x_{2i}, P_{t,i}^g, S_{i,t}^{b,eq}, S_{i,t}^{s,eq}} \sum_{i=1}^{n} C_{i,t}(P_{t,i}^{g,eq} - P_{t,i}^{b,eq}) + \sum_{i=1}^{|E|} l_{i,t} r_{i} + \sum_{i=1}^{|CT_1|} b_{i,t}$

s.t. (1) – (3), (8), $x_{1i} = x_{1i}^{eq}, x_{2i} \in \chi_t$

$y_{t} = y_{t}^{1-1hr}, z_t^{\min} \leq z_t \leq z_t^{\max}$

$\text{CvxBranchFlowModel}(z_t), P_t^{eq} \in \mathcal{P}$

$P_{t,i}^{b} = P_{t,i}^{b,i}(x_{1i}^{eq} + \lambda_{t,i}^{eq}), b_{i,t} \geq 0, i \in CT_2$

$\sum_{i=1}^{|CT_1|} b_{i,t} \geq b_{t}^{\text{total}}$, $i \in CT_2$.

In CES-ID, the objective is to minimize the (a) deviation in the cost of generation from the day-ahead operating point, (b) the network’s thermal losses, and (c) the budget $\sum_{i=1}^{|CT_1|} b_{i,t}$ (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 $P_{t,i}^{b,i}, S_{i,t}^{b,eq}, S_{i,t}^{s,eq} \in x_{2i}$ which is obtained by CES-OPF to schedule
DERs that are controlled by the utility. For \(i \in CT_2\), the willingness to sell energy to the utility is set in preference \(X_{2i}\) which sent to system operator. The constraints on \(y_t, z_t, P_t^q\) are the same as CES-OPF (7) except that \(y_t\) is set to the hour-ahead (or shorter) available forecast \(y_{t-1hr}\).

Besides the optimization variables mentioned above, we consider that Type 2 crowdsources receive the final incentive price \(\lambda_{eq}^i + \lambda_a^i\), where \(\lambda_a^i\), additional variable, is an adjustment price which varies with the net energy injected to grid and location of \(CT_2\); \(\lambda_{eq}^i\) is DLMP computed by CES-OPF. The variable budget \(b_{i,t}\) for \(i \in CT_2\) at \(t\) is equal to \(P_{eq}^i(\lambda_{eq}^i + \lambda_a^i)\), which is always greater than 0. As mentioned above, \(P_{eq}^i\) are known. When the crowdsourcer \(i\) has no energy sell to utility \((P_{eq}^i \leq 0)\), variable \(\lambda_a^i\) is forced to approach \(-\lambda_{eq}^i\) to make \(b_{i,t}\) as close to zero. Hence no incentive is offered to those who inject no power into the grid. When \(P_{eq}^i > 0\) which means crowdsourcer \(i\) at \(t\) has excess energy to sell, variable \(\lambda_a^i\) is forced to be small while also minimizing the final incentive price \(\lambda_{eq}^i + \lambda_a^i\) and budget \(b_{i,t}\) for all Type 2 crowdsources. At time \(t\), the total budget for \(CT_2\) is \(b_{t\text{ total}}\), which can be set as a reasonable value.

The CES-ID is solved hourly, and the computed incentives are sent to users at the end of the day. Thus, the energy trading (ETT Type A) between \(CT_2\) users and the utility is finished. The transactions are done by the assist of blockchain, which is explained in 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 energy trading transactions. 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 Tab. I summarizes the attributes of different blockchain platforms, this section identifies the properties most applicable for the proposed CES model and algorithms introduced in Section 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 Tab. IV.

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 crowdsourceres. 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.

This, unlike other blockchain applications, still requires a central authority—the utility company or the system operator managing the grid. Small-scale energy trading without a central authority can take place (see [37]), yet the scaling of these transactions to include thousands of people and millions of daily energy transactions without the utility interfering is remote in todays markets. To that end, the presented architecture in this paper still requires a central authority to manage the grid.

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 blockchain implementation consists of three modules—surrounded by the dotted lines in Fig. 4. Module I includes the optimization problems in Section IV which are coded by CVXPY [38]. Module II is a Node.js application, the communication between Python-written optimization problem and Node.js is via the child_process standard library to spawn a python process which computes the solutions to CES-OPF (7), CES-ID (9) while returning results back to Node.js program.

Module III is the IBM Hyperledger Fabric Network deployed in cloud to provide the blockchain service. This service can be provided by a central authority (the utility or system operator). 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 lifecycle of transactions which is one key difference between Hyperledger Fabric and many other blockchain platforms. The lifecycle 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

| Participants | CES Requirements | Blockchain Features |
|--------------|------------------|---------------------|
|              | 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 |
| Computational Needs | 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 | Crowdsourcer preferences and usages likely exposes privacy data | Permissioned model that protects crowdsourceee data from external observers |
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 [39] 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 utility. The operator or utility also can log in via browser to manage the overall system—screenshots are given in the next section showing the graphical user interface. After enrolling in the network via Fabric-CA [40], a certificate needed for enrollment through a software development kit (SDK), crowdsourcees can communicate with the network through fabric-sdk-node [41], update their preferences to blockchain and store it in World State [42] 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 their commitments. Then, ordering service, akin to mining in Bitcoin, generates new blocks in Fabric. Each peer updates its 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 in each peer 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(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz. We use the Southern California Edison (SCE) 56-bus test feeder [43] as a distribution network. Reasonable uncontrollable load profile is generated for T = 24 hrs from California Independent System Operator (CAISO) [44] and normalized to ensure that the optimization problems have feasible sets for different time-periods. We place stationary batteries, solar panels, uncontrollable and shapeable loads at each bus in the network; see Fig. 5. Similar to [32], batteries are set up with a power capacity of 80% of the peak uncontrollable load at the bus, a 4-hour 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 hours. The scheduling time of shapeable loads is from 8 am to 11 pm.

We also assume that each bus is connected to a crowdsourcee of Type 1 (\(CT_1\)) or Type 2 (\(CT_2\)). We make the following assignment: If the number of a bus is a prime number, then the user belongs to \(CT_2\), otherwise they belong to \(CT_1\) (we have \(|CT_1| = 40\) and \(|CT_2| = 16\). From the above setup, Nodes 2, 43 and 53 belong to \(CT_2\) in Fig. 5. As we present in Tab. III, two types of energy trading transactions (ETT) take place in CESs. Type A ETTs occur between \(CT_1\) or

![Diagram of combining blockchain and smart contract with the optimization formulations presented in this paper.](image-url)

**Algorithm 1** Blockchain-Assisted CES Operation

**Phase I:**
1. Obtain crowdsourcees preferences \(\mathcal{X}_t\)
2. Request/obtain day-ahead P2P ETT requests via blockchain implementation developed (Fig. 4)
3. Estimate day-ahead forecasts \(y_{i,t-24hr}\)
4. Solve CES-OPF (7) and obtain generator and DER schedules
5. Establish Type A ETTs smart contracts for users \(i \in G \cup CT_1\)
6. Establish Type B ETTs smart contracts for users \(i \in CT_2\)

**Phase II:**

```
while t \in 1, \ldots, 24 \text{ hrs} do
    Select Type 2 crowdsourcees willing to sell solar power to the utility at time t according to the preferences \(\mathcal{X}_t\)
    Obtain hour-ahead forecasts \(y_{i,t-1hr}\)
    Solve CES-ID (9) at time t
    Communicate to crowdsourcees \(i \in CT_2\) incentives \(\lambda_{i,t}^{CT_1} + \lambda_{i,t}^{CT_2}\)
    Establish Type A ETTs smart contracts for users \(i \in CT_2\)
end while
```

Reconcile payments weekly or monthly
CT\textsubscript{2} with utilities, while the trading transactions among CT\textsubscript{2} users are Type B ETTs. In Fig. 5, we present two scenarios of energy trading transaction for further explanation: (1) ETT Type A where Node 2 decides to sell excess solar energy to the utility, (2) ETT Type B where Node 43 chooses to buy energy from Node 53. The next section presents the outcomes of the two-phase optimization discussed in Section IV.

B. Results and Discussions

1) Phase I: Day-Ahead CES Operation: In order to present the effectiveness of our algorithm, we compare the cases with and without considering the DERs, energy trading among crowdsourcers in grid when solving CES-OPF (7). Fig. 6 shows $P_{\text{org}}$, $P_{\text{org}}^{\text{CES-OPF}}$ and $P_{\text{org}}^{\text{g}}$ (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 crowdsourcers. Fig. 6 also shows the aggregate load profile and generation after solving the CES-OPF for $T = 24$ hrs. The figure shows that battery variable $P_{\text{CES-OPF}}^{b}$ charges when the solar panel produces and injects power $P_{\text{CES-OPF}}^{g}$ into network. The reason why the curve of $P_{\text{CES-OPF}}^{b}$ 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 the section of Islanded Microgrid Test (VI-B3). Fig. 6 indicates that shapeable loads of CT\textsubscript{1} are rescheduled to $P_{\text{CES-OPF}}^{s}$. The updated power generation $P_{\text{CES-OPF}}^{s}$ is smaller than $P_{\text{org}}^{g}$ due to the injections of solar power, scheduling of batteries and shapeable loads from crowdsourcers CT\textsubscript{1}.

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

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

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 DMLPs for both nodes are smaller due to the net injection from Type 1 crowdsorcers (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 every 5–15 minutes depending on the availability of accurate weather/load forecasts. The monetary rewards offered to Type 2 crowdsourcers are obtained from CES-ID. We assume that the crowdsourcers 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 am and 7 pm. The solar panel of Node 2 produces solar power and incentives are earned by the customer between 6 am and 2 pm as shown in Fig. 8. However, the load at Node 2 starts to consume energy at 5 pm making the net injection of Node 2 is 0 MWh. Hence, no monetary incentives are offered from 7 pm to 11 pm. Fig. 9 presents the results for Type B ETTs for CT\textsubscript{2} user at Node 53. The user at Node 43 decides to charge the battery at a constant charging rate between 9 am and 2 pm, 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 crowdsourcers are summarized in Tab. V.

Fig. 10 depicts the aggregate load profile and generation af-
Algorithm 1 terminates. More renewable energy is injected into the grid and traded via the designed incentives for $CT_2$ crowdsourcees. The net contribution of $CT_2$ crowdsourcees is shaded in red. It is noteworthy to mention that the utility cannot schedule the shapeable loads of $CT_2$ crowdsourcees. The blue area in Fig. 10 displays the unexpected load demand of $CT_2$ crowdsourcees. The generator at the substation covers this demand shortage; see Fig. 10 where $P_{CES-ID}^g$ is greater than $P_{CES-OPF}^g$ from 3 pm to 11 pm.

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 (a) enough solar power to produce enough energy to supply the grid, and (b) 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 am and 7 pm, 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 energy trading transactions 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 energy trading transactions 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

The paper introduces the notion of blockchain-assisted crowdsourcees in energy systems (CES) with a specific implementation and prototype of blockchain that scales to include millions of crowdsourcees and P2P energy trading transactions (ETT). 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, shapable loads, and other DERs is introduced for distribution networks—considering ETTs and crowdsourcees preferences—yielding a day-ahead market equilibrium. Then, 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. Then, 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. Case studies are given to illustrate the practicality of the presented methods for classical distribution networks, as well as self-sufficient and islanded microgrids. In future work, we plan to extend the presented research to address (a) distributed consensus mechanisms for blockchain in CESs, and (b) threats from malicious crowdsourcees, market operators, and outsiders.

REFERENCES

[1] S. Wang, A. Taha, and J. Wang, “Blockchain-assisted crowdsourcees in energy systems,” in 2018 IEEE PES General Meeting, Portland, Oregon, August 2018. https://arxiv.org/abs/1802.03099
[2] J. Wang, M. Shahidehpour, and Z. Li, “Security-constrained unit commitment with volatile wind power generation,” IEEE Transactions on Power Systems, vol. 23, no. 3, pp. 1319–1327, 2008.

[3] S. S. Guggilam, E. DallAnese, Y. C. Chen, S. V. Dhople, and G. B. Giannakis, “Scalable optimization methods for distribution networks with high pv integration,” IEEE Transactions on Smart Grid, vol. 7, no. 4, pp. 2061–2070, 2016.

[4] H. Jiayi, J. Chuanwen, and X. Rong, “A review on distributed energy resources and microgrid,” Renewable and Sustainable Energy Reviews, vol. 12, no. 9, pp. 2472–2483, 2008.

[5] G. Carpinelli, G. Celli, S. Mocci, F. Mottola, F. Pilo, and D. Proto, “Optimal integration of distributed energy storage devices in smart grids,” IEEE Transactions on Smart Grid, vol. 4, no. 2, pp. 985–995, 2013.

[6] 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 Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1482–1492, 2018.

[7] M. H. Hajiemsaili, M. Chen, E. Mallada, and C.-K. Chau, “Crowd-sourced storage-assisted demand response in microgrids,” in Proceedings of the Eighth International Conference on Future Energy Systems. ACM, 2017, pp. 91–100.

[8] R. Deng, Z. Yang, M. Y. Chow, and J. Chen, “A survey on demand response in smart grids: Mathematical models and approaches,” IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 570–582, June 2015.

[9] A. Jokic, M. Lazar, and P. van den Bosch, “Real-time control of power systems using nodal prices,” International Journal of Electrical Power & Energy Systems, vol. 31, no. 9, pp. 522 – 530, 2009, power Systems Computation Conference (PSCC) 2008.

[10] C. Langbort, “On real-time pricing for strategic agents,” in Workshop on Multi-Agent Coordination and Estimation, 2010.

[11] 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 Transactions on Smart Grid, vol. 6, no. 6, pp. 2714–2724, Nov 2015.

[12] Y. Yuan and F. Wang, “Blockchain and cryptocurrencies: Model, techniques, and applications,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1421–1428, Sept 2018.

[13] M. Swan, “Blockchain: Blueprint for a new economy.” O’Reilly Media, Inc.”, 2015.

[14] A. Baliga, “Understanding blockchain consensus models,” Tech. rep., Persistent Systems Ltd, Tech. Rep. Tech. Rep., 2017.

[15] https://digiconomist.net/bitcoin-energy-consumption

[16] V. Zamfir, “Introducing casper the friendly ghost, aug. 2015. blog. ethereum.org,” 2015.

[17] https://github.com/hyperledger/fabric

[18] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. bft replication,” in International Workshop on Open Problems in Network Security. Springer, 2015, pp. 112–125.

[19] J. Horta, D. Kofman, and D. Menga, “Novel paradigms for advanced distribution grid energy management,” Telecom ParisTech, 2016.

[20] E. Munsing, J. Mather, and S. Moura, “Blockchain for decentralized optimization of energy resources in microgrid networks,” in Conference on Control Technology and Applications, 2017.

[21] J. Licata, “500 billion reasons blockchain can be a power play in energy,” Mar 2017.

[22] A. Hahn, R. Singh, C. Liu, and S. Chen, “Smart contract-based campus demonstration of decentralized transactive energy auctions,” in 2017 IEEE PES Innovative Smart Grid Technologies, 2017.

[23] J. Cottrell and M. Basden, “How utilities are using blockchain to meet the energy transition. a survey among decision-makers in the german energy industry,” Tech. Rep., 2016.

[24] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov, “Options for control of reactive power by distributed photovoltaic generators,” in Proceedings of the IEEE, vol. 99, no. 6, pp. 1063–1073, 2011.

[25] M. Farivar and S. H. Low, “Branch flow model: relaxations and convexification—part i,” IEEE Transactions on Power Systems, vol. 28, no. 3, pp. 2554–2564, 2013.

[26] Q. Peng and S. H. Low, “Distributed algorithm for optimal power flow on a radial network,” in Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on. IEEE, 2014, pp. 167–172.

[27] S. Mocci, G. Celli, L. Lissia, and P. Terium, “The global energy system is changing, and customers must take centre stage.” https://www.weforum.org/agenda/2017/03/how-customers-are-taking-centre-stage-to-transform-the-energy-system/

[28] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov, “Options for control of reactive power by distributed photovoltaic generators,” Proceedings of the IEEE, vol. 99, no. 6, pp. 1063–1073, 2011.

[29] J. Cottrell and M. Basden, “How utilities are using blockchain to modernize the grid.” Harvard Business Review, Mar 2017.

[30] https://github.com/hyperledger/fabric

[31] [32] E. Munsing, J. Mather, and S. Moura, “Blockchain for decentralized optimization of energy resources in smartgrid networks,” in Control Technology and Applications (CCTA), 2017 IEEE Conference on. IEEE, 2017, pp. 2164–2171.

[33] M. Baran and F. F. Wu, “Optimal sizing of capacitors placed on a radial distribution system,” IEEE Transactions on power Delivery, vol. 4, no. 1, pp. 735–743, 1989.

[34] M. Fairvar and S. H. Low, “Branch flow model: relaxations and convexification—part i,” IEEE Transactions on Power Systems, vol. 28, no. 3, pp. 2554–2564, 2013.

[35] Q. Peng and S. H. Low, “Distributed algorithm for optimal power flow on a radial network,” in Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on. IEEE, 2014, pp. 167–172.

[36] M. Baran and F. F. Wu, “Optimal capacitor placement using nodal prices,” International Journal of Electrical Power & Energy Systems, vol. 10, no. 3, pp. 167–175, 1988.

[37] M. Swan, “Blockchain and cryptocurrencies: Model, techniques, and applications,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1421–1428, Sept 2018.