Decentralized Optimal Power Flow in Distribution Networks Using Blockchain

Tarek AlSkaif and Gijs van Leeuwen
Copernicus Institute of Sustainable Development, Utrecht University, Utrecht, The Netherlands

***Abstract*—The rapid development of distributed energy resources (DER) in the distribution grid calls for novel control and coordination solutions. Optimal management of DER will enable end-users to decrease their electricity costs and provide crucial services to grid operators. In this paper, a decentralized Optimal Power Flow (OPF) model is used to locally coordinate DER in distribution networks, while considering the network constraints, in a distributed, transparent and secure fashion. To achieve that, a consensus-based distributed optimization algorithm is developed using the general form Alternating Direction Method of Multipliers (ADMM). To enable transparent and verifiable management of the network, the paper provides a comprehensive procedure for the implementation of the decentralized OPF on a private blockchain-smart contracts platform. The performance of the proposed framework is tested using real data from a case study in a residential neighborhood in Amsterdam with different varieties of DER. The implementation procedure on a blockchain-smart contracts platform may be adopted in other problems that require a smart contract to act as a virtual aggregator.**

***Index Terms*—Distributed energy resources, optimal power flow, distributed optimization, consensus algorithms, blockchain**

I. Introduction

The rapid growth and adoption of distributed energy resources (DER) in the distribution network, such as rooftop Photovoltaics (PV) panels and electric vehicles (EV), calls for novel management solutions for coordinating these DER. Optimally managed DER can enhance end-users flexibility, which in turn results in electricity cost reduction, provides crucial services to grid operators and enables the matching of demand and (local) renewable energy supply [1]. For example, battery energy storage and EV charging can be scheduled to periods of the day when PV energy is available, or when electricity prices are low [2].

Proposed management schemes of DER in the low-voltage (LV) network have typically assumed that the operation of DER is centrally managed by an aggregator, a cooperative utility or a microgrid operator [3], [4]. However, these centralized schemes suffer from scalability issues when the number of DER is large [5]. Besides, they are typically met with low acceptance by households due to limited financial incentives and privacy concerns [6], [7]. In addition to considering DER operation constraints for optimal economic dispatch, some of these centralized control schemes also consider the LV network constraints by means of solving the celebrated Optimal Power Flow (OPF) problem [8].

Solving an OPF problem optimizes the power flow in the network while ensuring that the system constraints are not violated. Various decentralized versions of the OPF problem have been proposed in literature which are surveyed in [9]. These decentralized versions typically employ decomposition techniques that achieve optimality in a distributed manner. One of the most common techniques to solve decentralized optimization problems in this area is the Alternating Direction Method of Multipliers (ADMM). ADMM is based on a decomposition coordination procedure that allows to define local subproblems for each node in a LV network or a microgrid. The solutions of those local subproblems are coordinated to solve the large global problem of the whole network [10].

However, without a central agent (i.e., an aggregator or a microgrid operator) the exchange of information and the validation and verification of the OPF solution for the whole network becomes a challenge. Information and Communication Technologies (ICT) are being broadly employed in the energy system and some emerging technologies such as blockchain-smart contracts can enable the distribution of the central agent role across all nodes in the LV network. Indeed, blockchain technology has recently received significant attention in the area of smart energy systems as potential solution for decentralized data storage and management as well as computation without third-party supervision [7], [11].

The aim of this work is to provide a comprehensive framework for implementing the decentralized ADMM-based OPF on a private blockchain network that utilizes a smart contract to act as a virtual aggregator. From the family of ADMM-based consensus methods, we use the general form ADMM-based consensus optimization class [10] since it represents a natural fit to the OPF problem [12]. Each end-user optimizes its own DER operation and power flow individually whilst optimality of the whole network is achieved via a transparent and secure blockchain-smart contracts platform using the ADMM consensus algorithm. The paper provides a detailed modeling of the consensus method and shows how a smart contract can act as a virtual aggregator. Several steps in the ADMM algorithm are distinguished where communication between the end-users and the smart contract occurs to illustrate which information must be shared by end-users with the blockchain network. A case study from a realistic LV-network is used for testing the performance of the proposed framework for a group of households with different varieties of DER.

The paper is structured as follows. System design is presented in Section II. The OPF model and the ADMM-based consensus formulation are described in Section III. Section IV is dedicated to illustrate the implementation of the decentralized OPF on the blockchain-smart contracts platform. A case study is provided in Section V. Finally, the paper is concluded in Section VI.

978-1-7281-1156-8/19/$31.00

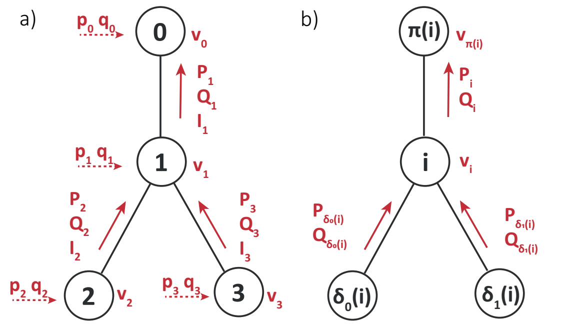

Figure 1: An example of a graph-based network of 4 nodes.

II. System Design

In this work we consider a radial LV-grid that resembles a network of power lines. This network can be modeled as a graph with a set of nodes $\mathcal{N}$, indexed by $i = 0, 1, \ldots, n$, and a set of edges (i.e., lines) $\mathcal{E}$, where node 0 refers to the root node (i.e., the feeder 0). A line in $\mathcal{E}$ is denoted by the pair of nodes it connects (e.g., ij, where j is closer to the root node). The neighbouring nodes of a particular node i can be regarded as the parent and children of that node. The node closer to the root node j is called the parent of i and is denoted by $\pi(i)$, while i is called the child of j. The set of children of a node j is denoted as $\delta(j) := \{i : (i, j) \in \mathcal{E}\}$ [13]. For simplicity, the number of a line is assigned the number of the child node.

The complex line impedance is denoted as $z_{\text{i}} = r_{\text{i}} + \mathtt{i}x_{\text{i}}$, where r_{i} and x_{i} are the line resistance and reactance, respectively. v_{i} is the complex voltage on each node $i \in \mathcal{N}$ and I_{i} is the complex current flowing from node i to j (i.e., or to its parent node $\pi(i)$). The complex power flow from node i to node j is denoted $S_{\text{i}} = P_{\text{i}} + \mathtt{i}Q_{\text{i}}$, where P_{i} and Q_{i} are the real and reactive power flow from node i to node j, respectively. We follow the convention that departing power is positive (i.e., when it flows along the line in the direction of the edge) and arriving power is negative (i.e., in the direction opposite the edge). Each edge can support power flow in either direction up to a given maximum apparent power capacity $\overline{S_{\text{i}}}$. Fig. 1 shows a simplified example of a graph-based network of 4 nodes.

A dispatchable thermal generator is attached to the feeder 0 to provide power to the network with a maximum generation capacity G_{max}. At any time t, the generator has a cost function $C_{\text{i,t}}(p^{\text{g}}_{\text{i,t}})$ indicating the cost of generating electricity by the energy source for each node in the network i (i.e., the cost of the same load can be different at different times of the day). We assume the thermal generator has a quadratic cost function that is increasing and strictly convex [14] as

$$C_{\text{i,t}}(p^{\text{g}}_{\text{i,t}}) = \alpha_{\text{i,t}} p^{\text{g}^2}_{\text{i,t}} + \beta_{\text{i,t}} p^{\text{g}}_{\text{i,t}} + \gamma_{\text{i,t}}. \tag{1}$$

Each node in the considered LV network represent a household. A node could also be any other building or DER in the network, such as a community battery or an EV charging station. Households have a base electricity demand at each period of the day that is uncontrollable. A household might have different varieties of DER (e.g., an on-site PV system and/or an EV). We assume that the Vehicle-to-Grid (V2G) service is not provided by EV in this study. Households with PV systems (i.e., prosumers) can generate electricity locally at some periods of the day. Households with EV have a flexible demand that represents the amount of electricity needed to charge their EV, which can be shifted during some periods of the day. There is an Energy Management System (EMS) in each household that is responsible for optimization and communication with the different system components.

III. Optimization Problem Formulation

A. Branch Flow Model

In this work we adapt the branch flow formulation to model the AC power flow at the steady-state in the considered single-phase radial network [15]. The branch flow model is a Second-order Cone (SOC) convex relaxation of the AC OPF problem [8], [16]. We consider a multi-period OPF where at each period t, the branch flow equations are derived as follows. Consider the complex power loss from node i to j

$$|I_{\text{i,t}}|^2 z_{\text{i,t}} = \frac{P^2_{\text{i,t}} + Q^2_{\text{i,t}}}{|v_{\text{i,t}}|^2} z_{\text{i,t}}, \tag{2}$$

The squared voltage and current magnitudes are denoted ν_{i} and ψ_{i}, respectively, and (2) can be rewritten as

$$\psi_{\text{i,t}} z_{\text{i,t}} = \frac{P^2_{\text{i,t}} + Q^2_{\text{i,t}}}{\nu_{\text{i,t}}} z_{\text{i,t}}, \tag{3}$$

Assuming that the departing power from node i to j is positive and the arriving power to j from i (i.e., P_{j} and Q_{j}) is negative, the real and reactive line losses can be written as

$$P^2_{\text{i,t}} + Q^2_{\text{i,t}} = \psi_{\text{i,t}} \nu_{\text{i,t}}, \tag{4}$$

$$P_{\text{i,t}} + P_{\text{j,t}} = r_{\text{i,t}} \psi_{\text{i,t}}, \tag{5}$$

$$Q_{\text{i,t}} + Q_{\text{j,t}} = x_{\text{i,t}} \psi_{\text{i,t}}. \tag{6}$$

The above equations define the power through line ij into node j. The real and reactive power injection by node i are the power flows from i to j minus the sums of the flows through the lines that arrive to i (i.e., from i's children $\delta_{(i)}$) considering the losses on those lines. This can be formulated as

$$p_{\text{i,t}} = P_{\text{i,t}} - \sum_{k \in \delta_{(i,t)}} (P_{\text{k,t}} - r_{\text{k,t}} \psi_{\text{k,t}}), \quad \forall i, t, \tag{7}$$

$$q_{\text{i,t}} = Q_{\text{i,t}} - \sum_{k \in \delta_{(i)}} (Q_{\text{k,t}} - x_{\text{k,t}} \psi_{\text{k,t}}), \quad \forall i, t, \tag{8}$$

where $p_{\text{i,t}}$ and $q_{\text{i,t}}$ are the difference between the real and reactive power generation on one hand and real and reactive power consumption on the other hand, such that $p_{\text{i,t}} = p^g_{\text{i,t}} - p^d_{\text{i,t}}$ and $q_{\text{i,t}} = q^g_{\text{i,t}} - q^d_{\text{i,t}}$. We note that $p^g_{0,\text{t}} - p^d_{0,\text{t}}$ can also be interpreted as the total real power injection into the network from the main grid through the feeder 0.

The voltage at the node j, the parent node of i (also denoted as $v_{\pi(i)}$), can be defined by Ohm's law, as

$$v_{\text{j,t}} = v_{\text{i,t}} - I_{\text{i,t}} z_{\text{i,t}}, \quad \forall j, t. \tag{9}$$

By taking the squared complex magnitude of each side of (9)

$$\nu_{\text{j,t}} = |v_{\text{i,t}} - I_{\text{i,t}} z_{\text{i,t}}|^2, \quad \forall j, t, \tag{10}$$

$$\nu_{\text{j,t}} = \nu_{\text{i,t}} - 2\text{Re}[v_{\text{i,t}} I_{\text{i,t}} z_{\text{i,t}}] + |I_{\text{i,t}}|^2 |z_{\text{i,t}}|^2, \quad \forall j, t, \tag{11}$$

$$\nu_{\text{j,t}} = \nu_{\text{i,t}} - 2(r_{\text{i,t}} P_{\text{i,t}} + x_{\text{i,t}} Q_{\text{i,t}}) + (r^2_{\text{i,t}} + x^2_{\text{i,t}}) \psi_{\text{i,t}}, \quad \forall j, t. \tag{12}$$

The only remaining nonconvexity in this formulation is (3), which can be relaxed by changing the equality to "$\leq$". This should not change the resulting optimum since increasing current, and hence line losses, should not improve most realistic objectives [8]. Thus, (3) is relaxed as follows, which results in a standard SOC form

$$P_{\text{i,t}}^2 + Q_{\text{i,t}}^2 \leq \psi_{\text{i,t}}\nu_{\text{i,t}}, \quad i = 1, \ldots, n, \forall t. \tag{13}$$

Equations (7), (8), (12) and (13) define a system of equations in the variables $(P, Q, \nu, \psi) := (P_{\text{i}}, Q_{\text{i}}, \psi_{\text{i}}, (i, j) \in \mathcal{E}, i = 1, \ldots, n)$. This system of equations do not include phase angles of voltages and currents which can be determined for radial networks given (P, Q, ν, ψ) [16].

In addition to these equations, and in order to complete the formulation of the AC-OPF problem, the following power flow limits need to be respected

$$\underline{v}_{\text{j,t}}^2 \leq \nu_{\text{j,t}} \leq \overline{v}_{\text{j,t}}^2, \quad \forall j, t, \tag{14}$$

$$\underline{p}_{\text{i,t}}^g \leq p_{\text{i,t}}^g \leq \overline{p}_{\text{i,t}}^g, \quad \forall i, t, \tag{15}$$

$$\underline{p}_{\text{i,t}}^d \leq p_{\text{i,t}}^d \leq \overline{p}_{\text{i,t}}^d, \quad \forall i, t, \tag{16}$$

$$\underline{q}_{\text{i,t}}^g \leq q_{\text{i,t}}^g \leq \overline{q}_{\text{i,t}}^g, \quad \forall i, t, \tag{17}$$

$$\underline{q}_{\text{i,t}}^d \leq q_{\text{i,t}}^d \leq \overline{q}_{\text{i,t}}^d, \quad \forall i, t, \tag{18}$$

$$P_{\text{i,t}}^2 + Q_{\text{i,t}}^2 \leq \overline{S_{\text{i,t}}}^2, \quad i = 1, \ldots, n, \forall t. \tag{19}$$

As mentioned in Section II, each node in the network represents a household (i.e., either a consumer or a prosumer) Further, some households also own an EV. Therefore, some nodes have additional local private constraints which must be respected. An EV at node i is considered as shiftable load with an average daily charging demand ($E_{\text{ev, av}}$) which must be met such as

$$\sum_{t=1}^{\mathtt{T}} p_{\text{i,t}}^{\text{ev}} \Delta t = E_{\text{ev, av}}, \quad \forall i, \tag{20}$$

where $p_{\text{i,t}}^{\text{ev}}$ is a decision variable for scheduling EV charging power at households with EV. The EV charging demand ($E_{\text{ev, av}}$) is equal to zero at households without EV.

EV charging power should be in between the upper ($\overline{p}^{\text{ev,d}}$) and lower ($\underline{p}^{\text{ev,d}}$) charging limits to ensure that the EV power rating is respected. The parameter $\omega_{\text{i,t}}$ is used as an input for scheduling EV charging at a certain timeslots (i.e., $\omega_{\text{i,t}}$=1 when an EV is plugged in and can charge, and =0 otherwise) as

$$\omega_{\text{i,t}}\underline{p}^{\text{ev,d}} \leq p_{\text{i,t}}^{\text{ev}} \leq \omega_{i,t}\overline{p}^{\text{ev,d}}, \quad \forall t, i. \tag{21}$$

At each t, the following system balance constraint for each node i applies, which balances the dispatchable thermal generation (i.e., at the feeder 0) for node i ($p_{\text{i,t}}^{\text{g}}$) and the local PV electricity supply ($p_{\text{i,t}}^{\text{pv}}$) with the baseload demand ($p_{\text{i,t}}^{\text{base}}$) and EV ($p_{\text{i,t}}^{\text{ev}}$) demand as

$$p_{\text{i,t}} = p_{\text{i,t}}^{\text{base}} + p_{\text{i,t}}^{\text{ev}} - p_{\text{i,t}}^{\text{pv}} - p_{\text{i,t}}^{\text{g}}, \quad \forall i, t, \tag{22}$$

where the local PV supply is equal to zero at households without a PV system.

The market objective in this paper is to minimize the total power generation costs over a period of time T by maximizing the use of locally produced energy from the PV systems and scheduling the shiftable demand of the EV. The objective function can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N}\sum_{t=1}^{T} C_{\text{i,t}}(p_{\text{i,t}}^{\text{g}}) \\ \text{subject to} \quad & (7), (8) \text{ and } (12) - (22) \end{aligned} \tag{23}$$

B. General Form Consensus Optimization

Consensus problems provide a general framework for distributed optimization. In this work, we use an ADMM-based consensus optimization problem to solve the multi-period AC-OPF formulated in (23) in a decentralized manner. For the sake of clarity, the time index from each variable is omitted.

From the family of ADMM-based consensus optimization formulations, we adapt the Global Form Consensus Optimization where there are a number of local variables $x_l \in R^{c_l}, l = 1, \ldots, L$ that are related to their subproblems only, with the objective $f_1(x_1) + \ldots + f_L(x_L)$ separable in x_l. Each of these local variables consists of different components (c_l) and each local variable component $(x_l)_c$ corresponds to a global variable component $z_g \in R^c$, where the mapping from local variable indices into a global variable index can be written as $g = \mathcal{G}(l, c)$. This means that a local variable component $(x_l)_c$ corresponds to a global variable component z_g. Global variables couple subproblems together and the consensus between local and global variables is achieved when

$$(x_l)_c = z_{\mathcal{G}(l,c)}, \quad l = 1, \ldots, L, \quad c = 1, \ldots, c_l. \tag{24}$$

We define the variable $\tilde{z}_l \in R^{c_l}$ which is the global variable's idea of what the local variable x_l should be (i.e., $(\tilde{z}_l)_c = z_{\mathcal{G}(l,c)}$). The consensus constraint can then be written as $x_l - \tilde{z}_l = 0, \quad l = 1, \ldots, L$. Consequently, the general form consensus problem can be written as

$$\begin{aligned} \text{minimize} \quad & \sum_{l=1}^{L} f_l(x_l) \\ \text{subject to} \quad & x_l - \tilde{z}_l = 0, \quad l = 1, \ldots, L \end{aligned} \tag{25}$$

The augmented Lagrangian of this problem is:

$$L_\rho(x, z, y) = \sum_{l=1}^{L}(f_l(x_l) + y_l^{\intercal}(x_l - \tilde{z}_l) + (\rho/2)\left\|x_l - \tilde{z}_l\right\|_2^2), \tag{26}$$

where $y_l \in R^{c_l}$ is the dual variable. The general form consensus problem of ADMM consists of the iterations

$$x_l^{k+1} := \underset{x_l}{\operatorname{argmin}}(f_l(x_l) + y_l^{k\intercal} x_l + (\rho/2)\left\|x_l - \tilde{z}_l^k\right\|_2^2) \tag{27}$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}}\Big(\sum_{l=1}^{m}(-y_l^{\intercal}\tilde{z}_l + (\rho/2)\left\|x_l^{k+1} - \tilde{z}_l\right\|_2^2)\Big) \tag{28}$$

$$y_l^{k+1} := y_l^k + \rho(x_l^{k+1} - \tilde{z}_l^{k+1}), \tag{29}$$

where k is the iteration and for each l, x_l- and y_l-updates at each k can be carried out locally, independently and in parallel.

The z-update step decouples across the components of z and is found by averaging all entries of $x_l^{k+1} + (1/\rho)y_l^k$ that correspond to the global index g as

$$z_g^{k+1} := \frac{\sum_{\mathcal{G}(l,c)=g}\left((x_l^{k+1})_c + (1/\rho)(y_l^k)_c\right)}{\sum_{\mathcal{G}(l,c)=g} 1}. \tag{30}$$

With the average of a vector denoted with an overline, the z-update can be written as

$$z_g^{k+1} = \overline{x}^{k+1} + (1/\rho)\overline{y}^k. \tag{31}$$

And in a similar manner, the y-update can be written as

$$\overline{y}^{k+1} = \overline{y}^k + \rho(\overline{x}^{k+1} - z_g^{k+1}). \tag{32}$$

Substituting (31) in (32) shows that after the first iteration

$$\sum_{\mathcal{G}(l,c)=g} (y_l^k)_c = 0, \tag{33}$$

Therefore, the z-update can be written as

$$z_g^{k+1} := (1/k_g) \sum_{\mathcal{G}(l,c)=g} (x_l^{k+1})_c, \tag{34}$$

where k_g is the number of a local variable's components that correspond to a global variable component z_g. In other words, the z-update is calculated by averaging all local copies of the global variable z_g.

For consensus ADMM, the primal and dual residuals are defined as

$$r^k = x_l^k - \tilde{z}_l^k, \quad s^k = -\rho(z_g^k - z_g^{k-1}). \tag{35}$$

The consensus is obtained when the ADMM algorithm converges which is determined using the following conditions

$$\left\|r^k\right\|_2 \leq \epsilon_p, \quad \left\|s^k\right\|_2 \leq \epsilon_d, \tag{36}$$

where ϵ_p and ϵ_d are the tolerances in the primal and dual residuals which are typically assigned a very low value.

C. OPF and ADMM-based General Form Consensus

The AC-OPF of (23) can be reformulated in a distributed manner using the ADMM-based general form consensus algorithm. In this form, the global optimization problem can be split into a number of subproblems. Every node will solve its own subproblem locally and independently using its own optimization objective and set of constraints. The local subproblem for a node i can be formulated as

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T} C_{\mathrm{i,t}}(p_{\mathrm{i,t}}^{\mathrm{g}}) \\ &\text{subject to} \quad (7), (8) \text{ and } (12) - (22) \end{aligned} \tag{37}$$

The variables in the constraints can be divided in two parts. The set of local private variables of a node contains the variables pertaining to its local energy infrastructure, and is denoted as $x_p := [p_i, p_i^{\mathrm{g}}, q_i, q_i^{\mathrm{g}}, p_i^{\mathrm{ev}}, \omega_i, \psi_{\mathrm{i}}]$. The local private variables are not shared with the network and remain private. The set of coupling variables of a node contains the local variables pertaining to its set of branch flow equations and is denoted as $x_l := [P_i, Q_i, v_i, P_{\delta(i)}, Q_{\delta(i)}, v_{\pi(i)}]$. Every coupling variable corresponds to a particular global variable, the set of which is denoted as $z_g := [P, Q, v]$. Fig. 2 shows how the locally calculated coupling variables correspond to the global variables in the present OPF problem. Since every node will only have to calculate variables corresponding to the adjacent edges and nodes, the full network topology will not

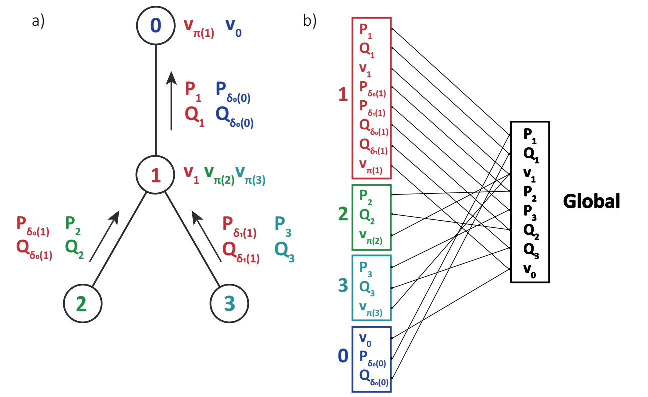

Figure 2: An illustration of the ADMM-based general form consensus method for the OPF problem in a 4-nodes network.

have to be known by any node. Every node must only know who its neighbouring nodes (i.e., parent and children) are.

Using the general form consensus ADMM formulation described in Section III-B, in iteration $k+1$ each node i receives z^{k} and solves the following local optimization problem

$$\text{minimize} \sum_{t=1}^{T} C_{\mathrm{i,t}}(p_{\mathrm{i,t}}^{\mathrm{g}}) + y_l^{k\intercal}(x_l - \tilde{z}_l^k) + (\rho/2)\left\|x_l - \tilde{z}_l^k\right\|_2^2 \tag{38}$$

The nodes then send their coupling variables to the aggregator, which updates z as

$$z_g^{k+1} := (1/k_g) \sum_{\mathcal{G}(l,c)=g} (x_l^{k+1})_c \tag{39}$$

The nodes then receive $\tilde{z}_l^{k+1}$ and update the penalty as

$$y_l^{k+1} = y_l^k + \rho(x_l^{k+1} - \tilde{z}_l^{k+1}) \tag{40}$$

Finally, the values for the residuals are evaluated using (35) and (36). If the conditions in (36) are satisfied, the optimization process is complete. If not, the ADMM algorithm will proceed with the next iteration.

IV. Implementation on Blockchain

A. Blockchain and the role of the smart contract

Blockchain is an emerging technology for decentralized data storage and management as well as computation. It is secured by a combination of cryptographic signatures and a distributed consensus mechanism. Agents on the blockchain network are able to come to a universal agreement on the system state at each time step, even in the presence of cyberattacks and participants joining/departing the network. Therefore, a blockchain network allows agents to connect in a secure fashion without reliance on a third party operation of the platform. Possible applications for blockchain have been recognized in many areas including in the energy sector [11].

The utility of blockchain technology can be expanded through the use of smart contracts. A smart contract is a piece of computer code that is deployed on the blockchain and can execute certain functions when called upon by other agents [17]. Smart contract technology allows decentralized optimization on a blockchain network, enabling execution of the distributed OPF/ADMM algorithm without dependence on a central agent (e.g., a third party aggregator). The smart contract takes over the function of this central aggregator, thus effectively functioning as a virtual aggregator. In this role, the smart contract acts as the primary agent for nodes to interact with during execution. Parts of the algorithm are moved to the smart contract, and essential data that is needed by the nodes for optimization is retrieved from the contract. The contract also ensures that all nodes operate simultaneously by giving permission to proceed with the next operation after all nodes have declared completion of the previous section. The inherent built-in delay in the blockchain verification process may make the use of blockchain for real-time optimization implausible or impractical. Therefore, the platform proposed in this study is intended to provide a day-ahead forecast optimization of energy flows.

B. *Workflow of the network*

Upon setting up the blockchain network every node i is assigned a personal account with address λ_i. This account is used to interact with the network and send/retrieve data to/from the smart contract. The smart contract σ is deployed to the network using the set of constructor variables $\theta := [n, \rho, \epsilon_p, \epsilon_d, \mu]$ that configure the algorithm. The variable μ represents the maximum number of iterations. As the contract is deployed, a new address λ_σ and the bytecode of the contract's contents ABI_σ are generated. λ_σ and ABI_σ must be known by all other nodes to allow them to interact with the contract. After the contract is deployed it will remain on the network indefinitely.

In the execution of the ADMM algorithm on the blockchain network for OPF there are several steps that can be distinguished where there is communication between the smart contract σ and the nodes $i \in \mathcal{N}$. These steps are visualised in Fig. 3. In step 1), i connects to σ by using the adress λ_σ and bytecode ABI_σ. This action only has to be performed once. In step 2) a new round of optimization starts, and all nodes will declare their participation to the smart contract by passing its number i and the numbers of their parent node $\pi(i)$ and children nodes $\delta(i)$. As all nodes send this information, full network topology will be implicitly known by the contract without having to state it explicitly. Also, the nodes will retrieve θ from the contract to configure the local optimization problem. When all nodes have declared participation, the contract will communicate to all nodes that the optimization process may commence. When local optimization (38) is complete, the nodes send their sets of coupling variables x_l to the smart contract in step 3) and wait for further instructions. When all nodes submitted their coupling variables the smart contract will execute the z-update step (39) of ADMM. When the z-update step is complete, the recalculated global variables are sent back to the nodes in step 4) which will then calculate the new penalty value (40). The nodes will also calculate the partial residual value for their subproblem and send this to the contract in step 5). This contract then calculates the global residual and sends this value back to the nodes in step 6). The nodes evaluate the convergence conditions (36) and revert to step 3) to start a new iteration if the conditions are not satisfied.

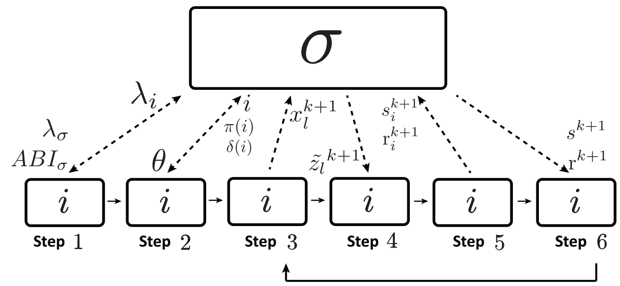

Figure 3: A flowchart showing the interaction between the smart contract σ and node i in the steps of the ADMM algorithm.

V. Case Study

The system is tested using measured consumption and generation data from 23 households who are part of the East Harbour Prosumers Community in Amsterdam [18] and a test distribution network whose topology and parameters are the same as the first 23 bus network described in [16]. Of these households, 8 are prosumers with a PV system, 3 are prosumers with a PV system and an EV, 8 are consumer households and 3 are consumer households with an EV. We assume that each household with EV drives 36 km each day, which is the average daily distance travelled in the Netherlands [19], with a driving efficiency of 5 km per kWh. This results in an average EV daily charging demand ($E_{\text{ev, av}}$) of 7.06 [kWh]. The charging scheduling input parameter ω_i is different for every EV owner, with some households allowing scheduling during the day and others during the night. For the cost function of the thermal generator at feeder 0, we set the values of $\alpha = 1$, $\beta = 10$ and $\gamma = 100$. For the reactive power load, we use a random number generator where the reactive power load is between 0 and -1 in every timeslot. It is assumed that the households do not have access to a private connection to the external grid.

For testing the proposed framework, generation and consumption data on 21 June 2018 as an arbitrary day is used. Python and CVXPY [20] are used to solve the local optimization problems. Ganache-cli [21] is used to run a private Ethereum blockchain network and Web3.py [22] is used to enable communication between Python and the network. The convergence of the algorithm is visualised in Fig. 4. The algorithm is run for $\mu = 300$ iterations, at which point the total costs are 0.8 percent higher than in a centralized solution. The total generation schedule of the feeder 0 and the scheduling of EV charging for all households are visualised in Fig. 5, as well as the total PV generation and the total base load. It can be seen that the thermal generator is most active during the peak hours in early morning and early evening, when there is no solar generation and load is relatively high. It can also be seen that EV charging is lowest during these times, and is instead scheduled during the day and night.

VI. Conclusions

In this study it is shown how blockchain-smart contracts and ADMM can be used to implement a decentralized OPF

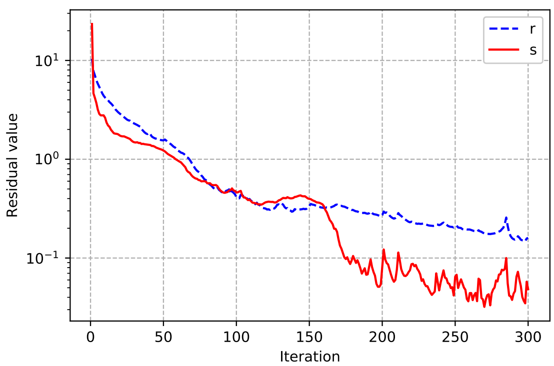

Figure 4: Convergence of the ADMM algorithm.

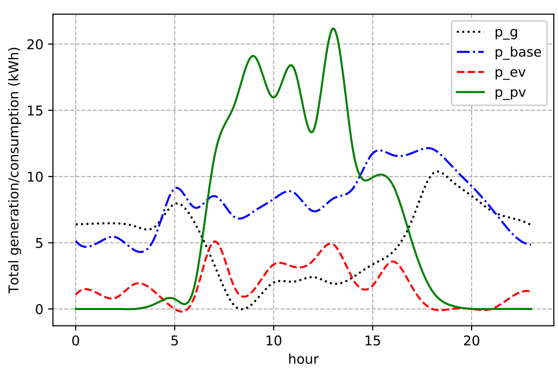

Figure 5: Total electricity generation and consumption schedule of all housheholds on 21 June 2018.

algorithm in a LV network or a microgrid environment to optimize electricity flows between households that possess different varieties of DER. We have shown how the OPF algorithm can be decomposed into local subproblems that are solved locally by every household and how ADMM is used to ensure consensus between the different households on the final state of the system. Finally, it is shown how the ADMM algorithm is implemented on a blockchain network making use of smart contracts technology. The proposed framework achieves the desired goal of optimization whilst providing security from tampering and cyberattacks, as well as independence from a central agent. Limitations arise from the fact that the model has only been tested on a private blockchain node in a closed environment. For a more elaborate assessment of practical applicability in terms of speed and security, the model could be deployed and tested on a live blockchain network. Furthermore, this study does not assume any trading between the different households. In future work, the usefulness of the platform and the smart contract can be expanded by including a trading mechanism that allows payments between households to take place based on the quantity of electricity that is shared with the network.

References

[1] J. Driesen and F. Katiraei, "Design for distributed energy resources," *IEEE Power and Energy Magazine*, vol. 6, no. 3, 2008.

[2] T. AlSkaif, A. C. Luna, M. G. Zapata, J. M. Guerrero, and B. Bellalta, "Reputation-based joint scheduling of households appliances and storage in a microgrid with a shared battery," *Energy and Buildings*, vol. 138, pp. 228–239, 2017.

[3] S. M. Nosratabadi, R.-A. Hooshmand, and E. Gholipour, "A comprehensive review on microgrid and virtual power plant concepts employed for distributed energy resources scheduling in power systems," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 341–363, 2017.

[4] I. Lampropoulos, T. Alskaif, J. Blom, and W. van Sark, "A framework for the provision of flexibility services at the transmission and distribution levels through aggregator companies," *Sustainable Energy, Grids and Networks*, vol. 17, p. 100187, 2019.

[5] S. Mhanna, A. C. Chapman, and G. Verbič, "A fast distributed algorithm for large-scale demand response aggregation," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2094–2107, 2016.

[6] C. Sijie and L. Chen-Ching, "From demand response to transactive energy: state of the art," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 1, pp. 10–19, 2017.

[7] E. Münsing, J. Mather, and S. Moura, "Blockchains for decentralized optimization of energy resources in microgrid networks," in *Control Technology and Applications (CCTA), 2017 IEEE Conference on*. IEEE, 2017, pp. 2164–2171.

[8] J. A. Taylor, *Convex optimization of power systems*. Cambridge University Press, 2015.

[9] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized dc optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2018.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[12] M. Kraning, E. Chu, J. Lavaei, S. Boyd *et al.*, "Dynamic network energy management via proximal message passing," *Foundations and Trends® in Optimization*, vol. 1, no. 2, pp. 73–126, 2014.

[13] N. Li, "A market mechanism for electric distribution networks," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2276–2282.

[14] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE transactions on Smart Grid*, vol. 1, pp. 320–331, 2010.

[15] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[16] N. Li, L. Chen, and S. H. Low, "Exact convex relaxation of OPF for radial networks using branch flow model," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2012, pp. 7–12.

[17] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[18] EHPC, "East Harbour Prosumers Community platform for sustainable energy use)," 2019. [Online]. Available: http://www.prosumers.nl/

[19] CBS, "CBS StatLine - Verkeersprestaties personenauto's; kilometers, brandstofsoort, grondgebied," 2018. [Online]. Available: https://bit.ly/2OJtaCy

[20] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[21] TruffleSuite, "Ganache-cli," 2019. [Online]. Available: https://github.com/trufflesuite/ganache-cli

[22] Ethereum, "Web3.py," 2019. [Online]. Available: https://https://github.com/ethereum/web3.py

VII. Acknowledgements

This work is funded by the Dutch TKI Urban Energy project 1621404: "A Blockchain-based platform for peer-to-peer energy transactions between Distributed Energy Resources (B-DER)".