Deep Reinforcement Learning for Demand Response in Distribution Networks

Shahab Bahrami, *Member, IEEE*, Yu Christine Chen, *Member, IEEE*, and Vincent W.S. Wong, *Fellow, IEEE*

***Abstract*—Load aggregators can use demand response programs to motivate residential users toward reducing electricity demand during peak time periods. This paper proposes a demand response algorithm for residential users, while accounting for uncertainties in the load demand and electricity price, users' privacy concerns, and power flow constraints imposed by the distribution network. To address the uncertainty issues, we develop a deep reinforcement learning (DRL) algorithm using an actor-critic method. We apply federated learning to enable users to determine the neural network parameters in a decentralized fashion without sharing private information (e.g., load demand, users' potential discomfort due to load scheduling). To tackle the nonconvex power flow constraints, we apply convex relaxation and transform the problem of updating the neural network parameters into a sequence of semidefinite programs (SDPs). Simulations on an IEEE 33-bus test feeder with 32 households show that the proposed demand response algorithm can reduce the peak load by 33% and the expected cost of each user by 13%. Also, we demonstrate the scalability of the proposed algorithm in 330-bus and 1650-bus feeders with real-time pricing scheme.**

***Keywords*: demand response, deep reinforcement learning, federated learning, power flow, semidefinite program.**

List of Key Notations

$\mathcal{N}$	Set of buses
$\mathcal{N}^-$	Set of households
$\mathcal{L}$	Set of transmission lines
$\mathcal{T}$	Set of time slots
$P^{\mathrm{b}}_{n,t}$	Base load in household n in time slot t
$P^{\mathrm{c,min}}_{n,t}$	Lower bound for controllable load in household n in time slot t
$P^{\mathrm{c,des}}_{n,t}$	Desirable value for controllable load in household n in time slot t
$\boldsymbol{s}_{n,t}$	State of household n in time slot t
ρ_t	Electricity price in time slot t
$\boldsymbol{s}_t$	System state in time slot t
$\mathcal{S}$	Set of system states
$P^{\mathrm{c}}_n(\boldsymbol{s}_t)$	Scheduled controllable load demand in household n in time slot t
$\boldsymbol{a}_n(\boldsymbol{s}_t)$	Action vector for household n in state $\boldsymbol{s}_t$
$\boldsymbol{a}_N(\boldsymbol{s}_t)$	Action vector for substation bus N in state $\boldsymbol{s}_t$
$\boldsymbol{a}(\boldsymbol{s}_t)$	Joint action profile in state $\boldsymbol{s}_t$
$\mathcal{A}(\boldsymbol{s}_t)$	Feasible action space
$d_n(\cdot)$	Discomfort cost of household n
$c_n(\cdot)$	Cost (i.e., bill payment and discomfort cost) of household n
$c(\cdot)$	Social cost of all households
$\boldsymbol{\pi}(\boldsymbol{s})$	Policy in an arbitrary state $\boldsymbol{s}$
$\boldsymbol{\pi}$	Policy profile, i.e., $\boldsymbol{\pi} = (\boldsymbol{\pi}(\boldsymbol{s}),\ \boldsymbol{s} \in \mathcal{S})$
$V^{\boldsymbol{\pi}}(\boldsymbol{s})$	Value function in state $\boldsymbol{s}$
β	Discount factor in interval $[0,\ 1)$
$\widetilde{P}^{\mathrm{c}}_n(\boldsymbol{s}_t)$	Load control action before projection for household n in state $\boldsymbol{s}$
$\widetilde{\boldsymbol{P}}^{\mathrm{c}}(\boldsymbol{s}_t)$	Load control action vector before projection in state $\boldsymbol{s}_t$
$\mathcal{A}^{\mathrm{c}}(\boldsymbol{s}_t)$	Feasible space for load control action in state $\boldsymbol{s}_t$
$\widetilde{\boldsymbol{\pi}}(\boldsymbol{s}_t)$	Policy to choose load control action vector $\widetilde{\boldsymbol{P}}^{\mathrm{c}}(\boldsymbol{s}_t)$ in state $\boldsymbol{s}_t$
$\widetilde{\boldsymbol{\pi}}$	Policy profile, i.e., $\widetilde{\boldsymbol{\pi}} = (\widetilde{\boldsymbol{\pi}}(\boldsymbol{s}),\ \boldsymbol{s} \in \mathcal{S})$
$\alpha_n(\boldsymbol{s}_t)$	Auxiliary variable to replace $(P^{\mathrm{c}}_n(\boldsymbol{s}_t) - \widetilde{P}^{\mathrm{c}}_n(\boldsymbol{s}_t))^2$
$\boldsymbol{\vartheta}$	Parameters vector of the DNN for value function
$\boldsymbol{\theta}$	Parameters vector of the DNN for policy
$V^{\widetilde{\boldsymbol{\pi}}}(\boldsymbol{s}, \boldsymbol{\vartheta})$	Parameterized value function for policy $\widetilde{\boldsymbol{\pi}}$ and initial state $\boldsymbol{s}$
$\mathcal{J}$	Set of nodes in the first hidden layer of the global DNN for the value function
$\lambda_{j,n,t}$	Aggregate input from the nodes of state $\boldsymbol{s}_{n,t}$ to node $j \in \mathcal{J}$ in the DNN for the value function
$\overline{\lambda}_{j,n,t}$	Sum of parameters $\lambda_{j,n',t}$ for $n' \in \mathcal{N}^- \setminus \{n\}$
$\boldsymbol{\lambda}_{n,t}$	Profile of parameters $\lambda_{j,n,t}$ for $j \in \mathcal{J}$
$\overline{\boldsymbol{\lambda}}_{n,t}$	Profile of parameters $\overline{\lambda}_{j,n,t}$ for $j \in \mathcal{J}$
$\delta_n(\cdot)$	TD error for household n

I. Introduction

Balancing electricity generation and demand during peak time periods is an important issue in distribution networks. A demand response program with real-time pricing scheme can encourage residential users toward reducing electricity consumption during peak hours, which eliminates the need for backup power plants and potential electricity supply interruption. Smart meters and electricity consumption controllers (ECCs) can use the energy consumption and pricing information to provide users with autonomous load scheduling plans that take advantage of the potential cost savings offered by a demand response program with real-time pricing scheme [1].

Despite the aforementioned benefits, there are major challenges in deploying a demand response program for residential users. First, as a coordinator for users' load scheduling decisions, the load aggregator has uncertainty in the electricity price and demand variation of users. Second, due to privacy concerns, the load aggregator generally lacks information

Manuscript was received on Aug. 8, 2020, revised on Oct. 20, 2020, and accepted on Oct. 30, 2020. This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Strategic Project Grant (STPGP 447607-13). S. Bahrami, Y.C. Chen, and V.W.S. Wong are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, V6T 1Z4, email: {bahramis, chen, vincentw}@ece.ubc.ca

about the users' discomfort due to curtailing their load, which may lead to an uncertain amount of aggregate demand reduction during peak hours. Furthermore, the load aggregator should coordinate the users' load changes to satisfy operational constraints imposed by the distribution network. Otherwise, the users' load scheduling may overload the transmission lines or lead to an undesirable voltage deviations.

There are several studies in the literature for demand response programs that tackle uncertainty in price and users' demand. Related work falls within three main threads. The first line of research pertains to offline algorithms for demand response programs in day-ahead electricity markets using bidding mechanism design [2], [3], forecasting techniques [4]–[6], and scenario-based approaches [7], [8]. Offline algorithms, however, require predictive models with historical data for uncertain parameters, which may not always be available. The second line of research comprises online algorithms for demand response using model-based methods such as dynamic programming [9], randomized alternating direction method of multipliers (ADMM) [10], chance-constrained optimization [11], robust optimization [12], and stochastic programming [13]. The model-based methods, however, require knowledge of the stochastic process of the uncertain parameters, which may not be available in practice.

The third line of research is related to applying model-free approaches to design online algorithms for demand response. Zheng *et al.* [14] and Elghitan *et al.* [15] applied Lyapunov optimization to design online load control algorithms for users with thermostatically controlled loads. Lyapunov optimization, however, is limited to the loads that can be modeled as a queuing system, which may not be applicable for the loads other than thermostatically controlled ones. Kim *et al.* in [16] and Lesage-Landry *et al.* in [17] applied online convex optimization technique to develop online load control algorithms with bounded regret for the users in a demand response program. However, the application of online convex optimization is limited, since it cannot directly handle inter-temporal constraints. Recently, there is growing interest in applying deep reinforcement learning (DRL) [18] to design demand response algorithms for residential users. Wang *et al.* [19] developed a DRL-based load scheduling algorithm using a dueling deep Q-network model with a time-of-use pricing scheme. Li *et al.* [20] modeled the charging coordination of electric vehicles in a demand response program as a Markov decision process (MDP) and applied DRL to design an online charging scheduling algorithm. Du *et al.* [21] designed a pricing mechanism for multiple microgrids and applied DRL to capture the response of the microgrids to the price rates in a demand response program. Liu *et al.* [22] applied double deep Q-learning approach to design an online household energy management algorithm. Wan *et al.* [23] considered a long short-term memory (LSTM) network to extract informative features from the price rates and developed a DRL-based online charging scheduling algorithm for electric vehicles. Li *et al.* [24] used trust region policy optimization technique to develop a DRL-based algorithm for household appliances scheduling in a demand response program with real-time pricing scheme. Yu *et al.* [25] designed a DRL-based demand response algorithm for scheduling the demand of the energy storage system and thermostatically controlled loads in a household. Mocanu *et al.* [26] applied policy gradient evaluation to design a DRL-based online load scheduling algorithm for peak load reduction in residential households. Claessens *et al.* [27] used convolutional neural networks to extract a set of state-time features for residential households to be used in a DRL-based load scheduling algorithm. Ruelens *et al.* [28] applied batch reinforcement learning to coordinate the power consumption of users with thermostatically controlled loads. Babar *et al.* [29] proposed a decentralized Q-learning algorithm for bidding of multiple residential households in a demand response program. Lu *et al.* [30] proposed a DRL-based home energy management algorithm combined with a neural network for price forecasting. Although the aforementioned papers do not require stochastic models for uncertain parameters, they do not consider the constraints imposed by the distribution network topology and operations. Hence, the developed algorithms may not yield a feasible power flow in the distribution network. Furthermore, the decision making processes of individual households become coupled if the power flow constraints are taken into account. Preserving privacy becomes a concern for the households with coupling constraints, which is not addressed in prior art.

In this paper, we model the users' load control problem as an MDP and apply DRL to develop a demand response algorithm in a distribution network. We take into account the uncertainty in the electricity price, load demand, and users' discomfort cost. This paper extends our previous work [31] by designing a decentralized learning framework, such that the users make load control decisions in parallel without revealing their private information (e.g., load demand, discomfort cost) to the load aggregator. The learning process accounts for distribution network constraints to guarantee a feasible power flow solution. Specific contributions of this paper are as follows.

- *Decentralized Learning Algorithm Design:* The centralized load control may violate users' privacy, as users must provide the load aggregator with information about their desirable load demand and discomfort cost. To address the households' privacy concerns, we apply federated learning [32], [33], which enables us to develop a decentralized DRL algorithm [18] with actor-critic method [34] and update the parameters of the local neural networks associated with each household in parallel. Instead of revealing private information, the households share only their local network parameters with the load aggregator to update the neural network parameters associated with the policy and value function.
- *Distribution Network Constraints:* The load control decisions of the households are coupled by the ac power flow constraints imposed by the distribution network. It is difficult to obtain a policy that satisfies power flow constraints. We decouple the tasks of scheduling the load demand and obtaining a feasible power flow in the distribution network by projecting the selected load control action of the households onto the feasible set defined by the ac power flow constraints. To tackle the nonconvex power flow constraints, we apply convex

relaxation to transform the action projection problem into a sequence of semidefinite programs (SDPs). Solving the obtained SDP yields the global optimal solution to the action projection problem under the given policy and distribution network constraints.

- *Performance Evaluation:* We evaluate the performance of the proposed DRL algorithm in an IEEE 33-bus test feeder with 32 households. The proposed decentralized algorithm converges to the solution of the centralized load control in an acceptable number of iterations. When compared with Q-learning and double Q-learning approaches, our proposed algorithm based on actor-critic method converges faster to a local optimum with lower user cost. We demonstrate the scalability of our proposed algorithm in 330-bus and 1650-bus test feeders with real-time pricing scheme. Additionally, our case study demonstrates that the proposed algorithm leads to 33% reduction in the peak load and 13% reduction in the households' expected daily cost.

The remainder of this paper is organized as follows. Section II introduces the operational constraints for the residential households and distribution network. In Section III, we formulate the centralized demand response problem as an MDP. In Section IV, we propose a decentralized DRL algorithm to solve the underlying MDP. In Section V, we evaluate the performance of the proposed algorithm through simulations. Section VI concludes the paper.

II. System Model

Consider a distribution feeder consisting of N buses collected in set $\mathcal{N} = \{1, \ldots, N\}$. Suppose that bus N corresponds to the substation bus and bus $n \in \mathcal{N}^-$ corresponds to household n, where $\mathcal{N}^- = \{1, \ldots, N-1\}$. Without loss of generality, we assume a virtual household with zero load demand is connected to bus $n \in \mathcal{N}^-$ if no household is connected to it. Let $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ denote the set of transmission lines in the feeder. Each household is equipped with an ECC, which is responsible for load control in that household. The ECCs are connected to a load aggregator via a two-way communication network. We consider long-term load control (e.g., several weeks) and approximate the load control problem with an infinite operation horizon. We consider a discrete set $\mathcal{T} = \{1,\, 2,\, \ldots\}$ containing indices corresponding to time slots, each with equal duration, e.g., 15 minutes per slot.

A. Household State and Load Control Action

The active power demand for a household consists of a controllable portion and an uncontrollable base portion. Let $P^{\text{b}}_{n,t}$ denote the base load demand in household $n \in \mathcal{N}^-$ in time slot $t \in \mathcal{T}$. For load scheduling, ECC n considers the lower bound $P^{\text{c,min}}_{n,t}$ and the desirable value $P^{\text{c,des}}_{n,t}$ for the controllable load demand in time slot t. In practice, parameters $P^{\text{c,min}}_{n,t}$ and $P^{\text{c,des}}_{n,t}$ can be obtained from the operating modes of the household appliances [35]. We assume that ECC n observes the base load $P^{\text{b}}_{n,t}$ as well as parameters $P^{\text{c,min}}_{n,t}$ and $P^{\text{c,des}}_{n,t}$ at the beginning of time slot t, *just before* scheduling the controllable loads. ECC n is uncertain about $P^{\text{b}}_{n,t'}$, $P^{\text{c,min}}_{n,t'}$, and $P^{\text{c,des}}_{n,t'}$ for upcoming time slot $t' \geq t+1$.

We define the *state* of household $n \in \mathcal{N}^-$ in time slot $t \in \mathcal{T}$ as vector $\boldsymbol{s}_{n,t} = (P^{\text{b}}_{n,t},\, P^{\text{c,min}}_{n,t},\, P^{\text{c,des}}_{n,t})$. Let ρ_t denote the electricity price in time slot t. We define the system state in time slot t as vector $\boldsymbol{s}_t = (\boldsymbol{s}_{n,t},\, n \in \mathcal{N}^-,\, \rho_t)$, which includes the state of all households and the electricity price in time slot t. We use $\mathcal{S}$ to denote the set of system states.

Given state $\boldsymbol{s}_t \in \mathcal{S}$ in time slot $t \in \mathcal{T}$, the *load control action* for household $n \in \mathcal{N}^-$ is defined as the scheduled controllable load demand $P^{\text{c}}_n(\boldsymbol{s}_t)$. We consider curtailing the load in the households. Thus, in state $\boldsymbol{s}_t$, we have

$$P^{\text{c,min}}_{n,t} \leq P^{\text{c}}_n(\boldsymbol{s}_t) \leq P^{\text{c,des}}_{n,t}, \qquad n \in \mathcal{N}^-,\, t \in \mathcal{T}. \tag{1}$$

Finally, household appliances may require reactive power. We consider the overall power factor $\phi_{n,t} \in [-1, 1] \setminus \{0\}$ for household n in time slot t, which is assumed to be known *a priori* by ECC n. The reactive power demand of household n in state $\boldsymbol{s}_t$ and time slot t is obtained as $\kappa_{n,t}(P^{\text{c}}_n(\boldsymbol{s}_t) + P^{\text{b}}_{n,t})$, where $\kappa_{n,t} = \sqrt{1/\phi^2_{n,t} - 1}$ for lagging power factor and $\kappa_{n,t} = -\sqrt{1/\phi^2_{n,t} - 1}$ for leading power factor.

B. Power Flow Constraints

Let Y denote the distribution network admittance matrix. For bus $n \in \mathcal{N}$, let $e_n \in \mathbb{R}^N$ denote the n^{th} basis column vector and $Y_n = e_n e_n^{\text{T}} Y$. Row n of matrix Y_n is equal to row n of the admittance matrix Y, and other entries of Y_n are zero. We use the lumped-element Π model for transmission lines. Let y_{nm} and $\overline{y}_{nm}$, respectively, denote the series and shunt admittances connected to bus n for line $(n,m) \in \mathcal{L}$. We define $Y_{nm} = (\overline{y}_{nm} + y_{nm}) e_n e_n^{\text{T}} - y_{nm} e_n e_m^{\text{T}}$, so that the entries (n,n) and (n,m) of Y_{nm} are $\overline{y}_{nm} + y_{nm}$ and $-y_{nm}$, respectively, and all other entries of Y_{nm} are zero. For bus $n \in \mathcal{N}$, we define matrices $\mathbf{Y}_n$, $\overline{\mathbf{Y}}_n$, and $\mathbf{M}_n$ as follows:

$$\mathbf{Y}_n = \frac{1}{2}\begin{bmatrix} \text{Re}\{Y_n + Y_n^{\text{T}}\} & \text{Im}\{Y_n^{\text{T}} - Y_n\} \\ \text{Im}\{Y_n - Y_n^{\text{T}}\} & \text{Re}\{Y_n + Y_n^{\text{T}}\} \end{bmatrix}, \tag{2a}$$

$$\overline{\mathbf{Y}}_n = -\frac{1}{2}\begin{bmatrix} \text{Im}\{Y_n + Y_n^{\text{T}}\} & \text{Re}\{Y_n - Y_n^{\text{T}}\} \\ \text{Re}\{Y_n^{\text{T}} - Y_n\} & \text{Im}\{Y_n + Y_n^{\text{T}}\} \end{bmatrix}, \tag{2b}$$

$$\mathbf{M}_n = \begin{bmatrix} e_n e_n^{\text{T}} & 0 \\ 0 & e_n e_n^{\text{T}} \end{bmatrix}. \tag{2c}$$

For each line $(n,m) \in \mathcal{L}$, we define matrices:

$$\mathbf{Y}_{nm} = \frac{1}{2}\begin{bmatrix} \text{Re}\{Y_{nm} + Y_{nm}^{\text{T}}\} & \text{Im}\{Y_{nm}^{\text{T}} - Y_{nm}\} \\ \text{Im}\{Y_{nm} - Y_{nm}^{\text{T}}\} & \text{Re}\{Y_{nm} + Y_{nm}^{\text{T}}\} \end{bmatrix}, \tag{3a}$$

$$\overline{\mathbf{Y}}_{nm} = -\frac{1}{2}\begin{bmatrix} \text{Im}\{Y_{nm} + Y_{nm}^{\text{T}}\} & \text{Re}\{Y_{nm} - Y_{nm}^{\text{T}}\} \\ \text{Re}\{Y_{nm}^{\text{T}} - Y_{nm}\} & \text{Im}\{Y_{nm} + Y_{nm}^{\text{T}}\} \end{bmatrix}. \tag{3b}$$

The sinusoidal steady-state voltage of bus n can be expressed as a phasor quantity [36, Sec. 2.1]. We use $V_n(\boldsymbol{s}_t)$ to denote the voltage phasor of bus n in state $\boldsymbol{s}_t$. Let $\mathbf{v}(\boldsymbol{s}_t) = (V_n(\boldsymbol{s}_t),\, n \in \mathcal{N})$ denote the vector of voltage phasors in state $\boldsymbol{s}_t$. Also, let $\mathcal{N}_n \subseteq \mathcal{N}$ denote the set of buses electrically connected to bus n. We construct vector $\mathbf{v}_n(\boldsymbol{s}_t)$ for bus n

from vector $\mathbf{v}(\boldsymbol{s}_t)$ such that the entries $m \in \mathcal{N}_n \cup \{n\}$ of vectors $\mathbf{v}_n(\boldsymbol{s}_t)$ and $\mathbf{v}(\boldsymbol{s}_t)$ are equal, and other entries in vector $\mathbf{v}_n(\boldsymbol{s}_t)$ are set to zero. For bus $n \in \mathcal{N}$, we define vector $\mathbf{u}_n(\boldsymbol{s}_t) = (\,(\text{Re}\{\mathbf{v}_n(\boldsymbol{s}_t)\})^\text{T}\ (\text{Im}\{\mathbf{v}_n(\boldsymbol{s}_t)\})^\text{T}\,)^\text{T}$ consisting of the real and imaginary parts of $\mathbf{v}_n(\boldsymbol{s}_t)$ in state $\boldsymbol{s}_t$. For $n \in \mathcal{N}$, we define matrix $\mathbf{W}_n(\boldsymbol{s}_t) = \mathbf{u}_n(\boldsymbol{s}_t)\,\mathbf{u}_n(\boldsymbol{s}_t)^\text{T}$ in state $\boldsymbol{s}_t$. Since matrix $\mathbf{W}_n(\boldsymbol{s}_t)$ is the outer product of vectors $\mathbf{u}_n(\boldsymbol{s}_t)$ and $\mathbf{u}_n(\boldsymbol{s}_t)^\text{T}$, it is rank-one by construction.

Let P_N^max denote the upper limit for the injected active power into the substation bus N. Let Q_N^min and Q_N^max, respectively, denote the lower and upper limits for the injected reactive power into the substation bus N. We denote the lower and upper limits of the voltage magnitude at bus $n \in \mathcal{N}$ by V_n^min and V_n^max, respectively. Let S_{nm}^max denote the upper limit for the apparent power flow in line $(n, m) \in \mathcal{L}$. Parameters P_N^max, Q_N^min, Q_N^max, V_n^min, V_n^max, $n \in \mathcal{N}$, and $S_{n,m}^\text{max}$, $(n, m) \in \mathcal{L}$, are constant and are chosen according to the design and operation of the underlying power distribution network. Let $\mathbf{W}_n^{k,k'}(\boldsymbol{s}_t)$ denote the entry $(k,\, k')$ of matrix $\mathbf{W}_n(\boldsymbol{s}_t)$. We leverage the matrices defined in (2a)$-$(2c), (3a), and (3b) to obtain the following distribution network constraints in time slot $t \in \mathcal{T}$ and state $\boldsymbol{s}_t \in \mathcal{S}$ [37]:

$$P_n^\text{c}(\boldsymbol{s}_t) + P_{n,t}^\text{b} = -\text{Tr}\{\mathbf{Y}_n \mathbf{W}_n(\boldsymbol{s}_t)\}, \qquad n \in \mathcal{N}^-, \tag{4a}$$

$$\kappa_{n,t}(P_n^\text{c}(\boldsymbol{s}_t) + P_{n,t}^\text{b}) = -\text{Tr}\{\overline{\mathbf{Y}}_n \mathbf{W}_n(\boldsymbol{s}_t)\}, \qquad n \in \mathcal{N}^-, \tag{4b}$$

$$0 \le \text{Tr}\{\mathbf{Y}_N \mathbf{W}_N(\boldsymbol{s}_t)\} \le P_N^\text{max}, \tag{4c}$$

$$Q_N^\text{min} \le \text{Tr}\{\overline{\mathbf{Y}}_N \mathbf{W}_N(\boldsymbol{s}_t)\} \le Q_N^\text{max}, \tag{4d}$$

$$(V_n^\text{min})^2 \le \text{Tr}\{\mathbf{M}_n \mathbf{W}_n(\boldsymbol{s}_t)\} \le (V_n^\text{max})^2, \qquad n \in \mathcal{N}, \tag{4e}$$

$$\begin{bmatrix} (S_{nm}^\text{max})^2 & \text{Tr}\{\mathbf{Y}_{nm}\mathbf{W}_n(\boldsymbol{s}_t)\} & \text{Tr}\{\overline{\mathbf{Y}}_{nm}\mathbf{W}_n(\boldsymbol{s}_t)\} \\ \text{Tr}\{\mathbf{Y}_{nm}\mathbf{W}_n(\boldsymbol{s}_t)\} & 1 & 0 \\ \text{Tr}\{\overline{\mathbf{Y}}_{nm}\mathbf{W}_n(\boldsymbol{s}_t)\} & 0 & 1 \end{bmatrix} \succeq 0,$$

$$n \in \mathcal{N},\ (n, m) \in \mathcal{L}, \tag{4f}$$

$$\mathbf{W}_n^{n,n}(\boldsymbol{s}_t) = \mathbf{W}_m^{n,n}(\boldsymbol{s}_t), \qquad m \in \mathcal{N}_n,\ n \in \mathcal{N}, \tag{4g}$$

$$\mathbf{W}_n^{n+N,n+N}(\boldsymbol{s}_t) = \mathbf{W}_m^{n+N,n+N}(\boldsymbol{s}_t),\ m \in \mathcal{N}_n,\ n \in \mathcal{N}, \tag{4h}$$

$$\text{rank}\{\mathbf{W}_n(\boldsymbol{s}_t)\} = 1, \qquad n \in \mathcal{N}. \tag{4i}$$

Constraints (4a) and (4b) represent power balance at bus n. Constraints (4c) and (4d) represent the limits on the injected active and reactive power into the substation bus, respectively. Constraint (4e) shows the limits on the voltage magnitude of bus n. Constraint (4f) represents the limit on the apparent power flow in line (n, m). Constraints (4g) and (4h) establish that for two connected buses n and m, the diagonal entries $(n,\, n)$ and $(n+N,\, n+N)$ of matrices $\mathbf{W}_n(\boldsymbol{s}_t)$ and $\mathbf{W}_m(\boldsymbol{s}_t)$ are equal, as they represent the square of real and imaginary parts of voltage phasor $V_n(\boldsymbol{s}_t)$ at bus n in state $\boldsymbol{s}_t$, respectively. Constraint (4i) ensures that $\mathbf{W}_n(\boldsymbol{s}_t)$ is a rank-one matrix.

III. Problem Formulation

In this section, we formulate the *centralized* load control problem as an MDP with an infinite operation horizon, where the system state in the next time slot $t+1$ can be inferred from the state and action in the current time slot t [35].

A. Centralized Load Control MDP

In the centralized load control MDP, the system state $\boldsymbol{s}_t$ is observed and the controllable load demand $P_n^\text{c}(\boldsymbol{s}_t)$ for household $n \in \mathcal{N}^-$ is determined as the load control action in time slot $t \in \mathcal{T}$. The power flow constraints (4a)$-$(4i) depend on matrix $\mathbf{W}_n(\boldsymbol{s}_t)$, $n \in \mathcal{N}$, in state $\boldsymbol{s}_t$. Matrix $\mathbf{W}_n(\boldsymbol{s}_t)$ is a decision variable for bus $n \in \mathcal{N}$. Thus, we define the action associated with household $n \in \mathcal{N}^-$ in state $\boldsymbol{s}_t \in \mathcal{S}$ as $\boldsymbol{a}_n(\boldsymbol{s}_t) = (P_n^\text{c}(\boldsymbol{s}_t),\, \mathbf{W}_n(\boldsymbol{s}_t))$ to include both the load control action $P_n^\text{c}(\boldsymbol{s}_t)$ and decision variable $\mathbf{W}_n(\boldsymbol{s}_t)$. We also define the action associated with the substation bus N in state $\boldsymbol{s}_t \in \mathcal{S}$ as $\boldsymbol{a}_N(\boldsymbol{s}_t) = \mathbf{W}_N(\boldsymbol{s}_t)$. Let $\boldsymbol{a}(\boldsymbol{s}_t) = (\boldsymbol{a}_n(\boldsymbol{s}_t),\, n \in \mathcal{N})$ denote the joint action profile in state $\boldsymbol{s}_t$. The feasible action space $\mathcal{A}(\boldsymbol{s}_t)$ is defined by constraints (1) and (4a)$-$(4i). ECC n can only control the load demand $P_n^\text{c}(\boldsymbol{s}_t)$, whereas matrix $\mathbf{W}_n(\boldsymbol{s}_t)$ is determined so that the power flow constraints are satisfied.

By performing load scheduling in time slot t, household n incurs a discomfort cost $d_n(P_n^\text{c}(\boldsymbol{s}_t),\, P_{n,t}^\text{c,des})$, which models the user's dissatisfaction with changing its controllable load demand from the desirable value $P_{n,t}^\text{c,des}$ to the scheduled value $P_n^\text{c}(\boldsymbol{s}_t)$. The discomfort cost captures the user's flexibility in scheduling the controllable load demand in the household.

Assumption 1: Although the closed-form expression for the user's discomfort cost is unknown to ECC n, the value of the discomfort cost is revealed to ECC n at the end of time slot t after scheduling the household's controllable loads.

Let $c_n(\boldsymbol{s}_t,\, \boldsymbol{a}(\boldsymbol{s}_t)) = \rho_t(P_n^\text{c}(\boldsymbol{s}_t) + P_{n,t}^\text{b}) + d_n(P_n^\text{c}(\boldsymbol{s}_t),\, P_{n,t}^\text{c,des})$ denote the cost (i.e., the bill payment and discomfort cost) of household n in state $\boldsymbol{s}_t$. We consider the *social* cost as the immediate cost in state $\boldsymbol{s}_t$. We have

$$c(\boldsymbol{s}_t,\, \boldsymbol{a}(\boldsymbol{s}_t)) = \sum_{n \in \mathcal{N}^-} c_n(\boldsymbol{s}_t,\, \boldsymbol{a}(\boldsymbol{s}_t)), \ \ \boldsymbol{s}_t \in \mathcal{S}. \tag{5}$$

We consider a stationary randomized policy as a mapping from states to probabilities of selecting a feasible action. Given state $\boldsymbol{s}_t = \boldsymbol{s}$ for any $\boldsymbol{s} \in \mathcal{S}$, the policy is defined as a probability distribution $\boldsymbol{\pi}(\boldsymbol{s}) = (\pi(\boldsymbol{a}(\boldsymbol{s}) \,|\, \boldsymbol{s}),\, \boldsymbol{a}(\boldsymbol{s}) \in \mathcal{A}(\boldsymbol{s}))$ that specifies the probability $\pi(\boldsymbol{a}(\boldsymbol{s}) \,|\, \boldsymbol{s})$ of choosing a feasible action $\boldsymbol{a}(\boldsymbol{s})$ in state $\boldsymbol{s}$. A policy is defined as $\boldsymbol{\pi} = (\boldsymbol{\pi}(\boldsymbol{s}),\, \boldsymbol{s} \in \mathcal{S})$. For a given policy $\boldsymbol{\pi}$, the value function $V^{\boldsymbol{\pi}} : \mathcal{S} \to \mathbb{R}$ in state $\boldsymbol{s}$ returns the expected discounted cost when starting from state $\boldsymbol{s}_t = \boldsymbol{s}$ in time slot t and following policy $\boldsymbol{\pi}$ in the upcoming time slots. For a discount factor $\beta \in [0,\, 1)$, we have

$$V^{\boldsymbol{\pi}}(\boldsymbol{s}) = \mathbb{E}^{\boldsymbol{\pi}} \left\{ \sum_{t'=t}^{\infty} \beta^{t'-t}\, c(\boldsymbol{s}_{t'},\, \boldsymbol{a}(\boldsymbol{s}_{t'})) \,\Big|\, \boldsymbol{s}_t = \boldsymbol{s} \right\}, \tag{6}$$

where $\mathbb{E}^{\boldsymbol{\pi}}\{\cdot\}$ is the expectation over selecting feasible actions under the given policy $\boldsymbol{\pi}$. Under the given policy $\boldsymbol{\pi}$, we define the action-value function $Q^{\boldsymbol{\pi}}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}))$ in state $\boldsymbol{s}$ and action $\boldsymbol{a}(\boldsymbol{s})$ as the expected discounted cost when starting from state $\boldsymbol{s}_t = \boldsymbol{s}$ in a given time slot t, performing action $\boldsymbol{a}(\boldsymbol{s})$, and following policy $\boldsymbol{\pi}$ in the upcoming time slots. We have

$$Q^{\boldsymbol{\pi}}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s})) =$$

$$\mathbb{E}^{\boldsymbol{\pi}} \left\{ \sum_{t'=t}^{\infty} \beta^{t'-t}\, c(\boldsymbol{s}_{t'},\, \boldsymbol{a}(\boldsymbol{s}_{t'})) \,\Big|\, \boldsymbol{s}_t = \boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}_t) = \boldsymbol{a}(\boldsymbol{s}) \right\}. \tag{7}$$

Let $\text{Pr}(\boldsymbol{s}' \,|\, \boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}))$ denote the transition probability from state $\boldsymbol{s}$ to $\boldsymbol{s}'$ with action $\boldsymbol{a}(\boldsymbol{s})$. We can express the action-value function for state $\boldsymbol{s} \in \mathcal{S}$ and feasible action $\boldsymbol{a}(\boldsymbol{s})$ in terms of

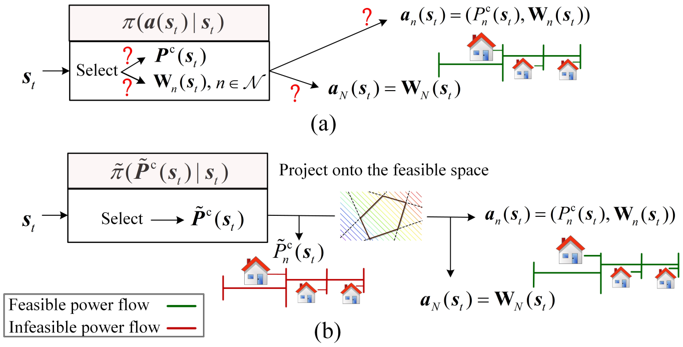

$\pi(\boldsymbol{a}(\boldsymbol{s}_t)\,|\,\boldsymbol{s}_t)$
$\boldsymbol{s}_t$
Select
?
$\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t)$
$\mathbf{W}_n(\boldsymbol{s}_t),\ n \in \mathcal{N}$
$\boldsymbol{a}_n(\boldsymbol{s}_t) = (P_n^{\rm c}(\boldsymbol{s}_t), \mathbf{W}_n(\boldsymbol{s}_t))$
$\boldsymbol{a}_N(\boldsymbol{s}_t) = \mathbf{W}_N(\boldsymbol{s}_t)$
(a)
$\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)\,|\,\boldsymbol{s}_t)$
Project onto the feasible space
$\boldsymbol{s}_t$
Select
$\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)$
$\widetilde{P}_n^{\rm c}(\boldsymbol{s}_t)$
$\boldsymbol{a}_n(\boldsymbol{s}_t) = (P_n^{\rm c}(\boldsymbol{s}_t), \mathbf{W}_n(\boldsymbol{s}_t))$
$\boldsymbol{a}_N(\boldsymbol{s}_t) = \mathbf{W}_N(\boldsymbol{s}_t)$
Feasible power flow
Infeasible power flow
(b)

Figure 1. Select action $\boldsymbol{a}(\boldsymbol{s}_t)$ (a) with policy $\boldsymbol{\pi}$; (b) by projecting $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)$ onto the feasible space with policy $\widetilde{\boldsymbol{\pi}}$.

the value function in the next state $\boldsymbol{s}' \in \mathcal{S}$ as follows:

$$Q^{\boldsymbol{\pi}}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s})) = c(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s})) + \beta \sum_{\boldsymbol{s}' \in \mathcal{S}} \Pr(\boldsymbol{s}' \,|\, \boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}))\, V^{\boldsymbol{\pi}}(\boldsymbol{s}'). \tag{8}$$

The load aggregator aims to obtain policy $\boldsymbol{\pi}$ such that the value function is minimized over all states $\boldsymbol{s} \in \mathcal{S}$. This is equivalent to solving the following Bellman equations:

$$\boldsymbol{\mathcal{P}}^{\rm MDP}: \quad V^{\boldsymbol{\pi}}(\boldsymbol{s}) = \underset{\boldsymbol{a}(\boldsymbol{s}) \in \mathcal{A}(\boldsymbol{s})}{\text{minimize}}\ \mathbb{E}\{Q^{\boldsymbol{\pi}}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}))\}, \quad \forall\, \boldsymbol{s} \in \mathcal{S}.$$

Problem $\boldsymbol{\mathcal{P}}^{\rm MDP}$ is a recursive optimization problem. In general, the solution policy $\boldsymbol{\pi}$ to problem $\boldsymbol{\mathcal{P}}^{\rm MDP}$ is *suboptimal*. Also, Fig. 1(a) shows that the rank-one constraint (4i) makes it difficult to determine policy $\boldsymbol{\pi}$ as a probability of choosing a feasible action vector $\boldsymbol{a}(\boldsymbol{s}_t)$. Hence, determining a policy $\boldsymbol{\pi}$ for $\boldsymbol{\mathcal{P}}^{\rm MDP}$ may lead to an infeasible action that does not satisfy the power flow constraints, thereby putting the operation of the power system at risk. To address this challenge, we decouple the tasks of scheduling the households' loads and obtaining a feasible power flow in the distribution network. As Fig. 1(b) illustrates, to obtain a suboptimal solution to problem $\boldsymbol{\mathcal{P}}^{\rm MDP}$, we consider a policy $\widetilde{\boldsymbol{\pi}} = (\widetilde{\boldsymbol{\pi}}(\boldsymbol{s}),\, \boldsymbol{s} \in \mathcal{S})$ that determines vector $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t) = (\widetilde{P}_n^{\rm c}(\boldsymbol{s}_t),\, n \in \mathcal{N}^-)$ of load control action $\widetilde{P}_n^{\rm c}(\boldsymbol{s}_t)$ for household n in feasible space $\mathcal{A}^{\rm c}(\boldsymbol{s}_t)$ defined by constraint (1) in state $\boldsymbol{s}_t$. Then, $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)$ is projected onto the feasible action space $\mathcal{A}(\boldsymbol{s}_t)$ to obtain action vector $\boldsymbol{a}(\boldsymbol{s}_t)$ that satisfies the power flow constraints.

The policy $\widetilde{\boldsymbol{\pi}}$ in state $\boldsymbol{s}_t = \boldsymbol{s}$ for any $\boldsymbol{s} \in \mathcal{S}$ specifies a probability distribution $\widetilde{\boldsymbol{\pi}}(\boldsymbol{s}) = (\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}) \,|\, \boldsymbol{s}),\, \widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}) \in \mathcal{A}^{\rm c}(\boldsymbol{s}))$ that includes the probability $\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}) \,|\, \boldsymbol{s})$ of choosing load control vector $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s})$ in feasible space $\mathcal{A}^{\rm c}(\boldsymbol{s})$. With $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)$, there may not exist matrices $\mathbf{W}_n(\boldsymbol{s}_t)$, $n \in \mathcal{N}$, that satisfy power flow constraints (4a)$-$(4i). We project $\widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)$ onto the feasible action space to obtain a new action profile $\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t) = (P_n^{\rm c}(\boldsymbol{s}_t),\, n \in \mathcal{N}^-)$, for which there exist rank-one matrices $\mathbf{W}_n(\boldsymbol{s}_t)$, $n \in \mathcal{N}$, that satisfy constraints (4a)$-$(4i). The load aggregator solves the following optimization problem:

$$\begin{aligned} \boldsymbol{\mathcal{P}}_1^{\rm proj}: \quad & \underset{\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t),\, \mathbf{W}_n(\boldsymbol{s}_t),\, n \in \mathcal{N}}{\text{minimize}} \quad ||\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t) - \widetilde{\boldsymbol{P}}^{\rm c}(\boldsymbol{s}_t)||_2^2 \\ & \text{subject to constraints (4a)$-$(4i)}, \\ & \qquad \boldsymbol{P}^{\rm c}(\boldsymbol{s}_t) \in \mathcal{A}^{\rm c}(\boldsymbol{s}_t). \end{aligned}$$

Problem $\boldsymbol{\mathcal{P}}_1^{\rm proj}$ is a nonconvex optimization problem due to the rank-one constraint (4i). We relax constraint (4i) and replace it with constraint $\mathbf{W}_n(\boldsymbol{s}_t) \succeq 0$, $n \in \mathcal{N}$, that enforces matrices $\mathbf{W}_n(\boldsymbol{s}_t)$, $n \in \mathcal{N}$, to be positive semidefinite. Furthermore, we define an auxiliary variable $\alpha_n(\boldsymbol{s}_t)$ for $n \in \mathcal{N}^-$, such that the inequality constraint $(P_n^{\rm c}(\boldsymbol{s}_t) - \widetilde{P}_n^{\rm c}(\boldsymbol{s}_t))^2 \leq \alpha_n(\boldsymbol{s}_t)$ can be expressed as the following linear matrix inequality:

$$\begin{bmatrix} \alpha_n(\boldsymbol{s}_t) & P_n^{\rm c}(\boldsymbol{s}_t) - \widetilde{P}_n^{\rm c}(\boldsymbol{s}_t) \\ P_n^{\rm c}(\boldsymbol{s}_t) - \widetilde{P}_n^{\rm c}(\boldsymbol{s}_t) & 1 \end{bmatrix} \succeq 0, \ \ n \in \mathcal{N}^-. \tag{9}$$

We replace the objective function with $\sum_{n \in \mathcal{N}^-} \alpha_n(\boldsymbol{s}_t)$ to transform $\boldsymbol{\mathcal{P}}_1^{\rm proj}$ into the following optimization problem:

$$\begin{aligned} \boldsymbol{\mathcal{P}}_2^{\rm proj}: \quad & \underset{\substack{\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t),\, \alpha_n(\boldsymbol{s}_t),\, n \in \mathcal{N}^-, \\ \mathbf{W}_n(\boldsymbol{s}_t),\, n \in \mathcal{N}}}{\text{minimize}} \quad \sum_{n \in \mathcal{N}^-} \alpha_n(\boldsymbol{s}_t) \\ & \text{subject to constraints (4a)$-$(4h) and (9)}, \\ & \qquad \mathbf{W}_n(\boldsymbol{s}_t) \succeq 0, \qquad n \in \mathcal{N}, \\ & \qquad \boldsymbol{P}^{\rm c}(\boldsymbol{s}_t) \in \mathcal{A}^{\rm c}(\boldsymbol{s}_t). \end{aligned}$$

Problem $\boldsymbol{\mathcal{P}}_2^{\rm proj}$ is an SDP and can be solved efficiently to obtain joint action $\boldsymbol{a}(\boldsymbol{s}_t) = (\boldsymbol{P}^{\rm c}(\boldsymbol{s}_t),\, \mathbf{W}_n(\boldsymbol{s}_t),\, n \in \mathcal{N})$, which is feasible for the distribution network. We show that the optimal solution to $\boldsymbol{\mathcal{P}}_2^{\rm proj}$ is the global optimal solution to $\boldsymbol{\mathcal{P}}_1^{\rm proj}$.

Theorem 1: *The relaxation gap between problems $\boldsymbol{\mathcal{P}}_1^{\rm proj}$ and $\boldsymbol{\mathcal{P}}_2^{\rm proj}$ is zero. That is, the solution matrices $\mathbf{W}_n(\boldsymbol{s}_t)$, $n \in \mathcal{N}$ to $\boldsymbol{\mathcal{P}}_2^{\rm proj}$ are rank-one.*

The proof can be found in Appendix A. Theorem 1 implies that a feasible action $\boldsymbol{a}(\boldsymbol{s}_t)$ can be obtained for the given policy $\widetilde{\boldsymbol{\pi}}$ by solving the convex optimization problem $\boldsymbol{\mathcal{P}}_2^{\rm proj}$.

B. DRL-based Solution Approach

Although the problem of determining a feasible action is addressed above, obtaining a policy that solves problem $\boldsymbol{\mathcal{P}}^{\rm MDP}$ is still challenging, since the transition probabilities between the states may not be available. We develop a DRL-based algorithm to gradually update the value function and policy without any knowledge of the transition probabilities between the states. Furthermore, to address high-dimensional and large state space $\mathcal{S}$ and continuous action space $\mathcal{A}^{\rm c}(\boldsymbol{s}_t)$ in state $\boldsymbol{s}_t$, we consider a *discretized* action space with *parameterized* value function and policy [18]. We use a deep neural network (DNN) shown in Fig. 2(a) with parameters vector $\boldsymbol{\vartheta}$, an input layer $\boldsymbol{s}$, and an output layer $V^{\widetilde{\boldsymbol{\pi}}}(\boldsymbol{s}, \boldsymbol{\vartheta})$ to obtain the value function in state $\boldsymbol{s}_t = \boldsymbol{s}$ for all $\boldsymbol{s} \in \mathcal{S}$. The continuous action space $\mathcal{A}^{\rm c}(\boldsymbol{s}_t)$ is discretized. We use a DNN shown in Fig. 2(b) with parameters vector $\boldsymbol{\theta}$, an input layer $\boldsymbol{s}$, and an output layer with $N-1$ softmax functions to obtain $\widetilde{\pi}(\widetilde{P}_n^{\rm c}(\boldsymbol{s}) \,|\, \boldsymbol{s},\, \boldsymbol{\theta})$ for household $n \in \mathcal{N}^-$ in state $\boldsymbol{s}_t = \boldsymbol{s}$ for all $\boldsymbol{s} \in \mathcal{S}$.

In practice, the load aggregator cannot perform the forward propagation and back propagation steps [18] in a centralized fashion since household n may prefer not to reveal its state $\boldsymbol{s}_{n,t}$ and discomfort cost to the load aggregator due to privacy concerns. To address this issue, in the next section, we develop a *decentralized* learning algorithm executed by the ECCs.

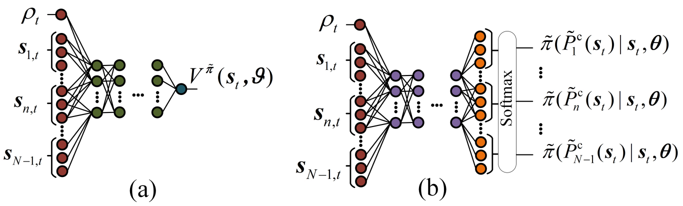

ρ_t
$s_{1,t}$
$s_{n,t}$
$s_{N-1,t}$
$V^{\tilde{\pi}}(s_t, \vartheta)$
(a)
ρ_t
$s_{1,t}$
$s_{n,t}$
$s_{N-1,t}$
Softmax
$\tilde{\pi}(\tilde{P}_1^c(s_t) \,|\, s_t, \theta)$
$\tilde{\pi}(\tilde{P}_n^c(s_t) \,|\, s_t, \theta)$
$\tilde{\pi}(\tilde{P}_{N-1}^c(s_t) \,|\, s_t, \theta)$
(b)

Figure 2. DNN for (a) the value function and (b) the policy.

IV. Decentralized Algorithm Design

We use an actor-critic-based reinforcement learning framework [34] to determine the optimal neural network parameters. To implement the proposed learning approach in a decentralized fashion, we apply federated learning [32], [33], where each ECC is responsible for updating the network parameters and sends the updated parameters to the load aggregator.

A. Household Partial Observability

Using federated learning, ECCs do not reveal their private information (i.e., state, discomfort cost) to the load aggregator. However, ECC n requires information about system state $\boldsymbol{s}_t$ to update the network parameters using back propagation.

Assumption 2: The ECC of household n can only observe the state in its own household.

Assumption 2 implies that ECC n partially observes system state $\boldsymbol{s}_t$. Due to privacy concerns, a household prefers not to reveal its local state to other households. We address the partial observability of the ECCs without revealing the state of each household to other households. Consider the set of nodes $\mathcal{J}$ in the first hidden layer of the global DNN for the value function. As Fig. 3(a) shows, for household n, we define parameter $\lambda_{j,n,t}$ as the aggregate input from the nodes of state $\boldsymbol{s}_{n,t}$ to node $j \in \mathcal{J}$ in the DNN for the value function. For household n and node $j \in \mathcal{J}$, we define parameter $\overline{\lambda}_{j,n,t} = \sum_{n' \in \mathcal{N}^- \setminus \{n\}} \lambda_{j,n',t}$. We define vectors $\boldsymbol{\lambda}_{n,t} = (\lambda_{j,n,t},\, j \in \mathcal{J})$ and $\overline{\boldsymbol{\lambda}}_{n,t} = (\overline{\lambda}_{j,n,t},\, j \in \mathcal{J})$ for $n \in \mathcal{N}^-$. Similarly, we define vectors $\boldsymbol{\gamma}_{n,t}$ and $\overline{\boldsymbol{\gamma}}_{n,t}$ for the DNN associated with the policy. To perform forward propagation and back propagation, observing electricity price ρ_t, local state $\boldsymbol{s}_{n,t}$, and vectors $\overline{\boldsymbol{\lambda}}_{n,t}$ and $\overline{\boldsymbol{\gamma}}_{n,t}$ is equivalent to observing the system state $\boldsymbol{s}_t$. Meanwhile, the states of other households are not revealed to ECC n. For household $n \in \mathcal{N}^-$, we use the DNN in Fig. 2(a) for the value function to construct another DNN shown in Fig. 3(b) with parameters vector $\boldsymbol{\vartheta}_n$. Instead of the joint state $\boldsymbol{s}_t$, the input layer in the DNN of household n for the value function includes the electricity price ρ_t, local state $\boldsymbol{s}_{n,t}$, and a bias node with input equal to 1 and weights $\overline{\boldsymbol{\lambda}}_{n,t}$. We use the DNN in Fig. 2(b) for the policy to construct another DNN shown in Fig. 3(c) for household n with parameters vector $\boldsymbol{\theta}_n$. The input layer includes the electricity price ρ_t, local state $\boldsymbol{s}_{n,t}$, and a bias node with input equal to 1 and weights $\overline{\boldsymbol{\gamma}}_{n,t}$. The output layer is the policy $\widetilde{\pi}(\widetilde{P}_n^{\rm c}(\boldsymbol{s}_t) \,|\, \boldsymbol{s}_t, \boldsymbol{\theta}_n)$ for household n.

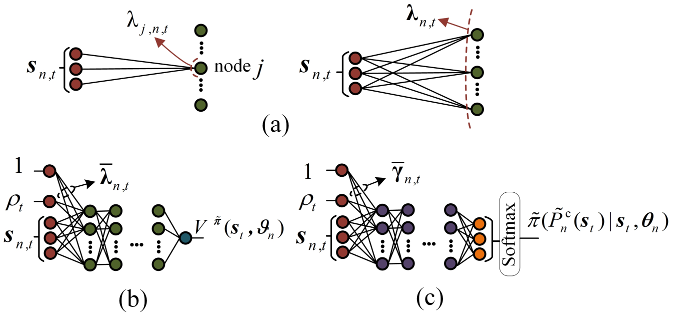

$\lambda_{j,n,t}$
node j
$s_{n,t}$
$\boldsymbol{\lambda}_{n,t}$
$s_{n,t}$
(a)
1
$\overline{\boldsymbol{\lambda}}_{n,t}$
ρ_t
$s_{n,t}$
$V^{\tilde{\pi}}(s_t, \vartheta_n)$
(b)
1
$\overline{\boldsymbol{\gamma}}_{n,t}$
ρ_t
$s_{n,t}$
Softmax
$\tilde{\pi}(\tilde{P}_n^c(s_t) \,|\, s_t, \theta_n)$
(c)

Figure 3. (a) Parameter $\lambda_{j,n,t}$ and vector $\boldsymbol{\lambda}_{n,t}$ for the DNN associated with the value function for household n. The DNN for household n associated with (b) the value function and (c) the policy.

B. Algorithm Description

Algorithm 1 describes our proposed decentralized load control algorithm. An iteration of Algorithm 1 corresponds to one time slot. Lines 1 and 2 describe the initialization in time slot $t = 1$, where each ECC n randomly chooses values for parameters $\boldsymbol{\theta}_{n,t}$ and $\boldsymbol{\vartheta}_{n,t}$ for its local DNNs. The loop involving Lines 3 to 20 encompasses the information exchange Lines 4 to 6, actor and critic updates Lines 8 to 13, and action selection Lines 15 to 18 in time slot t. In Lines 4 to 6, ECC n receives information about the joint state of other households by receiving vectors $\overline{\boldsymbol{\lambda}}_{n,t}$ and $\overline{\boldsymbol{\gamma}}_{n,t}$ from the load aggregator.

For time slot $t = 1$, ECC n performs the action selection in Lines 15 to 18 to determine a feasible action in state $\boldsymbol{s}_1$. In time slot $t > 1$, ECC n performs the actor and critic updates in Lines 8 to 12. In Line 8, ECC n computes the temporal difference (TD) error $\delta_n(\boldsymbol{\vartheta}_{n,t-1})$ corresponding to the previous time slot $t-1$, as follows:

$$\begin{aligned} \delta_n(\boldsymbol{\vartheta}_{n,t-1}) = {} & (N-1)\, c_n(\boldsymbol{s}_{t-1},\, \boldsymbol{a}(\boldsymbol{s}_{t-1})) \\ & + \beta\, V^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta}_{n,t-1})}(\boldsymbol{s}_t,\, \boldsymbol{\vartheta}_{n,t-1}) \\ & - V^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta}_{n,t-1})}(\boldsymbol{s}_{t-1},\, \boldsymbol{\vartheta}_{n,t-1}). \end{aligned} \tag{10}$$

The TD error in (10) approximates $Q^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s}, \boldsymbol{a}(\boldsymbol{s}))$ $-V^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s}, \boldsymbol{\vartheta})$ in time slot $t-1$. ECC n obtains the value function in states $\boldsymbol{s}_{t-1}$ and $\boldsymbol{s}_t$ using forward propagation in the local neural network for the value function [18].

In Appendix B, we show that ECC n can perform the following critic update to determine the updated network parameter $\boldsymbol{\vartheta}_{n,t}$ in Line 9:

$$\boldsymbol{\vartheta}_{n,t} = \boldsymbol{\vartheta}_{n,t-1} - \eta_t\, \nabla_{\boldsymbol{\vartheta}_n} \big(\delta_n(\boldsymbol{\vartheta}_n)\big)^2 \Big|_{\boldsymbol{\vartheta}_n = \boldsymbol{\vartheta}_{n,t-1}}, \tag{11}$$

where η_t is the step size for the critic update in time slot t. In (11), the gradient in $\boldsymbol{\vartheta}_n = \boldsymbol{\vartheta}_{n,t-1}$ is obtained using back propagation in the neural network for the value function [18].

Appendix B also shows that ECC n can perform the following actor update to determine the updated parameter vector $\boldsymbol{\theta}_{n,t}$ in Line 10:

$$\begin{aligned} \boldsymbol{\theta}_{n,t} = {} & \boldsymbol{\theta}_{n,t-1} + \mu_t\, (N-1)\, \delta_n(\boldsymbol{\vartheta}_{n,t-1}) \\ & \times \nabla_{\boldsymbol{\theta}_n} \Big(\ln \big(\widetilde{\pi}(\widetilde{P}_n^{\rm c}(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}_n)\big)\Big)\Big|_{\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n,t-1}}, \end{aligned} \tag{12}$$

where μ_t is the step size for the actor update in time slot t. In (12), the gradient vector in $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n,t-1}$ is obtained using back

Algorithm 1 Decentralized Load Control Algorithm.

```
1:  Set $t := 1$, $\varepsilon := 10^{-3}$.
2:  ECC $n$ randomly initializes parameters $\boldsymbol{\theta}_{n,1}$ and $\boldsymbol{\vartheta}_{n,1}$.
3:  Repeat
4:    ECC $n$ observes state $\boldsymbol{s}_{n,t}$ and electricity price $\rho_t$.
5:    ECC $n$ sends $\boldsymbol{\lambda}_{n,t}$ and $\boldsymbol{\gamma}_{n,t}$ to the load aggregator.
6:    Load aggregator sends vectors $\overline{\boldsymbol{\lambda}}_{n,t}$ and $\overline{\boldsymbol{\gamma}}_{n,t}$ to ECC $n$.
7:    If $t \neq 1$,
8:      ECC $n$ obtains the TD error according to (10).
9:      ECC $n$ obtains the updated $\boldsymbol{\vartheta}_{n,t}$ according to (11).
10:     ECC $n$ determines the updated $\boldsymbol{\theta}_{n,t}$ according to (12).
11:     ECC $n$ sends the updated network parameter vectors
        $\widehat{\boldsymbol{\vartheta}}_{n,t}$ and $\widehat{\boldsymbol{\theta}}_{n,t}$ to the load aggregator.
12:     Load aggregator computes the updated parameters
        $\widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_t$ according to (13a) and (13b), respectively.
        It broadcasts vectors $\widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_t$ to ECCs $n \in \mathcal{N}^-$.
13:     ECC $n$ sets $\widehat{\boldsymbol{\vartheta}}_{n,t} := \widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_{n,t} := \widehat{\boldsymbol{\theta}}_t$.
14:   End if
15:   ECC $n$ uses policy $\widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_t) \,|\, \boldsymbol{s}_t,\, \boldsymbol{\theta}_{n,t})$ to obtain
      controllable load demand $\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_t)$.
16:   Load aggregator solves optimization problem $\boldsymbol{\mathcal{P}}_2^{\text{proj}}$ to
      obtain a feasible action vector $\boldsymbol{a}(\boldsymbol{s}_t)$.
17:   Load aggregator sends the feasible action vector $\boldsymbol{a}_n(\boldsymbol{s}_t)$
      to ECC $n$.
18:   ECC $n$ receives the immediate cost $c_n(\boldsymbol{s}_t,\, \boldsymbol{a}(\boldsymbol{s}_t))$.
19:   $t := t + 1$.
20: Until $||\boldsymbol{\vartheta}_{n,t-1} - \boldsymbol{\vartheta}_{n,t-2}|| < \varepsilon$ and $||\boldsymbol{\theta}_{n,t-1} - \boldsymbol{\theta}_{n,t-2}|| <$
    $\varepsilon,\ n \in \mathcal{N}^-,\ t > 2$.
```

propagation in the local neural network for the policy [18].

In Lines 11 to 13, the updated local network parameters for the households are aggregated. ECC n constructs network parameters $\widehat{\boldsymbol{\vartheta}}_{n,t}$ and $\widehat{\boldsymbol{\theta}}_{n,t}$ from vectors $\boldsymbol{\vartheta}_{n,t}$ and $\boldsymbol{\theta}_{n,t}$ by excluding the parameters for the nodes associated with state $\boldsymbol{s}_{n,t}$ in the DNNs for the value function and policy, respectively. In Line 11, ECC n sends network parameter vectors $\widehat{\boldsymbol{\vartheta}}_{n,t}$ and $\widehat{\boldsymbol{\theta}}_{n,t}$ to the load aggregator. In Line 12, load aggregator computes the network parameters $\widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_t$ as the average value of parameters $\widehat{\boldsymbol{\theta}}_{n,t}$ and $\widehat{\boldsymbol{\vartheta}}_{n,t}$, $n \in \mathcal{N}^-$. We have

$$\widehat{\boldsymbol{\vartheta}}_t = \frac{1}{N-1} \sum_{n \in \mathcal{N}^-} \widehat{\boldsymbol{\vartheta}}_{n,t}, \tag{13a}$$

$$\widehat{\boldsymbol{\theta}}_t = \frac{1}{N-1} \sum_{n \in \mathcal{N}^-} \widehat{\boldsymbol{\theta}}_{n,t}. \tag{13b}$$

The load aggregator broadcasts parameter vectors $\widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_t$ to ECC $n \in \mathcal{N}^-$. In Line 13, ECC n sets network parameters $\widehat{\boldsymbol{\theta}}_{n,t}$ and $\widehat{\boldsymbol{\vartheta}}_{n,t}$ to parameter vectors $\widehat{\boldsymbol{\vartheta}}_t$ and $\widehat{\boldsymbol{\theta}}_t$.

In Line 15, ECC n selects $\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_t)$ and informs the load aggregator. In Line 16, the load aggregator determines a feasible action vector $\boldsymbol{a}(\boldsymbol{s}_t)$ by solving problem $\boldsymbol{\mathcal{P}}_2^{\text{proj}}$. In Line 17, the load aggregator sends the feasible action $\boldsymbol{a}_n(\boldsymbol{s}_t) = (P^{\text{c}}_n(\boldsymbol{s}_t),\, \mathbf{W}_n(\boldsymbol{s}_t))$ to ECC n. In Line 18, ECC n receives immediate cost for the action vector $\boldsymbol{a}_n(\boldsymbol{s}_t)$. Next time slot begins in Line 19. In Line 20, the stopping criterion is given.

Remark 1: For Algorithm 1 to converge to the solution of problem $\boldsymbol{\mathcal{P}}^{\text{MDP}}$, it is necessary that η_t and μ_t satisfy $\sum_{t=1}^{\infty}\eta_t = \sum_{t=1}^{\infty}\mu_t = \infty$ and $\sum_{t=1}^{\infty}(\eta_t)^2 = \sum_{t=1}^{\infty}(\mu_t)^2 < \infty$, and $\sum_{t=1}^{\infty}\big(\mu_t/\eta_t\big)^{\varsigma} < \infty$ for some $\varsigma > 0$ [34].

Remark 2: To solve problem $\boldsymbol{\mathcal{P}}_2^{\text{proj}}$ in Line 16 of Algorithm 1, the load aggregator requires the households' state, which is not available due to privacy concerns. Instead, the load aggregator can apply the proposed *decentralized* algorithm in [38], which is based on the proximal Jacobian alternating direction method of multipliers (PJ-ADMM) [39, Algorithm 4] with prox-linear method [40] to decompose problem $\boldsymbol{\mathcal{P}}_2^{\text{proj}}$ into subproblems for the substation and households.

Remark 3: Algorithm 1 preserves the privacy of each household. The elements of vectors $\boldsymbol{\lambda}_{n,t}$ and $\boldsymbol{\gamma}_{n,t}$ for household n are linear combinations of the household's state and the parameters for the nodes associated with state $\boldsymbol{s}_{n,t}$ in the DNNs for the value function and policy, respectively. Hence, the load aggregator cannot infer any single household's state from vectors $\boldsymbol{\lambda}_{n,t}$ and $\boldsymbol{\gamma}_{n,t}$. Moreover, the parameters for the nodes associated with state $\boldsymbol{s}_{n,t}$ are excluded from vectors $\widehat{\boldsymbol{\vartheta}}_{n,t}$ and $\widehat{\boldsymbol{\theta}}_{n,t}$. Thus, the load aggregator cannot infer the household's state from network parameters $\widehat{\boldsymbol{\vartheta}}_{n,t}$ and $\widehat{\boldsymbol{\theta}}_{n,t}$.

V. Performance Evaluation

We evaluate the performance of the proposed load control algorithm on an IEEE 33-bus distribution feeder with 32 households. The system data is sourced from [41]. We set the parameters used in power flow constraints (4a)$-$(4i). The lower and upper bounds for all bus voltage magnitudes are set to 0.9 pu and 1.1 pu, respectively. The maximum apparent power flow through each transmission line is set to 1.1 pu. Initially, no limit is considered for the active and reactive power injected at the substation bus. One day is divided into 96 time slots, each with duration of 15 minutes. We initially consider a time-of-use pricing scheme with rates shown in Fig. 4. We consider seven controllable and seven uncontrollable appliances for each household. We use the state model in [35] to obtain the MDP for the appliances in each household. Then we obtain the MDP for the base load $P^{\text{b}}_{n,t}$, and parameters $P^{\text{c,min}}_{n,t}$ and $P^{\text{c,des}}_{n,t}$, $t \in \mathcal{T}$, $n \in \mathcal{N}^-$. The discount factor β is set to 0.9. We use a piecewise linear function $d_n\big(P^{\text{c}}_n(\boldsymbol{s}_t),\, P^{\text{c,des}}_{n,t}\big) = \omega_{n,t}|P^{\text{c}}_n(\boldsymbol{s}_t) - P^{\text{c,des}}_{n,t}|$ to obtain the value of the discomfort cost for household n in time slot t, where the weighting coefficient $\omega_{n,t}$ is uniformly chosen at random from the interval [0.1, 1] cents / kW between 9 am of the current day and 6 am of the next day, and is set to 5 cents / kW otherwise. The power factor for each household in each time slot is uniformly sampled at random from interval [0.8, 0.9]. For the actor, we consider a neural network comprising three hidden layers with 30 nodes. We consider six levels to obtain a discrete action space for each household. For the critic, we consider a neural network comprising three hidden layers with 30 nodes. We use leaky rectified linear unit (ReLU) activation function. The step sizes for the critic and actor updates are set to $\eta_{n,t} = 15/t^{0.7}$ and $\mu_{n,t} = 8/t$. We perform simulations using MATLAB/CVX with MOSEK solver and PYTORCH library in PYTHON 3.7.

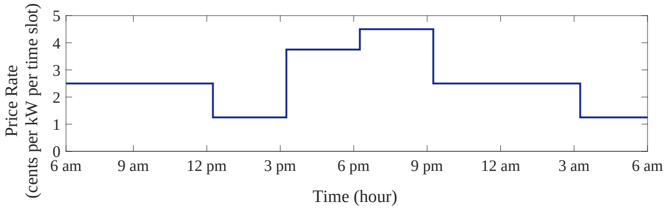


Figure 4. Electricity price rates during one day.

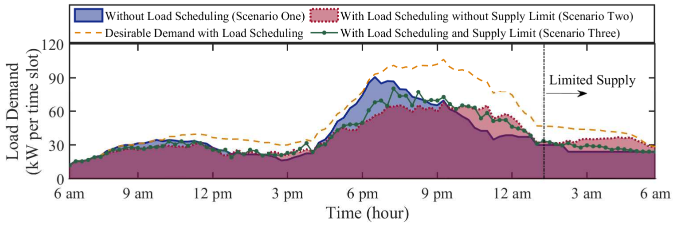


Figure 5. Aggregate demand of 32 households in the feeder in day 30.

A. Reducing Peak Load and User Cost

In Fig. 5, we show the feeder's aggregate load demand in three scenarios on day 30 as an example. In scenario one, we consider the load profile without load scheduling. The peak load demand is 90 kW at 7 pm. Scenario two shows the load demand when the households use Algorithm 1 for load scheduling. The peak load is reduced from 90 kW to 60 kW (i.e., 33% reduction). Reducing the controllable load in a time slot causes the desirable demand increases in upcoming time slots. As Fig. 5 shows, the desirable demand with load scheduling is greater than or equal to the demand without load scheduling. This can be interpreted as shifting the load demand to the future. In scenario three, we consider an upper limit of 30 kW for the active power injection into the substation from 1 am to 6 am to show that, indeed, the load control policy is updated to satisfy the network constraints. Fig. 5 shows that, with a limited injected active power, the scheduled load demand increases from 5 pm to 9 pm to avoid shifting too much load to the time period from 1 am to 6 am.

We show that the policy and value function converge to the (local) optimal solution of $\mathcal{P}^{\text{MDP}}$. The critic update in (11) aims to decrease the TD error. Fig. 6(a) shows the convergence of the average TD errors of all households during the first 30 days. The TD error may not be zero, since it *approximates* the difference between the action-value function and the value function. The goal of actor update in (12) is to obtain a policy that results in a lower value function. Fig. 6(b) shows the convergence of the value function from \$12 to \$7 for a given initial state of household 1. Fig. 6(c) shows the convergence of the expected daily cost of the *controllable* loads in a household from 95 cents to 65 cents after 30 days. Fig. 6(d) shows that the expected daily cost per household is reduced by 13% (from \$3.38 to \$2.94) with load scheduling in day 30.

B. Comparing with State-of-the-art Algorithms

We compare the convergence of Algorithm 1 with centralized learning algorithms based on actor-critic method, Q-learning, and double Q-learning, which have been proposed

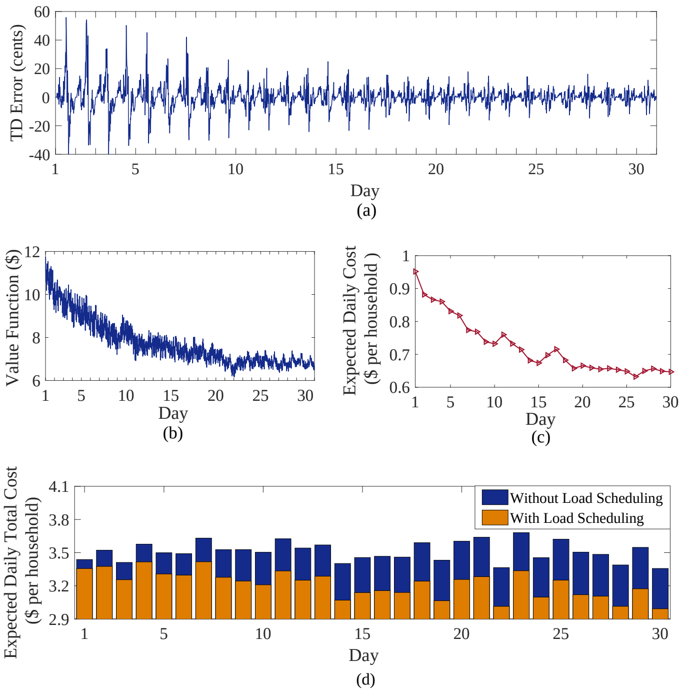


Figure 6. (a) Average TD error; (b) Value function for a given initial state; (c) Expected daily cost for controllable loads; (d) Expected daily total cost with and without load scheduling.

in the literature (e.g., [22], [27]–[29]). For the centralized algorithm with actor-critic method, we determine the parameters of the DNNs for the value function and policy. For the centralized algorithm with Q-learning, we consider a DNN comprising three hidden layers with 40 nodes. For the centralized algorithm with double Q-learning, we consider two DNNs comprising three hidden layers with 40 nodes. As shown in Fig. 7, Algorithm 1 converges to the suboptimal solution obtained from the centralized algorithm with actor-critic method. However, Algorithm 1 converges slightly slower and has higher oscillations around the suboptimal solution, as (10) approximates the TD error in the centralized algorithm by considering the cost of household n instead of the total cost of all households. When compared with the centralized algorithm using Q-learning, Algorithm 1 converges significantly more quickly to a suboptimal solution with a lower daily cost for the users. When compared with Q-learning, using double Q-learning improves the convergence speed and decreases the expected daily cost for the users. However, Algorithm 1 still converges more quickly to a suboptimal solution. Algorithm 1 with actor-critic method is preferred, since the ECC uses the critic DNN to learn the advantage function (i.e., the difference between the action-value function and the value function) instead of learning the action-value function. Thus, the evaluation is based on how much an action can improve the value function. Learning the advantage function significantly reduces the fluctuations in learning using the stochastic gradient decent method. Fig. 7 also demonstrates the performance of Algorithm 1 in training the DNNs parameters. Particularly, the expected daily cost for controllable loads with Algorithm 1 converges after day 30, which implies that the DNNs have successfully completed the training phase.

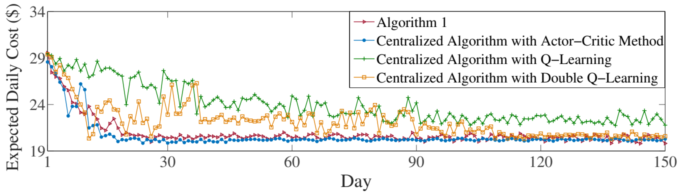
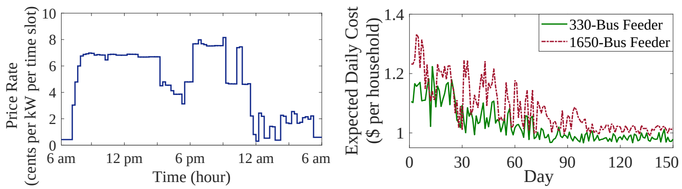


Figure 7. Expected daily cost for controllable loads with Algorithm 1, and centralized algorithms with actor-critic method, Q-learning, and double Q-learning.



(a)

(b)

Figure 8. (a) Real-time pricing rates during one day; (b) Expected daily cost for the controllable loads for a household in feeders with 330 and 1650 buses.

C. Demonstrating Scalability

We examine the convergence of Algorithm 1 in test systems with 330 and 1650 buses with real-time pricing scheme. We construct test systems with 330 and 1650 buses by connecting ten and fifty 33-bus feeders, respectively, via transmission lines with resistance of 0.01 pu and reactance of 0.0015 pu, . To mimic real-time pricing scheme, we use the historical price data for Ontario, Canada power grid database from Sept. 1, 2019 to Feb. 29, 2020 [42]. Fig. 8(a) shows the pricing rates per time slot from Feb. 25, 2020 to Feb. 26, 2020 as an example. In practice, load aggregators may need to mitigate any large fluctuation in the price rate. Nevertheless, this scenario helps to demonstrate the performance of Algorithm 1 in a volatile electricity market. Fig. 8(b) shows that, by using Algorithm 1, the expected daily cost of the controllable loads per household converges to the suboptimal solution in 90 days and 105 days in test systems with 330 and 1650 buses, respectively. When compared with our original case study, the required number of days for convergence is higher in these large test systems, because the ECCs need to adjust their policy according to the time-varying price rates as well as the distribution network constraints in a larger test system. Nevertheless, Algorithm 1 can still be used in large test systems, since the expected daily cost of the controllable loads decreases gradually (e.g., approximately 10% reduction in 30 days), leading to a lower total cost for the households.

Finally, we evaluate the communication overhead of Algorithm 1. Each iteration of Algorithm 1 involves six message exchange between ECC n and load aggregator. To reduce the communication overhead, one can use minibatch gradient descent, where the batch size is set to more than one. However, using a small batch size achieves better training stability, as the parameters of the DNNs for the policy and value function are updated more frequently by the load aggregator.

VI. Conclusion

In this paper, we applied DRL to design a load scheduling algorithm for residential households under uncertainty in the electricity price, load demand, and users' discomfort cost. To address the users privacy concerns, we applied federated learning technique to develop a decentralized load scheduling algorithm executed by the users. Also, we accounted for distribution network constraints and transformed the problem of updating neural network parameters into a sequence of SDPs to deal with nonconvex power flow constraints. Via numerical simulations, we showed that the proposed load scheduling algorithm can benefit the load aggregator by 33% reduction in the aggregate demand during peak hours. It also benefits a user by 13% reduction in its expected daily cost. The proposed decentralized algorithm converged to the solution of the centralized algorithm in an acceptable number of iterations. Results showed that the proposed algorithm is applicable in a system with large number of users and real-time pricing scheme. For future work, we will consider the impact of data tampering and false information from the households on the DRL-based demand response algorithm.

Appendix

A. Proof of Theorem 1

We obtain problem $\boldsymbol{\mathcal{P}}_1^{\text{r-proj}}$ by relaxing rank-one constraint (4i) in problem $\boldsymbol{\mathcal{P}}_1^{\text{proj}}$ as follows:

$$
\begin{aligned}
\boldsymbol{\mathcal{P}}_1^{\text{r-proj}}: \quad & \underset{\boldsymbol{P}^{\text{c}}(\boldsymbol{s}_t),\, \mathbf{W}_n(\boldsymbol{s}_t),\, n\in\mathcal{N}}{\text{minimize}} \quad ||\boldsymbol{P}^{\text{c}}(\boldsymbol{s}_t) - \widetilde{\boldsymbol{P}}^{\text{c}}(\boldsymbol{s}_t)||_2^2 \\
& \text{subject to constraints (4a)}-\text{(4h)}, \\
& \qquad \mathbf{W}_n(\boldsymbol{s}_t) \succeq 0, \qquad n\in\mathcal{N}, \\
& \qquad \boldsymbol{P}^{\text{c}}(\boldsymbol{s}_t) \in \mathcal{A}^{\text{c}}(\boldsymbol{s}_t).
\end{aligned}
$$

The objective function of problem $\boldsymbol{\mathcal{P}}_2^{\text{r}}$ can be expressed as

$$
\begin{aligned}
& ||\boldsymbol{P}^{\text{c}}(\boldsymbol{s}_t) - \widetilde{\boldsymbol{P}}^{\text{c}}(\boldsymbol{s}_t)||_2^2 = \\
& \sum_{n\in\mathcal{N}^-} \left(\left(P_n^{\text{c}}(\boldsymbol{s}_t)\right)^2 - 2\,\widetilde{P}_n^{\text{c}}(\boldsymbol{s}_t)\,P_n^{\text{c}}(\boldsymbol{s}_t) + \left(\widetilde{P}_n^{\text{c}}(\boldsymbol{s}_t)\right)^2 \right). \qquad (14)
\end{aligned}
$$

Considering (14), we can interpret problem $\boldsymbol{\mathcal{P}}_1^{\text{r-proj}}$ as an optimal power flow (OPF) problem in the underlying distribution network, where the load in bus n has a quadratic cost function $\left(P_n^{\text{c}}(\boldsymbol{s}_t)\right)^2 - 2\,\widetilde{P}_n^{\text{c}}(\boldsymbol{s}_t)\,P_n^{\text{c}}(\boldsymbol{s}_t) + \left(\widetilde{P}_n^{\text{c}}(\boldsymbol{s}_t)\right)^2$. With quadratic cost function, practical distribution networks satisfy the sufficient conditions given in [37, Sec. IV-C] for the network topology and constraints. The sufficient conditions can be summarized as $i)$ the graph induced by $\text{Re}\{Y\}$ is connected, and $ii)$ the Lagrange multipliers associated with the active power balance constraints are non-negative and greater than or equal to the Lagrange multipliers associated with the reactive power balance constraints. Consequently, the SDP relaxation gap between $\boldsymbol{\mathcal{P}}_1^{\text{proj}}$ and $\boldsymbol{\mathcal{P}}_1^{\text{r-proj}}$ is zero. Problem $\boldsymbol{\mathcal{P}}_1^{\text{r-proj}}$ is equivalent to $\boldsymbol{\mathcal{P}}_2^{\text{proj}}$. This completes the proof. ■

B. Critic and Actor Updates for ECC $n \in \mathcal{N}^-$

The goal of critic update is to reduce the advantage function $A(\boldsymbol{s},\, \boldsymbol{\vartheta},\, \boldsymbol{\theta}) = \mathbb{E}\{Q^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}),\, \boldsymbol{\vartheta})\} - V^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s},\, \boldsymbol{\vartheta})$ for state

$\boldsymbol{s} \in \mathcal{S}$ by minimizing the following objective function [34]:

$$f^{\text{critic}}(\boldsymbol{\vartheta},\, \boldsymbol{\theta}) = \sum_{\boldsymbol{s}\in\mathcal{S}} A^2(\boldsymbol{s},\, \boldsymbol{\vartheta},\, \boldsymbol{\theta}). \tag{15}$$

In time slot t, ECC n applies the stochastic gradient descent method and approximates the gradient of $f^{\text{critic}}(\boldsymbol{\vartheta},\, \boldsymbol{\theta})$ with respect to $\boldsymbol{\vartheta}$ by the gradient of $A^2(\boldsymbol{s}_{t-1},\, \boldsymbol{\vartheta},\, \boldsymbol{\theta})$ in state $\boldsymbol{s}_{t-1}$. Moreover, ECC n approximates the advantage function $A(\boldsymbol{s}_{t-1},\, \boldsymbol{\vartheta},\, \boldsymbol{\theta})$ by the TD error $\delta_n(\boldsymbol{\vartheta}_{n,t-1})$ in (10) for the local DNNs of household n. Hence, ECC n can use (11) to obtain the updated network parameter $\boldsymbol{\vartheta}_{n,t}$. By taking the average of the local network parameters in (13a), the global network parameter vector $\boldsymbol{\vartheta}_{t-1}$ is updated in the stochastic gradient direction of the objective function $f^{\text{critic}}(\boldsymbol{\vartheta},\, \boldsymbol{\theta})$.

To improve the policy, the actor aims to minimize the following objective function [34]:

$$\begin{aligned} &f^{\text{actor}}(\boldsymbol{\vartheta},\, \boldsymbol{\theta}) = \\ &\sum_{\boldsymbol{s}\in\mathcal{S}} \Big(-\mathbb{E}\big\{ \ln\big(\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\text{c}}(\boldsymbol{s}) \,|\, \boldsymbol{s},\, \boldsymbol{\theta})\big)\, \widehat{A}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}),\, \boldsymbol{\vartheta},\, \boldsymbol{\theta}) \big\} \Big). \end{aligned} \tag{16}$$

where $\mathbb{E}\{\cdot\}$ is the expectation over the actions in state $\boldsymbol{s}$. Function $\widehat{A}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}),\, \boldsymbol{\vartheta},\, \boldsymbol{\theta}) = Q^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s},\, \boldsymbol{a}(\boldsymbol{s}),\, \boldsymbol{\vartheta}) - V^{\widetilde{\boldsymbol{\pi}}(\boldsymbol{\theta})}(\boldsymbol{s},\, \boldsymbol{\vartheta})$ is the advantage function in state $\boldsymbol{s}$ and action $\boldsymbol{a}(\boldsymbol{s})$ for network parameters $\boldsymbol{\vartheta}$ and $\boldsymbol{\theta}$. ECC n approximates the gradient of objective function $f^{\text{actor}}(\boldsymbol{\theta},\, \boldsymbol{\vartheta})$ with respect to $\boldsymbol{\theta}$ by the gradient of $-\ln\big(\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\text{c}}(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta})\big)\, \widehat{A}(\boldsymbol{s}_{t-1},\, \boldsymbol{a}(\boldsymbol{s}_{t-1}),\, \boldsymbol{\vartheta},\, \boldsymbol{\theta})$ in state $\boldsymbol{s}_{t-1}$. ECC n approximates $\widehat{A}(\boldsymbol{s}_{t-1},\, \boldsymbol{a}(\boldsymbol{s}_{t-1}),\, \boldsymbol{\vartheta},\, \boldsymbol{\theta})$ by the TD error $\delta_n(\boldsymbol{\vartheta}_{n,t-1})$ in (10) for the local DNNs of household n. Moreover, the decision making of the households are independent. Hence, we have

$$\widetilde{\pi}(\widetilde{\boldsymbol{P}}^{\text{c}}(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}) = \prod_{n\in\mathcal{N}^-} \widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}). \tag{17}$$

ECC n observes $\widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}_n)$. We replace the right-hand side of (17) by $\widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}_n)^{N-1}$. We have

$$\begin{aligned} &\ln\big(\widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}_n)^{N-1}\big) \\ &= (N-1)\ln\big(\widetilde{\pi}(\widetilde{P}^{\text{c}}_n(\boldsymbol{s}_{t-1}) \,|\, \boldsymbol{s}_{t-1},\, \boldsymbol{\theta}_n)\big). \end{aligned} \tag{18}$$

Using (17) and (18), ECC n can use (12) to obtain $\boldsymbol{\theta}_{n,t}$. By taking the average of the local network parameters in (13b), the global network parameters are updated in the stochastic gradient direction of the objective function $f^{\text{actor}}(\boldsymbol{\vartheta},\, \boldsymbol{\theta})$. ■

References

[1] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Commun. Surveys & Tuts.*, vol. 17, no. 1, pp. 152–178, First quarter 2015.

[2] M. Song and M. Amelin, "Price-maker bidding in day-ahead electricity market for a retailer with flexible demands," *IEEE Trans. on Power Systems*, vol. 33, no. 2, pp. 1948–1958, Mar. 2018.

[3] J. Saez-Gallego, M. Kohansal, A. Sadeghi-Mobarakeh, and J. M. Morales, "Optimal price-energy demand bids for aggregate price-responsive loads," *IEEE Trans. on Smart Grid*, vol. 9, no. 5, pp. 5005–5013, Sept. 2018.

[4] O. Erdinç, A. Taşcıkaraoğlu, N. G. Paterakis, Y. Eren, and J. P. S. Catalão, "End-user comfort oriented day-ahead planning for responsive residential HVAC demand aggregation considering weather forecasts," *IEEE Trans. on Smart Grid*, vol. 8, no. 1, pp. 362–372, Jan 2017.

[5] J. Ponoćko and J. V. Milanović, "Forecasting demand flexibility of aggregated residential load using smart meter data," *IEEE Trans. on Power Systems*, vol. 33, no. 5, pp. 5446–5455, Sept. 2018.

[6] F. Wang, B. Xiang, K. Li, X. Ge, H. Lu, J. Lai, and P. Dehghanian, "Smart households' aggregated capacity forecasting for load aggregators under incentive-based demand response programs," *IEEE Trans. on Industry Applications*, vol. 56, no. 2, pp. 1086–1097, Mar. 2020.

[7] H. Wu, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah, "Demand response exchange in the stochastic day-ahead scheduling with variable renewable generation," *IEEE Trans. on Sustainable Energy*, vol. 6, no. 2, pp. 516–525, Apr. 2015.

[8] F. Elghitani and W. Zhuang, "Aggregating a large number of residential appliances for demand response applications," *IEEE Trans. on Smart Grid*, vol. 9, no. 5, pp. 5092–5100, Sept. 2018.

[9] J. Jin and Y. Xu, "Optimal storage operation under demand charge," *IEEE Trans. on Power Systems*, vol. 32, no. 1, pp. 795–808, Jan. 2017.

[10] S. Tsai, Y. Tseng, and T. Chang, "Communication-efficient distributed demand response: A randomized ADMM approach," *IEEE Trans. on Smart Grid*, vol. 8, no. 3, pp. 1085–1095, May 2017.

[11] Y. Huang, L. Wang, W. Guo, Q. Kang, and Q. Wu, "Chance constrained optimization in a home energy management system," *IEEE Trans. on Smart Grid*, vol. 9, no. 1, pp. 252–260, Jan. 2018.

[12] Y. F. Du, L. Jiang, Y. Li, and Q. Wu, "A robust optimization approach for demand side scheduling considering uncertainty of manually operated appliances," *IEEE Trans. on Smart Grid*, vol. 9, no. 2, pp. 743–755, Mar. 2018.

[13] S. Wang, H. Gangammanavar, S. D. Ekşioğlu, and S. J. Mason, "Stochastic optimization for energy management in power systems with multiple microgrids," *IEEE Trans. on Smart Grid*, vol. 10, no. 1, pp. 1068–1079, Jan. 2019.

[14] L. Zheng and L. Cai, "A distributed demand response control strategy using Lyapunov optimization," *IEEE Trans. on Smart Grid*, vol. 5, no. 4, pp. 2075–2083, Jul. 2014.

[15] F. Elghitani and E. El-Saadany, "Smoothing net load demand variations using residential demand management," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 1, pp. 390–398, Jan. 2019.

[16] S. Kim and G. B. Giannakis, "An online convex optimization approach to real-time energy pricing for demand response," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2784–2793, Nov. 2017.

[17] A. Lesage-Landry and D. S. Callaway, "Dynamic and distributed online convex optimization for demand response of commercial buildings," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 632–637, Jul. 2020.

[18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of Int'l. Conf. on Machine Learning*, NYC, NY, Jun. 2016.

[19] B. Wang, Y. Li, W. Ming, and S. Wang, "Deep reinforcement learning method for demand response management of interruptible load," *IEEE Trans. on Smart Grid*, vol. 11, no. 4, pp. 3146–3155, Jul. 2020.

[20] H. Li, Z. Wan, and H. He, "Constrained EV charging scheduling based on safe deep reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2427–2439, May 2020.

[21] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 11, no. 2, pp. 1066–1076, Mar. 2020.

[22] Y. Liu, D. Zhang, and H. B. Gooi, "Optimization strategy based on deep reinforcement learning for home energy management," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 3, pp. 572–582, Sept. 2020.

[23] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sept. 2019.

[24] H. Li, Z. Wan, and H. He, "Real-time residential demand response," *IEEE Trans. on Smart Grid*, vol. 11, no. 5, pp. 4144–4154, Sept. 2020.

[25] L. Yu, W. Xie, D. Xie, Y. Zou, D. Zhang, Z. Sun, L. Zhang, Y. Zhang, and T. Jiang, "Deep reinforcement learning for smart home energy management," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2751–2762, Apr. 2020.

[26] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019.

[27] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Trans. on Smart Grid*, vol. 9, no. 4, pp. 3259–3269, Jul. 2018.

[28] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuška, and R. Belmans, “Residential demand response of thermostatically controlled loads using batch reinforcement learning,” *IEEE Trans. on Smart Grid*, vol. 8, no. 5, pp. 2149–2159, Sept. 2017.

[29] M. Babar, P. H. Nguyen, V. Ćuk, I. G. Kamphuis, M. Bongaerts, and Z. Hanzelka, “The evaluation of agile demand response: An applied methodology,” *IEEE Trans. on Smart Grid*, vol. 9, no. 6, pp. 6118–6127, Nov. 2018.

[30] R. Lu, S. H. Hong, and M. Yu, “Demand response for home energy management using reinforcement learning and artificial neural network,” *IEEE Trans. on Smart Grid*, vol. 10, no. 6, pp. 6629–6639, Nov. 2019.

[31] S. Bahrami, Y.C. Chen, and V.W.S. Wong, “Deep reinforcement learning for direct load control in distribution networks,” in *Proc. of IEEE Power Energy Society General Meeting*, Aug. 2020.

[32] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in *Proc. of NIPS Workshop on Private Multi-Party Machine Learning*, Barcelona, Spain, Dec. 2016.

[33] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Mag*, vol. 37, no. 3, pp. 50–60, May 2020.

[34] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, “A survey of actor-critic reinforcement learning: Standard and natural policy gradients,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[35] S. Bahrami, V.W.S. Wong, and J. Huang, “An online learning algorithm for demand response in smart grid,” *IEEE Trans. on Smart Grid*, vol. 9, no. 5, pp. 4712–4725, Sept. 2018.

[36] J. D. Glover, M. S. Sarma, and T. J. Overby, *Power Systems Analysis and Design*. Boston, MA: Cengage Learning Press, 2017.

[37] J. Lavaei and S. H. Low, “Zero duality gap in optimal power flow problem,” *IEEE Trans. on Power Systems*, vol. 27, no. 1, pp. 92–107, Feb. 2012.

[38] S. Bahrami, Y.C. Chen, and V.W.S. Wong, “An autonomous demand response algorithm based on online convex optimization,” in *Proc. of IEEE SmartGridComm*, Aalborg, Denmark, Oct. 2018.

[39] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, “Parallel multi-block ADMM with $o(1/k)$ convergence,” *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, May 2017.

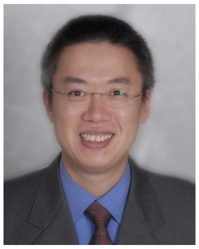
[40] G. Chen and M. Teboulle, “A proximal-based decomposition method for convex minimization problems,” *Mathematical Programming*, vol. 64, no. 1, pp. 81–101, Mar. 1994.

[41] R. D. Zimmerman and C. E. Murillo-Sanchez, “Matpower.” [Online]. Available: https://matpower.org.

[42] Independent Electricty System Operator (IESO). [Online]. Available: http://www.ieso.ca

Shahab Bahrami (M’17) received the B.A.Sc. and M.A.Sc. degrees both in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2010 and 2012, respectively. He received the Ph.D. degree in Electrical & Computer Engineering from the University of British Columbia (UBC), Vancouver, BC, Canada in 2017. Dr. Bahrami has received various prestigious scholarships at UBC, including the distinguished and highly competitive UBC’s Four Year Fellowship (2013−2017), and the Graduate Support Initiative Award from the Faculty of Applied Science at UBC (2014−2017). Currently, he works as a postdoctoral research fellow at UBC. His research interests include demand response, optimization, and algorithm design with applications to smart grid.

Yu Christine Chen (M’15) received the B.A.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 2009, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2011 and 2014, respectively. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, where she is affiliated with the Electric Power and Energy Systems Group. Her research interests include power system analysis, monitoring, and control.



Vincent W.S. Wong (S’94, M’00, SM’07, F’16) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microchip Technology Inc.). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, mobile edge computing, and Internet of Things. Currently, Dr. Wong is an Executive Editorial Committee Member of *IEEE Transactions on Wireless Communications*, an Area Editor of *IEEE Transactions on Communications* and *IEEE Open Journal of the Communications Society*, and an Associate Editor of *IEEE Transactions on Mobile Computing*. He is a Technical Program Co-chair of the *IEEE* 92*nd Vehicular Technology Conference (VTC*2020*-Fall)*. He has served as a Guest Editor of *IEEE Journal on Selected Areas in Communications* and *IEEE Wireless Communications*. He has also served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. He was a Tutorial Co-Chair of *IEEE Globecom*’18, a Technical Program Co-chair of *IEEE SmartGridComm*’14, as well as a Symposium Co-chair of *IEEE ICC*’18, *IEEE SmartGridComm* (’13, ’17) and *IEEE Globecom*’13. He is the Chair of the IEEE Vancouver Joint Communications Chapter and has served as the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications. He is an IEEE Fellow and an IEEE Communications Society Distinguished Lecturer.