Learning to Operate Distribution Networks with Safe Deep Reinforcement Learning

Hepeng Li, *Student Member, IEEE,* and Haibo He, *Fellow, IEEE,*

***Abstract*—In this paper, we propose a safe deep reinforcement learning (SDRL) based method to solve the problem of optimal operation of distribution networks (OODN). We formulate OODN as a constrained Markov decision process (CMDP). The objective is to achieve adaptive voltage regulation and energy cost minimization considering the uncertainty of renewable resources (RSs), nodal loads and energy prices. The control actions include the number of in-operation units of the switchable capacitor banks (SCBs), the tap position of the on-load tap-changers (OLTCs) and voltage regulators (VRs), the active and reactive power of distributed generators (DGs), and the charging and discharging power of battery storage systems (BSSs). To optimize the discrete and continuous actions simultaneously, a stochastic policy built upon a joint distribution of mixed random variables is designed and learned through a neural network approximator. To guarantee that safety constraints are satisfied, constrained policy optimization (CPO) is employed to train the neural network. The proposed approach enables the agent to learn a cost-effective operating strategy through exploring safe scheduling actions. Compared to traditional deep reinforcement learning (DRL) methods that allow agents to freely explore any behaviors during training, the proposed approach is more practical to be applied in a real system. Simulation results on a modified IEEE-34 node system and a modified IEEE-123 node system demonstrate the effectiveness of the proposed method.**

***Index Terms*—Distribution systems, safe deep reinforcement learning, constrained Markov decision process (MDP), mixed discrete and continuous actions, data-driven decision making.**

NOMENCLATURE

Abbreivations

OODN	Optimal Operation of Distribution Networks
DG	Distributed Generator
RS	Renewable Sources
BSS	Battery Storage System
SCB	Switchable Capacitor Bank
VR	Voltage Regulator
OLTC	On-Load Tap Changer
MDP	Markov Decision Process
CMDP	Constrained Markov Decision Process
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
SDRL	Safe Deep Reinforcement Learning
PPO	Proximal Policy Optimization
DDPG	Deep Deterministic Policy Gradient
SAC	Soft Actor Critic
CPO	Constrained Policy Optimization
VVC	Volt-Var Control
SP	Stochastic Programming
RO	Robust Optimization
MISOCP	Mixed Integer Second-Order Cone Programming

Subscripts

i	Index of node
ij	Index of branch
t	Index of time slot
c	Index of discrete control actions
k	Iteration of the CPO algorithm
ch	Charging
dch	Discharging

Superscripts

scb	Switchable capacitor bank
vr	Voltage regulator or on-load tap-changer
dg	Distributed generator
bss	Battery storage system
rs	renewable resources
d	power demand
s	Substation
targ	Target value

Sets

Ω_t	Set of time slots
Ω_n	Set of nodes
Ω_b	Set of branches

Variables

n	Number of SCB units in operation
l	Tap position of an OLTC/VR
P	Active power (kW)
Q	Reactive power (kVar)
S	Apparent power (kVA)
E	Energy stored in a BSS (kWh)
V	Nodal voltage (p.u.)
I	Branch current (p.u.)
R	Electricity rate ($/kWh)
s	State of the DN
a	Action of the policy
r	Reward
c	Constraint
p	Probability
θ	Parameters of the policy network
ϕ	Parameters of the value network

Constants

$\overline{n}$	Maximum number of SCB units
$\overline{l}$	Maximum number of regulation up/down steps

Hepeng Li and Haibo He are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, RI 02881 USA (e-mail: hepengli@uri.edu; haibohe@uri.edu).

This work was supported in part by the National Science Foundation under grant ECCS 1917275.

$\overline{P}$	Maximum active power (kW)
$\overline{S}$	Maximum apparent power (kVA)
$\overline{E}$	Maximum energy stored in a BSS (kWh)
$\overline{V}$	Maximum nodal voltage (p.u.)
$\overline{I}$	Maximum branch current (p.u.)
$\underline{V}$	Minimum nodal voltage (p.u.)
$\underline{E}$	Minimum energy stored in a BSS (kWh)
$\underline{pf}$	Minimum power factor
$\overline{\delta}$	KL-Divergence Limit
d	Constraint tolerance
λ	GAE parameter
γ	Discount factor
ϱ	Penalty coefficient
Δt	Interval of one time step
T	Number of steps

I. Introduction

DISTRIBUTION networks were traditionally operated to avoid loading and voltage limit violations for one-way power flow [1]. The increasing penetration of distributed resources violates this basic assumption and can disrupt the operation of distribution networks. For instance, intermittent RSs, such as photovoltaics and wind turbines, may cause swing of voltages due to their rapid power variations [2]. DGs injecting real power back upstream into the distribution networks can cause voltage boosts [3] and interfere with conventional Volt/VAR control (VVC) devices. This problem may become worse if BSSs are installed and dispatched to charge and discharge intermittently. Besides, unregulated charging power of BSSs may increase the burden of distribution lines and reduce the loading margin of distribution systems.

To coordinate VVC devices and distributed resources, extensive *model-based* methods have been proposed. For example, a mixed-integer non-linear programming model and an equivalent mixed integer quadratically constrained model were proposed in [4] for voltage constraint management in distribution networks. In [5], a mixed-integer second-order cone programming (MISOCP) model was proposed to minimize the energy cost for radial distribution networks using branch power flow models. To consider uncertainty of RSs, a two-stage stochastic programming (SP) model was proposed in [6] to coordinate VVC devices and inverter-interfaced RSs. However, SP-based methods generally assume a known probability distribution of the uncertainty. To relax the assumption, a two-stage robust optimization (RO) model was proposed in [7] to minimize the worst-case network loss. In [8], an extended two-stage RO model was developed to minimize the day-ahead operational cost considering the uncertainty of RSs generation and load demand. In [9], a distributionally robust model predictive control method was proposed to solve dynamic optimal power flow in a multi-microgrid system. Although RO-based methods do not require a known probability distribution of the uncertainty, they still depend on an uncertainty set to characterize the uncertainties. To construct the uncertainty set, a polyhedron or convex hull model is typically required. In [10], a deep neural network is employed to construct the uncertainty set from historical data for distribution network reconfiguration (DNR). Along this line, the authors in [11] developed a distributionally robust model for three-phase unbalanced DNR based on distributional ambiguity set.

Generally, *model-based* methods require an explicit physical model to formulate the distribution network, an accurate statistical model to characterize the uncertainty, and an efficient solver to obtain the optimal solution in a limited time. Developing such a method relies on extensive domain knowledge and human-effort on model selection, parameter estimation, and algorithm design. Improper physical models or inaccurate parameters may result in performance deterioration or unrealistic solutions.

To remove the dependency on an explicit model of the distribution network and of the uncertainty, *model-free* methods based on reinforcement learning (RL) techniques have received extensive attention in recent years. RL-based methods have been successfully used in many power system applications, such as reactive power control [12], [13], VVC [14], microgrid energy management [15]–[17], demand response [18] etc. For the OODN problem, some state-of-the-art methods have also adopted deep RL (DRL) based approaches by taking advantage of deep neural networks. For example, in [19], a two-timescale voltage control scheme was proposed to maintain bus voltage in distribution networks, where the on-off commitment of capacitor units was optimized by using deep-Q network (DQN). In [20], a safe DRL algorithm was developed to optimize the tap position of VRs and on/off switching of SCBS based on soft actor critic (SAC). In [2], the multi-agent deep deterministic policy optimization was adopted to solve the voltage regulation problem by coordinating the reactive power output of PV inverters. However, the VVC devices, such as SCBs, OLTCs, and VRs, have not been considered. In [21], a multi-agent DQN algorithm was developed to solve the VVC problem by controlling the VVC devices and PV inverters. In [22], a similar problem was solved by using a multi-agent trust region policy optimization based approach.

Nevertheless, the *model-free* methods mentioned above did not consider the co-optimization of conventional VVC devices and the emerging DGs and BSSs. As we pointed out, dispacthable DGs and BSSs can interfere with conventional VVC devices and undermine the operation of distribution networks. However, co-optimizion of VVC devices, dispatchable DGs, and BSSs may pose several challenges to traditional RL based methods. First, there are many inequality constraints in the OODN problem, which can be tricky for *reward-driven* RL methods. Second, there exist plenty of heterogeneous devices that are controlled via discrete or continuous actions. Third, distribution systems exhibit serious uncertainty and nonlinearity, which is a major challenge to the representation and learning ability of a completely *model-free* algorithm.

These challenges have motivated us to investigate a SDRL solution to the OODN problem. Along this line, we formulate the OODN problem in the framework of CMDP. In CMDP, we can handle the reward and the constraints independently and do not need to carefully design specific reward functions for constraint violation. To effectively restrict the constraints and maximize the reward, we explore the application of CPO,

which is a successful SDRL algorithm [23]. It can train complicated nonlinear policies for high-dimensional control problems with constraints on states and actions. It can also guarantee monotonic performance improvement and constraint satisfaction. These advantages make CPO suitable for the OODN problem. However, the OODN problem contains both discrete and continuous actions, which makes it challenging to directly apply the CPO algorithm. To deal with mixed discrete and continuous actions, we develop a stochastic control policy defined by a joint probabilistic distribution of discrete and continuous random variables. The policy can output discrete and continuous actions simultaneously by sampling from the joint distribution.

In this paper, we focus on the problem of OODN under uncertainty. We model the OODN problem as a CMDP considering the uncertainty of RSs generation, nodal loads, and prices of energy purchased from the utility. The objective is to achieve adaptive voltage regulation and energy cost minimization by coordinating SCBs, OLTCs, VRs, dispatchable DGs and BSSs. Compared to existing studies, the main contributions of this work are summarized as follows:

- We propose a CMDP formulation for the OODN problem considering the coordinated control of traditional VVC devices, i.e. SCBs, OLTCs, VRs, as well as the dispatchable DGs and BSSs. Compared to the existing model-free methods, the proposed formulation does not need to design specific reward function or tune penalty coefficients for constraint violation.
- We design a stochastic policy to handle mixed discrete and continuous actions. This policy enables us to explore a hybrid action space and generate discrete and continuous actions simultaneously. Since actions are generated by sampling from a joint distribution of mixed random variables, the designed policy does not have a scalability issue when the number of discrete actions increases.
- We employ the CPO algorithm to learn an optimal control policy for the OODN problem. Different from traditional DRL algorithms, the CPO algorithm can effectively handle the operational constraints and guarantee monotonic performance improvement and constraint satisfaction.

The rest of the paper is organized as follows. Section II presents the CMDP formulation. Section III introduces the CPO algorithm and the deisgned policy. In Section IV, case studies are carried out and discussed. Section V draws the conclusions.

II. CMDP Formulation of the OODN Problem

In our formulation, the operational horizon of a distribution network is divided into T time slots. We use the subscripts $t \in \Omega_t$ to index time intervals, $i \in \Omega_n$ to index network nodes, and $ij \in \Omega_b$ to index branches, where Ω_t, Ω_n, and Ω_b represent the set of time intervals, the set of nodes, and the set of branches, respectively.

Next, we introduce the characteristics of all controllable devices in the distribution network. Then, the operational limits and constraints of the distribution network are defined. Finally, the OODN problem is formulated as a CMDP. It is notable that in our formulation, the distribution network is considered as a black box. This means that the network topology, line parameters, and load fluctuation are unknown. Control policies and scheduling decisions have to be learned and made based on observations of system state.

A. Operational Characteristics of Controllable Devices

1) SCBs: The control variable of an SCB is the number of units in operation. For the SCB at node i, the control variable is denoted by $n_{i,t}^{\text{scb}}$, which can take integer values in the range

$$0 \le n_{i,t}^{\text{scb}} \le \overline{n}_i^{\text{scb}},\ i \in \Omega_n, t \in \Omega_t, \tag{1}$$

where $\overline{n}_i^{\text{scb}}$ represents the maximum number of units of the SCB. The total reactive power $Q_{i,t}^{\text{scb}}$ injected by the SCB is dependent on the susceptance of each unit, the number of units connected at the node, and the nodal voltage. A model of $Q_{i,t}^{\text{scb}}$ can be found in [21]. In our study, we do not need an explicit model of $Q_{i,t}^{\text{scb}}$.

2) OLTCs and VRs: The control variable of an OLTC/VR is the tap position. Commonly, an OLTC/VR can provide a voltage regulation from –10% to +10% with 5 or 33 steps [3]. For the OLTC/VR connected to branch ij, the control variable is denoted by $l_{ij,t}^{\text{vr}}$, which can take integer values in the range

$$-\overline{l}_{ij}^{\text{vr}} \le l_{ij,t}^{\text{vr}} \le \overline{l}_{ij}^{\text{vr}},\ ij \in \Omega_b, t \in \Omega_t, \tag{2}$$

where $\overline{l}_{ij}^{\text{vr}}$ is the maximum number of the regulation up/down steps of the OLTC/VR.

3) Dispatchable DGs: The control variables of a dispatchable DG are the active and reactive power outputs of the DG. For the DG at node i, the active and reactive power outputs are denoted by $P_{i,t}^{\text{dg}}$ and $Q_{i,t}^{\text{dg}}$, respectively. It is assumed that dispatchable DGs operate with a restricted power factor [5]; thus $P_{i,t}^{\text{dg}}$ and $Q_{i,t}^{\text{dg}}$ are constrained by:

$$0 \le P_{i,t}^{\text{dg}} \le \underline{pf}_i^{\text{dg}} \cdot \overline{S}_i^{\text{dg}},\ i \in \Omega_n, t \in \Omega_t, \tag{3}$$

$$\underline{pf}_i^{\text{dg}} \le \cos(\tan^{-1}(Q_{i,t}^{\text{dg}}/P_{i,t}^{\text{dg}})) \le 1,\ i \in \Omega_n, t \in \Omega_t, \tag{4}$$

where $\overline{S}_i^{\text{dg}}$ is the nominal capacity of the DG, and $\underline{pf}_i^{\text{dg}}$ is the minimum power factor.

4) BSSs: The control variable of a BSS is the charging and discharging power. For the BSS at node i, the control variable is denoted by $P_{i,t}^{\text{bss}}$, and a positive value of $P_{i,t}^{\text{bss}}$ represents the BSS is charging, and a negative value represents discharging. The value of $P_{i,t}^{\text{bss}}$ is restricted in the following range:

$$-\overline{P}_{i,dch}^{\text{bss}} \le P_{i,t}^{\text{bss}} \le \overline{P}_{i,ch}^{\text{bss}},\ i \in \Omega_n, t \in \Omega_t. \tag{5}$$

where $\overline{P}_{i,ch}^{\text{bss}}$ and $\overline{P}_{i,dch}^{\text{bss}}$ represent the maximum charging power and maximum discharging power, respectively. Due to the capacity limit of a energy storage, the energy $E_{i,t}^{\text{bss}}$ stored in the BSS at time interval t is constrained by

$$\underline{E}_i^{\text{bss}} \le E_{i,t}^{\text{bss}} \le \overline{E}_i^{\text{bss}},\ i \in \Omega_n, t \in \Omega_t, \tag{6}$$

where $\overline{E}_i^{\text{bss}}$ is the energy capacity of the BSS; $\underline{E}_i^{\text{bss}}$ denotes the allowable minimum energy stored in the BSS.

B. Limits and Constraints of the Distribution Networks

For the considered distribution network, we use the notation $V_{i,t}$ to represent the nodal voltage at node i, and $I_{i,t}$ to represent the current on branch ij. Also, we use P_t^{s} and Q_t^{s} to denote the active and reactive power injected from the substation, respectively.

The distribution network operates with the following limits:

$$\left(P_t^{\mathrm{s}}\right)^2+\left(Q_t^{\mathrm{s}}\right)^2 \leq (\overline{S}^{\mathrm{s}})^2,\ t \in \Omega_t. \tag{7}$$

$$\underline{V}_i \leq V_{i,t} \leq \overline{V}_i,\ i \in \Omega_n, t \in \Omega_t, \tag{8}$$

$$0 \leq I_{ij,t} \leq \overline{I}_{ij},\ ij \in \Omega_b, t \in \Omega_t, \tag{9}$$

Eq. (7) constrains the complex power exchanged at the substation between the distribution network and the upper level grid. Eq. (8) restricts the nodal voltages to their upper and lower limits. Eq. (9) defines the maximum branch currents.

C. CMDP Formulation

One major challenge in modeling the OODN problem is how to handle the constraints. In most model-free methods, constraints are modeled as a negative rewards in the framework of Markov decision process (MDP) by using penalty methods. However, as discussed in [24], it is difficult to determine a good penalty coefficient to balance the constraint violation and the reward. Besides, penalty methods usually cannot guarantee that constraints are strictly satisfied even if a very large penalty coefficient is used. To overcome this issue, we propose a CMDP formulation for the OODN problem. In the following subsections, the basic elements of the proposed CMDP formulation are elaborated.

1) States: The system states at any time interval t are defined as

$$\begin{aligned} s_t = (&P_{1,t-T},\ldots,P_{1,t-1},Q_{1,t-T},\ldots,Q_{1,t-1},E_{1,t}^{\mathrm{bss}},\ldots,\\ &P_{i,t-T},\ldots,P_{i,t-1},Q_{i,t-T},\ldots,Q_{i,t-1},E_{i,t}^{\mathrm{bss}},\ldots,\\ &R_{t-T}^{\mathrm{s}},\ldots,R_{t-1}^{\mathrm{s}}), i \in \Omega_n, t \in \Omega_t, \end{aligned} \tag{10}$$

where $P_{i,t-T},\ldots,P_{i,t-1}$ denote the historical net active power demand at node i over the past T slots; $Q_{i,t-T},\ldots,Q_{i,t-1}$ denote the historical net reactive power demand at node i over the past T slots; $R_{t-T}^{\mathrm{s}},\ldots,R_{t-1}^{\mathrm{s}}$ denote the historical energy prices over the past T slots. The net active and reactive power demand at node i are calculated by

$$P_{i,t}=P_{i,t}^{\mathrm{d}}-P_{i,t}^{\mathrm{rs}},\ Q_{i,t}=Q_{i,t}^{\mathrm{d}},\ i \in \Omega_n, t \in \Omega_t, \tag{11}$$

where $P_{i,t}^{\mathrm{rs}}$ represents the active power generated by the RS at node i, $P_{i,t}^{\mathrm{d}}$ denotes the active power demand at node t, and $Q_{i,t}^{\mathrm{d}}$ denotes the reactive power demand at node i. To sufficiently utilize RSs, we assume that RSs are nondispatchable sources operating with unity power factor [5].

2) Actions: The actions include the number of in-operation units of the SCBS, the tap position of OLTCs/VRs, the active and reactive power of DGs, and the charging/discharging power of BSSs:

$$\begin{aligned} a_t = (&n_{1,t}^{\mathrm{scb}},l_{1,t}^{\mathrm{vr}},P_{1,t}^{\mathrm{dg}},Q_{1,t}^{\mathrm{dg}},P_{1,t}^{\mathrm{bss}},\ldots,\\ &n_{i,t}^{\mathrm{scb}},l_{i,t}^{\mathrm{vr}},P_{i,t}^{\mathrm{dg}},Q_{i,t}^{\mathrm{dg}},P_{i,t}^{\mathrm{bss}},\ldots),\ i \in \Omega_n, t \in \Omega_t. \end{aligned} \tag{12}$$

3) Reward: The reward is the negative sum of the purchasing costs of energy at the substation and the fuel costs of DGs

$$r_t=-\left(R_t^{\mathrm{s}}P_t^{\mathrm{s}}\Delta t+\sum_{i\in\Omega_n}[a_i^{\mathrm{dg}}(P_{i,t}^{\mathrm{dg}})^2+b_i^{\mathrm{dg}}P_{i,t}^{\mathrm{dg}}+c_i^{\mathrm{dg}}]\Delta t\right), \tag{13}$$

where a_i^{dg}, b_i^{dg}, and c_i^{dg} are generation cost coefficients of the DG at node i.

4) Constraint: The constraint reflects the degree of constraint violations, which is defined by

$$c_t=C_t^{\mathrm{s}}+\sum_{i\in\Omega_n}C_{i,t}^{\mathrm{V}}+\sum_{ij\in\Omega_b}C_{ij,t}^{\mathrm{I}}+\sum_{i\in\Omega_n}C_{i,t}^{\mathrm{dg}}+\sum_{i\in\Omega_n}C_{i,t}^{\mathrm{bss}}. \tag{14}$$

The first term C_t^{s} measures the violation of the substation capacity constraint (7), which is calculated by:

$$C_t^{\mathrm{s}}=\max(0,\ \sqrt{(P_t^{\mathrm{s}})^2+(Q_t^{\mathrm{s}})^2}/\overline{S}^{\mathrm{s}}-1). \tag{15}$$

The second term $C_{i,t}^{\mathrm{V}}$ reflects the degree of violation of the nodal voltage limits (8), which is calculated by:

$$C_{i,t}^{\mathrm{V}}=\max(0,\ V_{i,t}-\overline{V}_i)+\max(0,\ \underline{V}_i-V_{i,t}). \tag{16}$$

The third term $C_{ij,t}^{\mathrm{I}}$ reflects the degree of violation of branch loading limits (9), which is calculated by:

$$C_{ij,t}^{\mathrm{I}}=\max(0,\ I_{ij,t}/\overline{I}_{ij}-1). \tag{17}$$

The fourth term $C_{i,t}^{\mathrm{dg}}$ assesses the violation of the power factor constraint (4) , which is calculated by:

$$C_{i,t}^{\mathrm{dg}}=\max(0,\ \underline{pf}_i^{\mathrm{dg}}-\cos(\tan^{-1}(Q_{i,t}^{\mathrm{dg}}/P_{i,t}^{\mathrm{dg}})). \tag{18}$$

The fifth term $C_{i,t}^{\mathrm{bss}}$ measures the violation of the BSS capacity constraint (6), which is calculated by:

$$C_{i,t}^{\mathrm{bss}}=[\max(0,E_{i,t}^{\mathrm{bss}}-\overline{E}_{i,t}^{\mathrm{bss}})+\max(0,\underline{E}_{i,t}^{\mathrm{bss}}-E_{i,t}^{\mathrm{bss}})]/\overline{E}_i. \tag{19}$$

5) Objective: We define $J(\pi)$ as the expected discounted return from time step 0 to T, which is calculated by

$$J(\pi)=\mathbb{E}_{\tau\sim\pi}\left[r_0+\gamma r_1+\cdots+\gamma^{T-1}r_T,\right]$$

where $\gamma \in [0,1]$ is the discount factor, τ denotes a trajectory ($\tau=(s_0,a_0,a_1,...,s_T)$), $\mathbb{E}_{\tau\sim\pi}[\cdot]$ is the expected value of the distribution over the trajectory τ, and $\tau\sim\pi$ is short for $s_0\sim\mu, a_t\sim\pi(\cdot|s_t), s_{t+1}\sim p(\cdot|s_t,a_t)$. The trajectory τ is a random process, in which the initial state s_0 follows the distribution μ, denoted by $s_0\sim\mu$; the action a_t follows the policy distribution $\pi(\cdot|s_t)$, denoted by $a_t\sim\pi(\cdot|s_t)$; the next state s_{t+1} follows the state transition probability distribution $p(\cdot|s_t,a_t)$, denoted by $s_{t+1}\sim p(\cdot|s_t,a_t)$. Since the CMDP formulation is model-free, the initial state distribution μ and the state transition probability distribution $p(\cdot|s_t,a_t)$ are unknown.

In addition, we define $J_C(\pi)$ as the expected discounted constraint violations (also denoted as C-return) from time step 0 to T, which is calculated by

$$J_C(\pi)=\mathbb{E}_{\tau\sim\pi}\left[c_0+\gamma c_1+\cdots+\gamma^{T-1}c_T\right].$$

For OODN, we aim to minimize the total operating cost over the horizon T without violating any safety constraints, so the problem can be formulated as

$$\begin{aligned} &\max_{\pi} \ J(\pi) \\ &s.t. \ J_C(\pi) \leq d \end{aligned} \tag{20}$$

where $d > 0$ is a tolerance parameter, which restricts the total constraint violation $J_C(\pi)$ to a very small number.

III. Safe Deep Reinforcement Learning Solution

In this section, we introduce a SDRL based solution to solve the CMDP. To handle the mixed discrete and continuous action space, we design a stochastic policy based on a multivariate joint distribution. Then, we adopt a neural network to learn the distribution parameters and train the neural network by CPO.

A. Constrained Policy Optimization Algorithm

For MDP problems, local policy search are usually used to find an optimal policy. Local policy search algorithms optimize a policy by iteratively searching for an improved one in a neighborhood of the most recent iterate π_k to maximize $J(\pi)$:

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi \in \Pi} \ & J(\pi) \\ s.t. \ & D(\pi, \pi_k) \leq \delta \end{aligned} \tag{21}$$

where D is a distance measure and δ defines the size of the neighborhood. A typical local policy search algorithm is trust region policy optimization [25], which uses the average KL-Divergence $\bar{D}_{KL}(\pi||\pi_k)[s] = \mathbb{E}_{s\sim\rho_{\pi_k}}[D_{KL}(\pi(\cdot|s)||\pi_k(\cdot|s))]$ to measure the searching area. Another one is the standard policy gradient, which uses the l-2 measure $D(\pi, \pi_k) = ||\theta - \theta_k||^2$ (policy π parameterized by θ) and maximizes a linearized objective $J(\pi_k) + \nabla_\theta J(\pi)(\theta - \theta_k)$ in the neighborhood of π_k.

For our CMDP problem, the searching area in each iteration is additionally confined by the constraint:

$$J_C(\pi) \leq d. \tag{22}$$

This makes local policy search algorithms difficult to implement because it requires evaluation of the constraint function $J_C(\pi)$ to determine whether a proposed policy is feasible.

To address this problem, CPO uses surrogate functions that are easy to evaluate from samples collected on π_k to approximate the constraint and the objective [23]. The surrogate functions are expressed by

$$\tilde{J}(\pi) = J(\pi_k) + \mathbb{E}_{\substack{s\sim\rho_{\pi_k} \\ a\sim\pi}}[A^{\pi_k}(s,a)] - \alpha_k \sqrt{\bar{D}_{KL}(\pi||\pi_k)} \tag{23}$$

$$\tilde{J}_C(\pi) = J_C(\pi_k) + \mathbb{E}_{\substack{s\sim\rho_{\pi_k} \\ a\sim\pi}}[A_C^{\pi_k}(s,a)] + \beta_k \sqrt{\bar{D}_{KL}(\pi||\pi_k)} \tag{24}$$

where $\alpha_k = \max_s |\mathbb{E}_{a\sim\pi}[A^{\pi_k}(s,a)]| \cdot \sqrt{2}\gamma/(1-\gamma)$, $\beta_k = \max_s |\mathbb{E}_{a\sim\pi}[A_C^{\pi_k}(s,a)]| \cdot \sqrt{2}\gamma/(1-\gamma)$, $A^{\pi_k}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ is the advantage function, and $A_C^{\pi_k}(s,a) = Q_C^{\pi}(s,a) - V_C^{\pi}(s)$ is the advantage functions with respect to the constraint.

According to [23], the surrogate functions satisfy the following properties:

$$\tilde{J}(\pi) \leq J(\pi), \ \tilde{J}_C(\pi) \geq J_C(\pi). \tag{25}$$

This means that $\tilde{J}$ is a lower bound of the objective and $\tilde{J}_C(\pi)$ is an upper bound of the constraint. If we replace the objective and the constraint with their surrogates and update the policy according to

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi \in \Pi} \ & \tilde{J}(\pi) \\ s.t. \ & \tilde{J}_C(\pi) \leq d, \end{aligned} \tag{26}$$

we can improve the worst-case performance and bound the worst-case constraint violation. This means that the policy update (26) can guarantee monotonic improvement in objective performance and constraint satisfaction.

One difficulty in applying the policy update (26) is the computation of the coefficients α_k and β_k because it involves solving the optimization $\max_s |\cdot|$. To solve this problem, CPO adopts a trust region constraint on the KL-Divergence instead of penalizing it by α_k and β_k. Consequently, the policy update (26) can be transformed into

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi \in \Pi} \ & \mathbb{E}_{\substack{s\sim\rho_{\pi_k} \\ a\sim\pi}}[A^{\pi_k}(s,a)] \\ s.t. \ & J_C(\pi_k) + \mathbb{E}_{\substack{s\sim\rho_{\pi_k} \\ a\sim\pi}}[A_C^{\pi_k}(s,a)] \leq d \\ & \bar{D}_{KL}(\pi||\pi_k) \leq \delta. \end{aligned} \tag{27}$$

Since the policy $\pi(a|s)$ is a function of s and a, we cannot directly optimize $\pi(a|s)$ using (27). Next, we will design a neural network to approximate the policy and optimize the neural network's weights to improve the policy.

B. Parameterized Policy for Discrete and Continuous Actions

In the OODN problem, SCBs, OLTCs, and VRs operate in discrete steps whereas dispatchable DGs and BSSs operate with continuous outputs. Therefore, the action space contains both discrete and continuous control variables:

$$a_t = (\underbrace{n_{i,t}^{\text{scb}}, l_{i,t}^{\text{vr}}}_{\text{discrete}}, \underbrace{P_{i,t}^{\text{dg}}, Q_{i,t}^{\text{dg}}, P_{i,t}^{\text{bss}}}_{\text{continuous}}), \forall i \in \Omega_n. \tag{28}$$

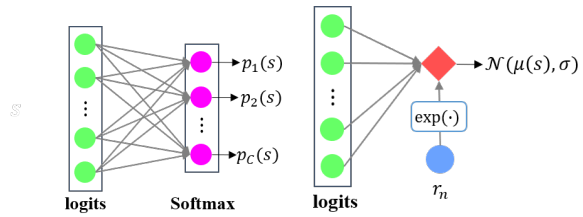
To deal with the mixed discrete and continuous action space, we approximate the policy by using a joint distribution:

$$\begin{aligned} &\pi(a_t|s_t) = \\ &\prod_{i\in\Omega_n} p(n_{i,t}^{\text{scb}}|s_t) p(l_{i,t}^{\text{vr}}|s_t) f(P_{i,t}^{\text{dg}}|s_t) f(Q_{i,t}^{\text{dg}}|s_t) f(P_{i,t}^{\text{bss}}|s_t), \end{aligned} \tag{29}$$

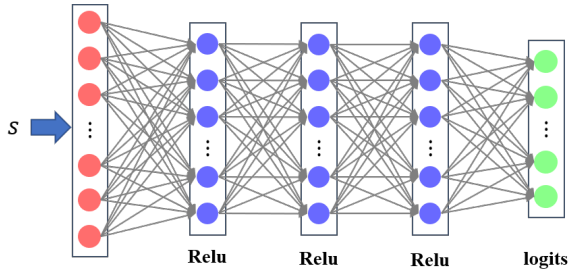
where $p(\cdot|s_t)$ is the probability mass function (PMF) of a categorical distribution; $f(\cdot|s_t)$ is the probability density function (PDF) of a normal distribution. Note that the actions $(n_{i,t}^{\text{scb}}, l_{i,t}^{\text{vr}}, P_{i,t}^{\text{dg}}, Q_{i,t}^{\text{dg}}, P_{i,t}^{\text{bss}})$ are considered as random variables and assumed to be independent from each other. In practice, this assumption generally holds because all actions are executed simultaneously at each time slot. The probability of each action being executed is dependent on only the system state s_t but not the observation of other actions.

For a discrete control variable x whose sample space has C individually identified items, the probability of x taking on the value c given $s_t = s$ is

$$p(x = c|s_t = s) = \prod_{c=1}^{C} p_c(s)^{[x=c]} \tag{30}$$



(a) Discrete action (b) Continuous action



(c) Feed-forward neural network.

Fig. 1. Designed policy network with mixed discrete and continuous actions.

where $p_c(s)$ represents the probability of seeing element c given s, and $[x = c]$ evaluates to 1 if $x = c$, 0 otherwise.

For a continuous control variable y, the conditional probability given $s_t = s$ is

$$f(y|s_t = s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{[y - \mu(s)]^2}{2\sigma^2}\right) \tag{31}$$

where $\mu(s)$ is the mean and σ is the standard deviation. Combining (30) and (31), the probability of an action a_t given $s_t = s$ can be calculated by (29). Also, we can generate discrete and continuous actions by sampling from the joint distribution (29).

The problem now becomes how to optimally determine the distribution parameters $p_c(s)$, $\mu(s)$ and σ such that the policy π solves the CMDP. Since the distribution parameters $p_c(s)$ and $\mu(s)$ depends on the state s, we use neural networks to learn these distribution parameters.

For discrete control variables, we evaluate the probabilities $(p_1(s), p_2(s), \ldots, p_C(s))$ by using a softmax function (Fig. 1(a):

$$p_c(s) = \frac{e^{z_c}}{e^{z_1} + e^{z_2} + \cdots + e^{z_C}}, \ \forall c \in \{1, ..., C\}, \tag{32}$$

where $z = [z_1, \ldots, z_C]^T$ is computed by

$$z = W_c \cdot logits(s) + b_c, \tag{33}$$

where W_c and b_c are trainable parameters.

For continuous control variables, we estimate the mean $\mu(s)$ and the standard deviation σ by the model (Fig. 1(b):

$$\begin{aligned} \mu(s) &= W_n \cdot logits(s) + b_n, \\ \sigma &= \exp(r_n), \end{aligned} \tag{34}$$

where W_n, b_n, and r_n are trainable parameters. The notation $logits(s)$ represents the feature vector extracted from s by the neural network (Fig. 1(c)).

C. Practical Implementation

Letting the vector θ denote all trainable parameters, we will use θ to represent the parameterized policy $\pi(a_t|s_t;\theta)$. We will replace the previous notations depending on π with the function of θ, e.g. $J(\theta) := J(\pi)$, $J_C(\theta) := J_C(\pi)$, $\rho_\pi := \rho_\theta$ and $\bar{D}_{KL}(\pi||\pi_k) := \bar{D}_{KL}(\theta||\theta_k)$. The parameters θ are then updated by solving the optimization problem (27).

To efficiently solve (27) in practice, we use a convex approximation of (27). Note that in a local neighborhood of θ_k, the expected advantage functions can be well approximated by

$$\mathbb{E}_{\substack{s\sim\rho_{\theta_k}\\ a\sim\theta}}\left[A^{\theta_k}(s,a)\right] \approx \mathbb{E}_{\substack{s\sim\rho_{\theta_k}\\ a\sim\theta_k}}[A^{\theta_k}(s,a)] + g^T(\theta - \theta_k) \tag{35}$$

$$\mathbb{E}_{\substack{s\sim\rho_{\theta_k}\\ a\sim\theta}}[A_C^{\theta_k}(s,a)] \approx \mathbb{E}_{\substack{s\sim\rho_{\theta_k}\\ a\sim\theta_k}}[A_C^{\theta_k}(s,a)] + b^T(\theta - \theta_k) \tag{36}$$

where g is the gradient $\nabla_\theta \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta}[A^{\theta_k}(s,a)]|_{\theta=\theta_k}$, and b is the gradient $\nabla_\theta \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta}[A_C^{\theta_k}(s,a)]|_{\theta=\theta_k}$. Also, the policy divergence can be well approximated by

$$\begin{aligned} \bar{D}_{KL}(\theta||\theta_k) =& \bar{D}_{KL}(\theta_k||\theta_k) + \nabla_\theta \bar{D}_{KL}(\theta||\theta_k)|_{\theta=\theta_k}(\theta - \theta_k) + \\ & \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \end{aligned} \tag{37}$$

where H is the hessian $\nabla^2_{\theta\theta}\bar{D}_{KL}(\theta||\theta_k)|_{\theta=\theta_k}$.

Since $\mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta_k}[A^{\theta_k}(s,a)] = \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta_k}[A_C^{\theta_k}(s,a)] = \bar{D}_{KL}(\theta_k||\theta_k) = \nabla_\theta \bar{D}_{KL}(\theta||\theta_k)|_{\theta=\theta_k} = 0$, the problem (27) is well approximated by

$$\begin{aligned} \theta_{k+1} = \arg\max_{\theta\in\Theta} \ & g^T(\theta - \theta_k) \\ s.t. \ & c + b^T(\theta - \theta_k) \le 0 \\ & \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \le \delta. \end{aligned} \tag{38}$$

where g is the gradient $\nabla_\theta \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta}[A^{\theta_k}(s,a)]|_{\theta=\theta_k}$, b is the gradient $\nabla_\theta \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\theta}[A_C^{\theta_k}(s,a)]|_{\theta=\theta_k}$, and $c = J_C(\theta_k) - d$. The problem is convex and has a closed-form solution:

$$\theta_{k+1} = -\frac{1}{\lambda^*} H^{-1}(g + \nu^* b) \tag{39}$$

where λ^* and ν^* are the optimal dual solutions.

In practice, we estimate the value of g, b, H, and c by using their sampling means. For the advantage functions $A^{\theta_k}(s,a)$ and $A_C^{\theta_k}(s,a)$, we use the generalized advantage estimation (GAE) [26]:

$$\begin{aligned} &\hat{A}^{\theta_k}(s_t = s, a_t = a) = \epsilon_t + (\gamma\lambda)\epsilon_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\epsilon_{T-1} \\ &\text{where } \epsilon_t = r_t + \gamma V^{\pi_k}(s_{t+1}) - V^{\pi_k}(s_t), \end{aligned} \tag{40}$$

$$\begin{aligned} &\hat{A}_C^{\theta_k}(s_t = s, a_t = a) = \epsilon_t^C + (\gamma\lambda)\epsilon_{t+1}^C + \cdots + (\gamma\lambda)^{T-t+1}\epsilon_{T-1}^C \\ &\text{where } \epsilon_t^C = c_t + \gamma V_C^{\pi_k}(s_{t+1}) - V_C^{\pi_k}(s_t), \end{aligned} \tag{41}$$

where λ is the GAE parameter. The value functions $V^{\pi_k}(s_t)$ and $V_C^{\pi_k}(s_t)$ are approximated by a neural network parameterized by ϕ, which is trained by minimizing a square-error loss:

$$L_\phi = \sum_t \left[(V^{\pi_k}(s_t;\phi) - V_t^{\text{targ}})^2 + (V_C^{\pi_k}(s_t;\phi) - V_{C,t}^{\text{targ}})^2\right], \tag{42}$$

where $V_t^{\text{targ}} = \sum_{l=t}^{T-1} \gamma^{l-t} r_l$ and $V_{C,t}^{\text{targ}} = \sum_{l=t}^{T-1} \gamma^{l-t} c_l$.

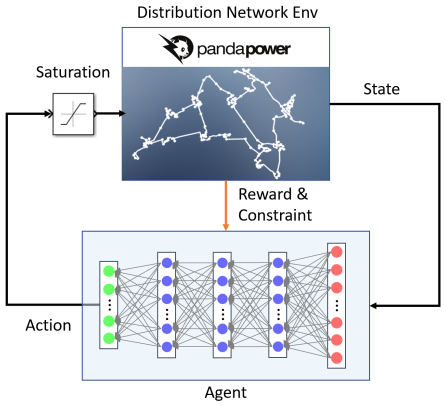


Fig. 2. Digraph of the overall training scheme. The saturation block limits out-of-range actions to their upper or lower bounds.

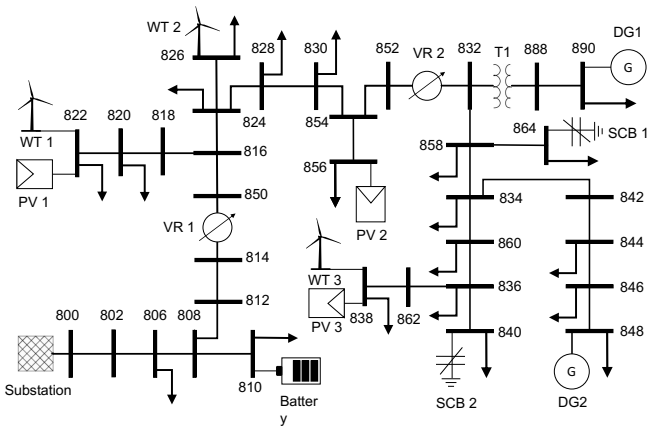


Fig. 3. Modified IEEE-34 node test feeder system [27].

In our implementation, we adopt parallel actors to accelerate the sampling process and improve exploration. In addition, to avoid calculation of the inverse of Hessian in the update formula (39), the conjugate gradient algorithm is used to directly compute the Hessian-vector products $H^{-1}x$. A backtracking line search method is used to find optimal dual solutions λ^* and ν^* [25]. The pseudocode of the CPO-based methods for the OODN problem is presented in Alg. 1.

Note that the continuous actions $P_{i,t}^{\text{dg}}$ and $P_{i,t}^{\text{bss}}$ are constrained by the upper and lower limits in (3) and (5). In our study, we do not formulate these constraints in the CMDP. Instead, we force an action to its feasible range if the action violates the corresponding upper or lower constraint. To this end, we assume that there is a signal saturation block installed before the input of the distribution network environment (Fig. 2). Hence, any action leading to a violation of the upper or lower constraints will be confined to the corresponding upper or lower saturation value.

IV. Case Studies

A. IEEE-34 Node System

In this subsection, we evaluate the performance of the proposed learning method on a modified IEEE-34 node test feeder system [27]. Fig. 3 shows the test system containing two VRs

Algorithm 1 CPO-based Learning Algorithm for OODN

Initialize network parameters θ_0, ϕ_0.
for $k = 1, 2, \dots$ **do**
  Initialize an empty set $\mathcal{D}$
  for $n = 1, 2, \dots, N$ **do in parallel**
    Sample an initial state $s_0 \sim \mu$
    for $t = 0, 1, \dots, T-1$ **do**
      Choose $a_t \sim \pi(\cdot|s_t; \theta_k)$ and do simulation
      Observe s_{t+1}, r_t, and c_t
    end for
    Store the trajectory $\tau = (s_0, a_0, r_0, c_0, s_1, \dots)$ in $\mathcal{D}$
  end for
  Use the sampled trajectories τ in $\mathcal{D}$ to calculate $\{\hat{A}^{\theta_k}(s_0, a_0), \dots, \hat{A}^{\theta_k}(s_T, a_T)\}$ according to (40)
  Use the sampled trajectories τ in $\mathcal{D}$ to calculate $\{\hat{A}_C^{\theta_k}(s_0, a_0), \dots, \hat{A}_C^{\theta_k}(s_T, a_T)\}$ according to (41)
  Use the sampled rewards $(r_0, \dots, r_T)$ in $\mathcal{D}$ to calculate $\{V_0^{\text{targ}}, \dots, V_T^{\text{targ}}\}$ according to $V_t^{\text{targ}} = \sum_{l=t}^{T-1} \gamma^{l-t} r_l$
  Use the samples $(c_0, \dots, c_T)$ in $\mathcal{D}$ to calculate $\{V_{C,0}^{\text{targ}}, \dots, V_{C,T}^{\text{targ}}\}$ according to $V_{C,t}^{\text{targ}} = \sum_{l=t}^{T-1} \gamma^{l-t} c_l$
  Estimate g, b, H and c using their sampling means
  Update $\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi L_\phi|_{\phi=\phi_k}$ using (42)
  Update θ_{k+1} by (39)
end for

with 33 tap positions $(-16; -15; \dots; 0; +1; +2; \dots; +16)$ and a regulation range of -10% to $+10\%$ (0.625% per tap); two SCBs of 0.48 MVAR with four units (0.12 MVAR/unit) at nodes 864 and 840; two dispatchable DGs with capacities of 0.825 MVA and 0.625 MVA and a minimum power factors of 0.8 at node 848 and 890, respectively; one BSS at node 810 with a capacity of 2 MWh and a maximum charging/discharging power of 0.5 MW; three photovoltaic RSs with power peaks of 0.1 MW at nodes 822, 856, and 838, respectively; three wind RSs with power peaks of 0.1 MW at nodes 822, 826, and 838, respectively. The objective is to minimize the total cost of energy purchased from the substation and the dispatchable DGs. The capacity of the substation is 2.5 MVA. The nodal voltages are bounded within 0.95 p.u. - 1.05 p.u. It is also assumed that the BSS has an efficiency of $\eta_{i,ch}^{\text{bss}} = \eta_{i,dch}^{\text{bss}} = 0.98$, and an allowable minimum energy of 0.2 MWh.

The electricity prices, load demand, and RSs generation power are simulated by using the time-series data in California Independent System Operator (CAISO) [28]. We downloaded a 3 year period of data, from 2018 to 2020, and used the first two years (2018-2019) of the data as the training set and the last year (2020) as the test set. The load demand data are scaled to a proper level according to the considered system. Specifically, the load data are first normalized to unity and then scaled up by multiplying a base power. The base power varies from node to node, and the nodal load data of the standard IEEE 34-node system [27] are used as the base power. The data on the photovoltaic and wind RSs are processed using the same method. The cost coefficients of the dispatchable DGs are $a_1^{\text{dg}} = 100\$/\text{MWh}^2$, $b_1^{\text{dg}} = 72.4\$/\text{MWh}$, $c_1^{\text{dg}} = 0.5\$/\text{h}$ for DG1; $a_2^{\text{dg}} = 100\$/\text{MWh}^2$, $b_2^{\text{dg}} = 51.6\$/\text{MWh}$, $c_2^{\text{dg}} = 0.46\$/\text{h}$

TABLE I
PARAMETER SETTINGS OF THE PROPOSED METHOD.

Parameter	Value
Constraint tolerance (d)	1e-3
GAE parameter (λ)	0.95
Discount factor(γ)	0.995
KL-Divergence Limit (δ)	0.02
Stepsize of value network update (α)	3e-4
Number of steps in one episode(T)	24
Number of total episodes	1M

TABLE II
AVERAGE TIME CONSUMPTION OF DIFFERENT METHODS ON TRAINING AND ONLINE COMPUTATION (ONE-STEP) FOR IEEE-34 NODE SYSTEM.

Method	DDPG	PPO	SAC*	CPO	MISOCP
Training (h)	15.82	12.2	56.7	15.29	-
Online Comp. (s)	0.0013	0.0012	0.0013	0.0013	280.1

*SAC is implemented using Stable Baselines, which does not provide multi-processing implementation.

for DG2. The scheduling horizon is $T = 24$h.

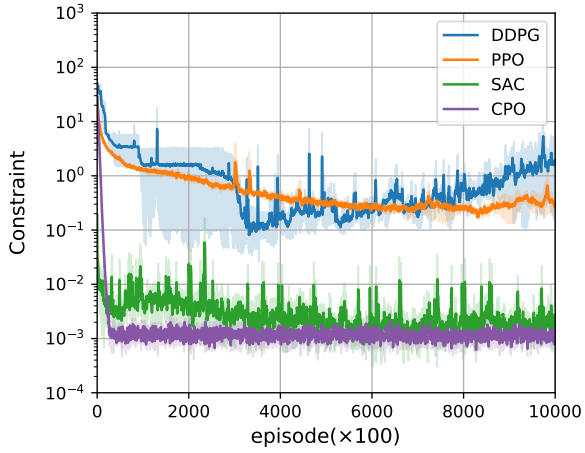
The parameter settings for the proposed method are given in Table I. The policy network has three layers of 128 ReLU neurons and the size of the logits is 64. The architecture of the value network is the same as that of the policy network. The network parameters θ and θ^v are orthogonally initialized. The simulation is carried out on a workstation with an Intel Core i7-7800X Processor 3.50GHz. The operation system is Ubuntu 20.04.3 LTS. The code is written in Python 3.7.6 using the deep learning package TensorFlow 2.2.0 [29], and DRL packages OpenAI Gym [30] and Baselines-tf2 [31].

1) Training Performance: The proposed method (CPO) is compared with several the state-of-the-art DRL-based approaches, including deep deterministic policy gradient (DDPG) [32], proximal policy optimization (PPO) [33], and soft actor critic (SAC) [34]. Since DDPG and SAC can only handle continuous actions, we round the control decisions to the nearest integer. For PPO, the mixed joint distribution policy proposed in our method is adopted to handle discrete and continuous actions. For these methods to confine the constraints, a penalty term is added to the reward, e.g.

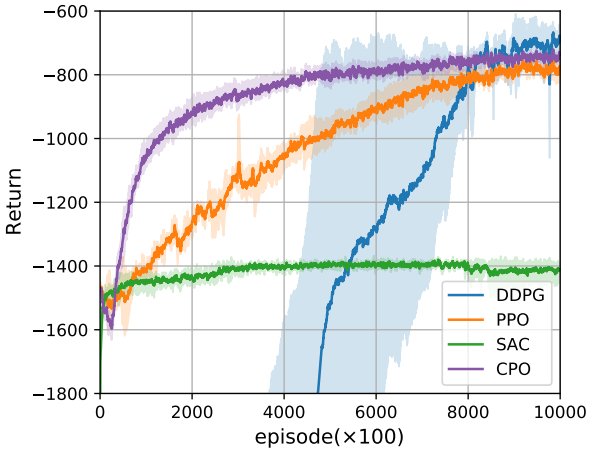
$$r_t := r_t + \varrho \cdot c_t \tag{43}$$

where ϱ is the penalty coefficient, which is set to 500. We run each method for 5 times with different random seeds. The average time consumption of different methods on training and online computation (one-step) is presented in Table II.

Fig. 4 compares the training performance of the proposed CPO and the state-of-the-arts DRL methods over 5 independent runs with different random seeds. From Fig. 4(a) we can observe that the constraint value of CPO (the purple line) decreases quickly to a level at $1e-3$, which is an acceptable tolerance defined in the parameter d. After that, the constraint value remains at this level whereas the reward starts to increase steadily, as shown in Fig. 4(b). This means that CPO learned a safe operation strategy for the distribution network in the first place, and then consistently improved it without undermining



(a) Constraint

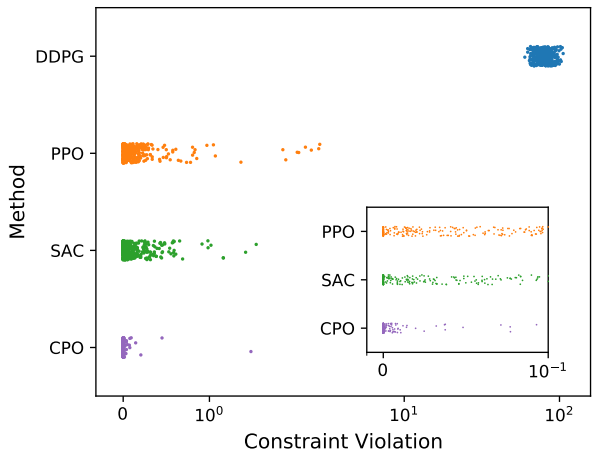


(b) Reward

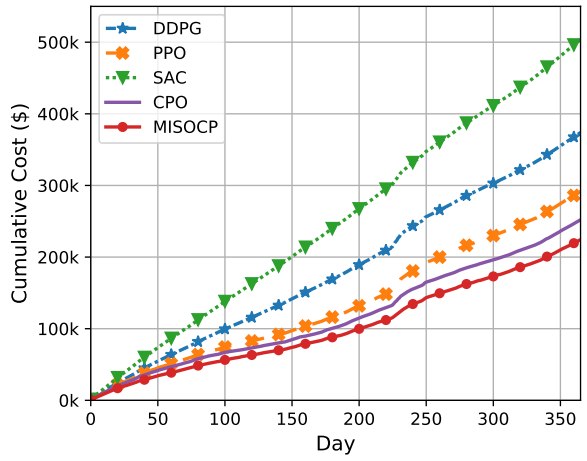
Fig. 4. Comparison of average performance over 5 runs on the modified IEEE-34 node system using different learning algorithms during the training process: a) constraint value, and b) return.

the safety of the operation. This makes the proposed method more practical than the other DRL based approaches. As we can observe from Fig. 4(a), DDPG and PPO failed to learn a safe operation strategy because of the large constraint value. Besides, even though SAC outperforms DDPG and PPO by restricting the constraint to a very small level, mostly below $1e-2$, the performance is not uniformly convergent on the entire training episodes. Also, as shown in Fig. 4(b), the reward of SAC does not improve much during the training process. This may result from the reason that the constraint is over-penalized.

2) Test Performance: After training, the well-trained model is tested on the test set. To verify the optimality of the proposed approach, we compare it with a model-based method, wherein the OODN problem is formulated as a mixed integer second-order cone programming (MISOCP) using DistFlow model [35]. We follow the method in [5] to build the MISOCP model. We assume that all uncertainties can be accurately predicted in the MISOCP formulation. To solve the MISOCP model, the optimization toolbox PySCIPOpt [36] is used. Fig.



(a) Constraint



(b) cumulative cost

Fig. 5. Performance of different algorithms for IEEE 34 node system on the test dataset (366 testing days): a) distribution of constraint violation, and b) cumulative cost.

5 compares the testing results obtained by the proposed and the benchmark methods. In the comparison, the constraint value is calculated by $\sum_{t=0}^{T-1} c_t$ and operational cost is calculated by $-\sum_{t=0}^{T-1} r_t$. As shown in this figure, CPO outperforms the state-of-the-art DRL methods by achieving the lowest cost and the least constraint violation. Specifically, in Fig. 5(a) we can observe that for the CPO algorithm, there are only a few cases of constraint violation on the whole-year testing data. It is worth noting that it is impossible to 100% guarantee the safety of hourly-ahead operation in any situation due to the existence of uncertainty. However, CPO can learn to safely operate the distribution system in most situations and guarantee near-constraint satisfaction. Although PPO and SAC can also confine the constraint violations to some extend, the performance varies largely one different days. Fig. 5(b) shows the cumulative operational cost on the 366 testing days. The total operating costs for DDGP, PPO, SAC, and CPO are \$373.95K, \$292.01K, \$503.21K, and \$252.06K, respectively. Compared to DDGP, PPO, and SAC, CPO reduces the cost by 32.5%, 13.6%, and 99.6%, respectively. It is notable that although DDPG obtains a lower cost than SAC, it causes serious violations of the operating constraints, which makes it impossible to implement in real distribution networks. The total operating cost of MISOCP is \$224.30K. Compared to CPO, the MISOCP method only reduces the operating cost by 11.01% even though it uses perfect forecast information of the uncertainty. It is worth mentioning that the performance of MISOCP is ideal and cannot be achieved in practice. These results verify the effectiveness of the proposed CPO-based method against uncertainty in the operation of distribution networks.

Fig. 6 shows the operating results obtained by CPO on 3 consecutive days in the test dataset. It can be observed from this figure that the BSS, the DGS as well as the VVC devices are properly scheduled in an efficient and cost-effective manner and the operating constraints of the distribution system are strictly satisfied. For instance, in Fig. 6 (c) we can see that, the BSS is scheduled to charge at low-price hours to store energy and discharge to supply power load at high-price hours. In Fig. 6 (d)-(e) we can observe that, the DGs are dispatched to reduce power generation when the price of electricity decrease and increase generation otherwise. Besides, the minimum power factors of the DGs are strictly constrained to be above 0.8. In Fig. 6 (f)-(h), the SCBs and VRs are controlled to confine the maximum and minimum nodal voltages to be within $[0.95, 1.05]$ to avoid under/over-voltage problems.

B. IEEE-123 Node System

In this subsection, we evaluate the performance of the proposed method on a modified IEEE-123 node system [37]. Fig. 7 shows the system containing two OLTCs with 5 tap positions $(-2; -1; 0; +1; +2)$ and a regulation range of -10% to $+10\%$ (2.5% per tap); two VRs with 33 tap positions $(-16; -15; \ldots; 0; +1; +2; \ldots; +16)$ and a regulation range of -10% to $+10\%$ (0.625% per tap); two SCBs of 1.2 MVAR with four units (0.3 MVAR/unit) at nodes 108 and 76; three dispatchable DGs with capacities of 0.825 MVA, 0.625 MVA, and 0.625 MVA and a minimum power factors of 0.8 at node 24, 94, and 114, respectively; two BSSs both with a capacity of 2 MWh and a maximum charging/discharging power of 0.5 MW at nodes 20 and 56, respectively; five photovoltaic RSs with power peaks of 0.1 MW at nodes 22, 250, 41, 450, and 39, respectively; five wind RSs with power peaks of 0.1 MW at nodes 4, 59, 46, 75, and 83, respectively. The capacity of the substation is 5 MVA. The nodal voltages are bounded within 0.95 p.u. - 1.05 p.u. It is also assumed that the BSS has an efficiency of $\eta^{\text{bss}}_{i,ch} = \eta^{\text{bss}}_{i,dch} = 0.98$, and an allowable minimum energy of 0.2 MWh. The cost coefficients of the dispatchable DGs are $a^{\text{dg}}_1 = 100\$/\text{MWh}^2$, $b^{\text{dg}}_1 = 72.4\$/\text{MWh}$, $c^{\text{dg}}_1 = 0.5\$/\text{h}$ for DG1; $a^{\text{dg}}_2 = a^{\text{dg}}_3 = 100\$/\text{MWh}^2$, $b^{\text{dg}}_2 = b^{\text{dg}}_3 = 51.6\$/\text{MWh}$, $c^{\text{dg}}_2 = c^{\text{dg}}_3 = 0.46\$/\text{h}$ for DG2 and DG3.

For the CPO algorithm, we use the same parameter settings as presented in Table I. The training and test datasets are the same as those used in the IEEE-34 node system. The policy and value network have three layers of 256 ReLU neurons and the size of the logits is 128. The simulation is conducted on the same workstation as the one used in IEEE-34 node

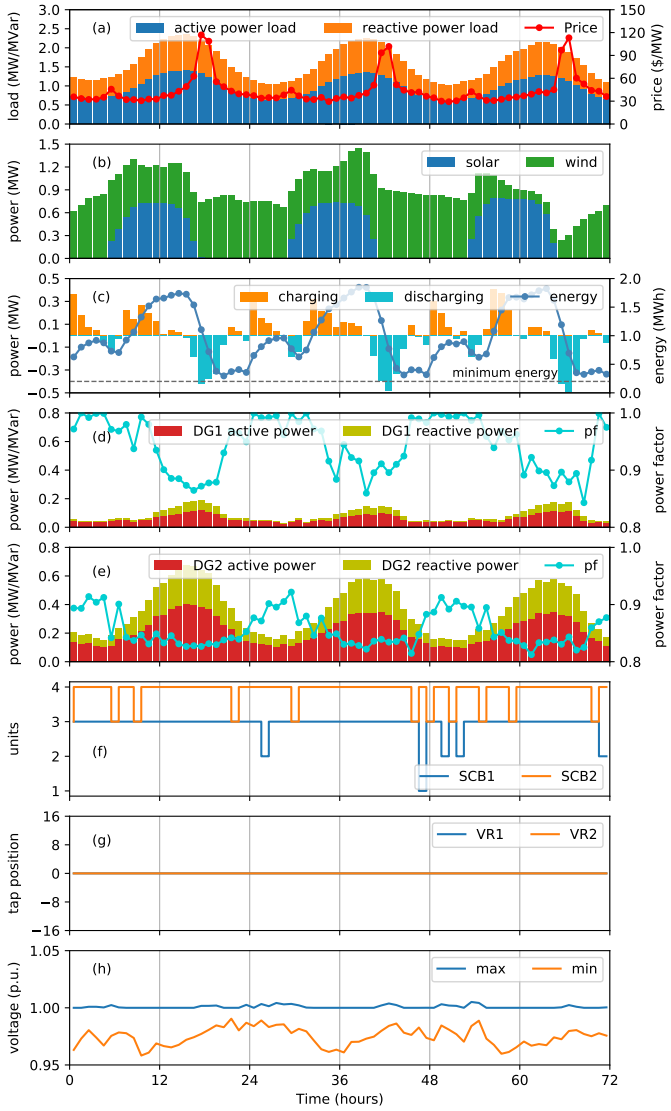


Fig. 6. Operating results of CPO for IEEE-34 node system on 3 consecutive testing days. (a)system load and electricity price. (b) solar and wind power generation. (c) charging/discharging power and energy status of the BSS. (d) active and reactive power generation of DG1. (e) active and reactive power generation of DG2. (f) units in operation of SCB 1 and SCB 2. (g) tap positions of VR 1 and VR 2. (h) maximum and minimum nodal voltages.

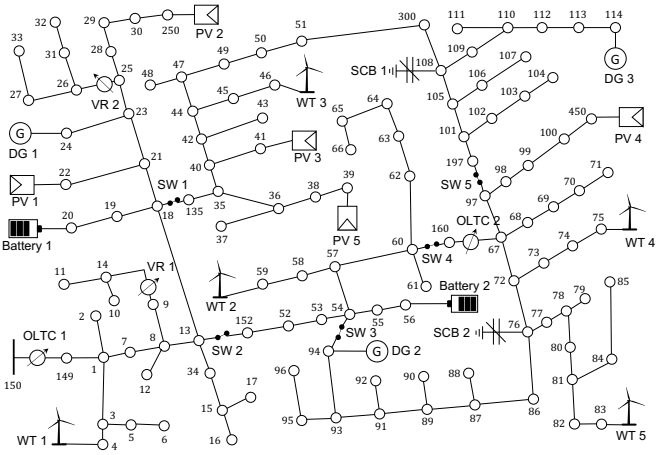
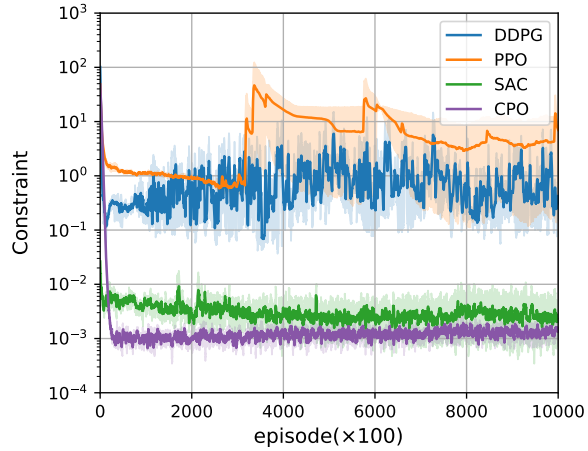
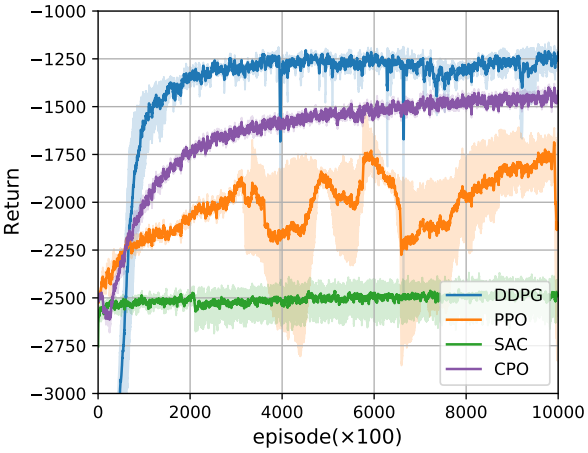


Fig. 7. Modified IEEE-123 node test feeder system [37].



(a) Constraint



(b) Return

Fig. 8. Comparison of average performance over 5 runs on the modified IEEE-123 node system using different learning algorithms during the training process: a) constraint value, and b) return.

system. The average time consumption of different methods on training and online computation (one-step) is presented in Table III.

1) Training Performance: Fig. 8 compares the training performance of CPO and other DRL methods over 5 independent runs with different random seeds. From Fig. 8(a) we can observe that CPO successfully learned a safe policy to restrict the constraint value to $1e-3$ in less than 20k episodes. However, the benchmark DRL methods, especially DDPG and PPO, failed to do so and led to large constraint violations. In addition, from Fig. 8 (b) we can observe that DDPG and CPO obtain higher returns during the training than PPO and SAC do. Compared to DDPG, however, CPO can learn to improve policy without violating operating constraints, which makes it more practical to be trained in real distribution networks. Combining the training performance in Fig. 4 we can conclude, CPO is more stable and effective in learning a good policy through safe exploration than the state-of-the-art DRL algorithms.

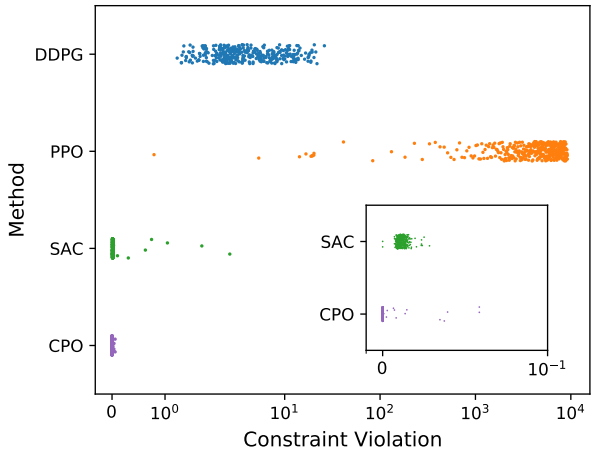
TABLE III
AVERAGE TIME CONSUMPTION OF DIFFERENT METHODS ON TRAINING AND ONLINE COMPUTATION (ONE-STEP) FOR IEEE-123 NODE SYSTEM.

Method	DDPG	PPO	SAC*	CPO	MISOCP
Training (h)	32.67	24.75	236.5	31.38	-
Online Comp.(s)	0.0028	0.0027	0.0028	0.0028	1182.1

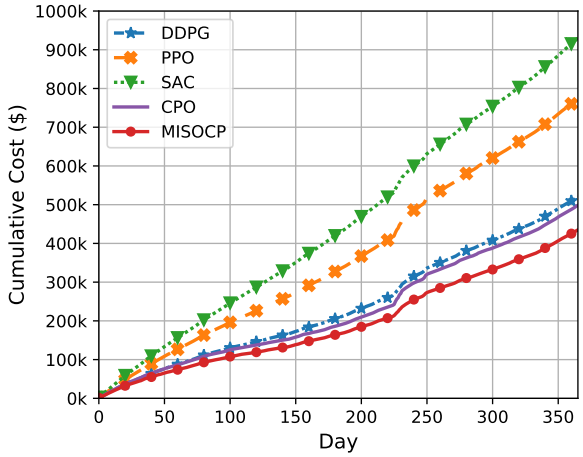
*SAC is implemented using Stable Baselines, which does not provide multi-processing implementation.

2) Test Performance: Fig. 9 compares the testing results of CPO and the benchmark methods on IEEE-123 node system. In the comparison we can observe that CPO outperforms the other DRL methods in terms of operating cost reduction and constraint satisfaction. As shown in Fig. 9 (a), CPO results in almost no constraint violation except a few testing days. DDPG and PPO, however, causes serious constraint violation on most of the testing days and is infeasible to implement in practice. Besides, although SAC shows comparable performance to CPO in handling constraints, it lead too very high operating costs as shown in Fig. 9 (b). The total operating cost for DDGP, PPO, SAC, and CPO are \$520.09K, \$773.85K, \$930.59K, and \$498.33K, respectively. Compared to DDGP, PPO, and SAC, CPO reduces the operational cost by 4.1%, 35.6%, and 46.4%, respectively. Besides, the total operating cost of MISOCP is \$434.87K, which is only 12.7% lower than that of CPO.

Fig. 10 shows the operating results of CPO on 3 consecutive testing days on the IEEE-123 node system. It can be observed from the subfigures that although there are many heterogeneous controllable devices, the learned policy can still efficiently coordinate these devices to safely and economically operate the distribution system. Specifically, in Fig. 10 (c)-(d), the BSS1 and BSS2 are scheduled to charge at off-peak hours and then discharge at peak hours. In Fig. 10 (e)-(g), the DGs are also appropriately operated to satisfy the minimum power factor requirement (above 0.8) and economically dispatched based on the price of electricity. Furthermore, in Fig. 10 (h)-(j) we can observe that, the SCBs and VRs/OLTCs are properly controlled to compensate reactive power and regulate nodal voltages within the range $[0.95, 1.05]$.



(a) Constraint



(b) cumulative cost

Fig. 9. Performance of different algorithms for IEEE 123 node system on the test dataset (366 testing days): a) distribution of constraint violation, and b) cumulative cost.

V. CONCLUSION

In this paper, we proposed a SDRL approach based on CPO for the OODN problem. The proposed approach enables an agent to learn a cost-effective operating strategy through safely exploring scheduling actions. Compared to traditional DRL methods, the proposed approach is more practical to be trained in a real distribution system. Besides, the proposed approach is suitable for training complex policies with a mixed discrete and continuous action space. The proposed approach is totally data-driven and does not rely on any physical model, statistical model, or mathematical programming optimizer. The learned policy is based on neural networks and can directly generate scheduling decision to minimize the operational cost and confine operating constraints simultaneously. Simulation studies on IEEE-34 and IEEE-123 bus systems using real-world power system data demonstrate that the proposed approach can successfully learn an effective policy to operate distribution networks in a safe and cost-efficient way. Comparison results verify the superiority of the proposed method over the state-of-the-art DRL approaches in terms of constraint satisfaction and cost reduction.

REFERENCES

[1] G. J. Peponis, M. P. Papadopulos, and N. D. Hatziargyriou, "Optimal operation of distribution networks," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 59–67, 1996.

[2] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, "A multi-agent deep reinforcement learning based voltage regulation using coordinated pv inverters," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 4120–4123, 2020.

[3] T. A. Short, *Electric Power Distribution Handbook*. Florida: CRC Press LLC, 2004.

[4] F. Capitanescu, I. Bilibin, and E. Romero Ramos, "A comprehensive centralized approach for voltage constraints management in active distribution grid," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 933–942, 2014.

[5] L. H. Macedo, J. F. Franco, M. J. Rider, and R. Romero, "Optimal operation of distribution networks considering energy storage devices," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2825–2836, 2015.

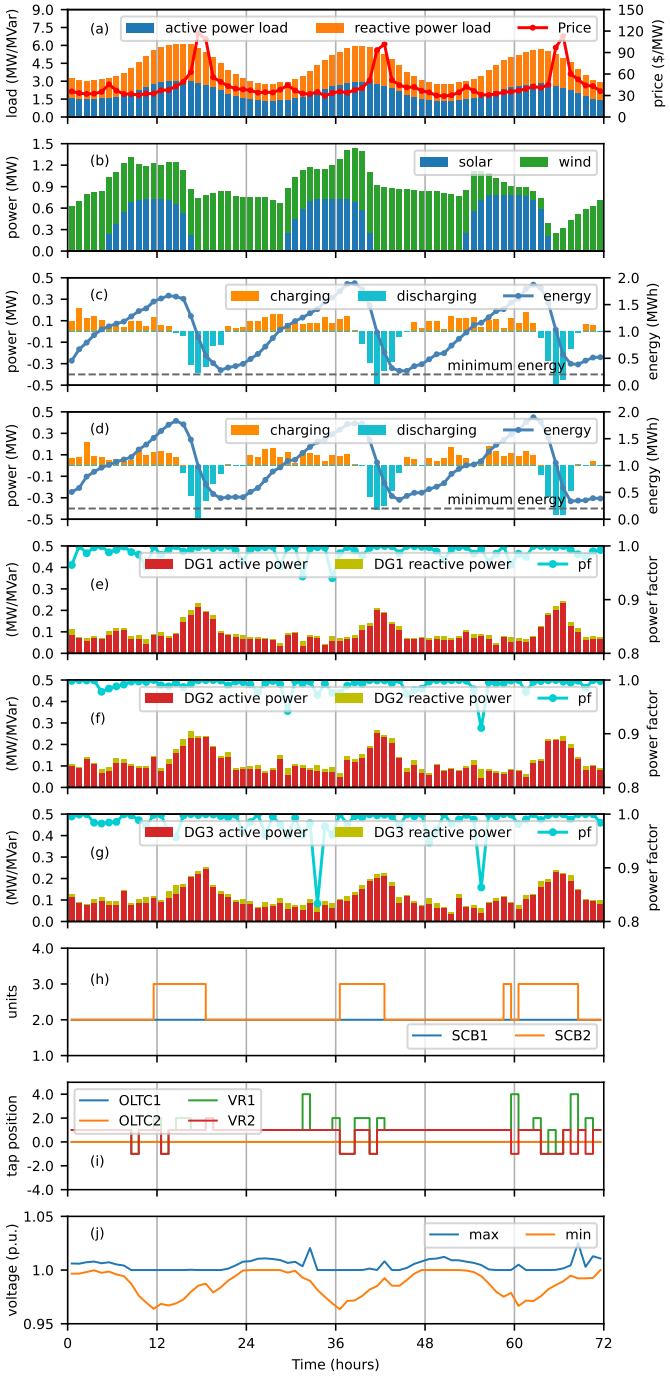


Fig. 10. Operating results of CPO for IEEE-123 node system on 3 consecutive testing days. (a) system load and electricity price. (b) solar and wind power generation. (c) charging/discharging power and energy status of BSS1. (d) charging/discharging power and energy status of BSS2. (e) active and reactive power generation of DG1. (f) active and reactive power generation of DG2. (g) active and reactive power generation of DG3. (g) units in operation of SCB 1 and SCB 2. (i) tap positions of VR1, VR2, OLTC1, OLTC2. (j) maximum and minimum nodal voltages.

[6] Y. Xu, Z. Y. Dong, R. Zhang, and D. J. Hill, "Multi-timescale coordinated voltage/var control of high renewable-penetrated distribution systems," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4398–4408, 2017.

[7] T. Ding, S. Liu, W. Yuan, Z. Bie, and B. Zeng, "A two-stage robust reactive power optimization considering uncertain wind power integration in active distribution networks," vol. 7, no. 1, pp. 301–311, 2016.

[8] H. Gao, L. Wang, J. Liu, and Z. Wei, "Integrated day-ahead scheduling considering active management in future smart distribution system," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6049–6061, 2018.

[9] W. Huang, W. Zheng, and D. J. Hill, "Distributionally robust optimal power flow in multi-microgrids with decomposition and guaranteed convergence," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 43–55, 2021.

[10] W. Zheng, W. Huang, and D. J. Hill, "A deep learning-based general robust method for network reconfiguration in three-phase unbalanced active distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 120, p. 105982, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519328091

[11] W. Zheng, W. Huang, D. J. Hill, and Y. Hou, "An adaptive distributionally robust model for three-phase distribution network reconfiguration," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1224–1237, 2021.

[12] J. G. Vlachogiannis and N. D. Hatziargyriou, "Reinforcement learning for reactive power control," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1317–1325, 2004.

[13] Y. Xu, W. Zhang, W. Liu, and F. Ferrese, "Multiagent-based reinforcement learning for optimal reactive power dispatch," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 6, pp. 1742–1751, 2012.

[14] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1990–2001, 2020.

[15] E. Foruzan, L. Soh, and S. Asgarpoor, "Reinforcement learning approach for optimal distributed energy management in a microgrid," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5749–5758, 2018.

[16] H. Shuai and H. He, "Online scheduling of a residential microgrid via monte-carlo tree search and a learned model," *IEEE Trans. Smart Grid*, pp. 1–1, 2020.

[17] Y. Ji, J. Wang, J. Xu, and D. Li, "Data-driven online energy scheduling of a microgrid based on deep reinforcement learning," *Energies*, vol. 14, no. 2120, 2021.

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

[19] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2313–2323, 2020.

[20] W. Wang, N. Yu, Y. Gao, and J. Shi, "Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3008–3018, 2020.

[21] Y. Zhang, X. Wang, J. Wang, and Y. Zhang, "Deep reinforcement learning based volt-var optimization in smart distribution systems," *IEEE Trans. Smart Grid*, pp. 1–1, 2020.

[22] H. Li, Z. Wang, and H. He, "Distributed volt-var optimization based on multi-agent deep reinforcement learning," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–7.

[23] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th ICML*, Sydney, NSW, Australia, 2017, p. 22–31.

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

[25] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 34th ICML*, Lille, France, 2015, pp. 1889–1897.

[26] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *ArXiv pre-prints*, 2018, arXiv:1506.02438.

[27] *IEEE 34-bus Feeder*, IEEE PES AMPS DSAS Test Feeder Working Group, 2020. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/feeder34.zip

[28] *OASIS*, california ISO open access same-time information system, 2021. [Online]. Available: http://oasis.caiso.com/mrioasis/logon.do

[29] *Tensorflow*. (2.2.0). [Online]. Available: https://www.tensorflow.org/

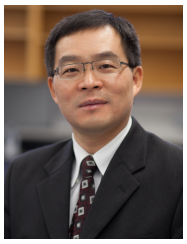
[30] "*OpenAI Gym*. (0.19.0)." [Online]. Available: https://gym.openai.com

[31] "*Baselines*. (0.1.5)." [Online]. Available: https://github.com/openai/baselines/tree/tf2

[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019, arXiv:1509.02971.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, arXiv:1707.06347.
[34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, arXiv:1801.01290.
[35] R. Cespedes, "New method for the analysis of distribution networks," *IEEE Trans. Power Del.*, vol. 5, no. 1, pp. 391–396, 1990.
[36] S. Maher, M. Miltenberger, J. P. Pedroso, D. Rehfeldt, R. Schwarz, and F. Serrano, "PySCIPOpt: Mathematical programming in python with the SCIP optimization suite," in *Mathematical Software – ICMS 2016*. Springer International Publishing, 2016, pp. 301–307.
[37] *IEEE 123-bus Feeder*, IEEE PES AMPS DSAS Test Feeder Working Group, 2020. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/feeder123.zip

Hepeng Li (S'19) received his B.S. degree in information and computing science and the M.S. degree in control theory and control engineering from the Northeastern University, China, in 2009 and 2012, respectively. He is currently pursuing his Ph.D. degree with the School of Electrical, Computer and Biomedical Engineering, University of Rhode Island, South Kingstown, RI, USA. His research interests include smart gird, demand response, and deep reinforcement learning, safe reinforcement learning.



Haibo He (SM'11-F'18) is the Robert Haas Endowed Chair Professor in Electrical Engineering at the University of Rhode Island, Kingston, RI, USA. His research interests include computational intelligence and its applications. He served/serves numerous capacities at the IEEE Computational Intelligence Society (IEEE CIS), including the Chair of IEEE CIS Neural Networks Technical Committee (NNTC), IEEE CIS Conference Committee, General Chair of IEEE Symposium Series on Computational Intelligence (IEEE SSCI'14, Orlando, Florida), among others. He was a recipient of the "IEEE CIS Outstanding Early Career Award," National Science Foundation "Faculty Early Career Development (CAREER) Award," among others. He served as the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems from 2016 to 2021.