Data, Parameters, and Training Procedure

Xiaosheng Zhang, *Student Member, IEEE*, Tao Ding, *Senior Member, IEEE*, Chenggang Mu, *Student Member, IEEE*, Biyuan Zhang, *Student Member, IEEE*, Yuankang He, Mohammad Shahidehpour, *Fellow, IEEE*

The simplified topology of a regional power grid in China is shown in Fig. 1. This system is a high-penetrated renewable power system. The capacity of conventional thermal plants, hydro plants, and RES plants is shown in Table I. The scenario tree is constructed by using the Latin Hypercube Sampling method. Fig. 2 plots the forecasted RES output and gross load profiles of the regional power grid. The maximum and minimum load demands are 95,000 MW and 83,500 MW.

The parameters of the adding VESs are listed in Table II.

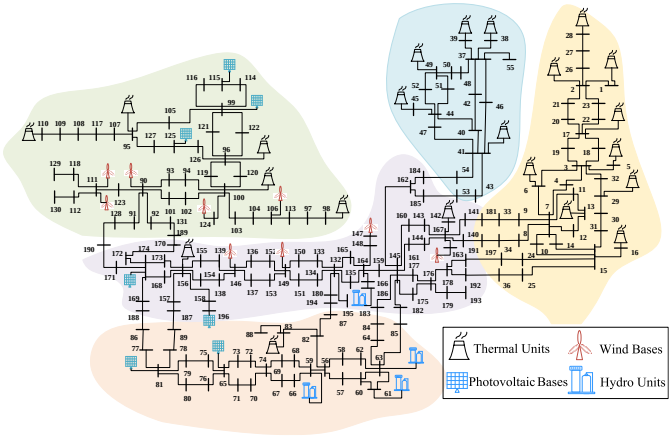

Fig. 1 Simplified topology of a regional power grid in China.

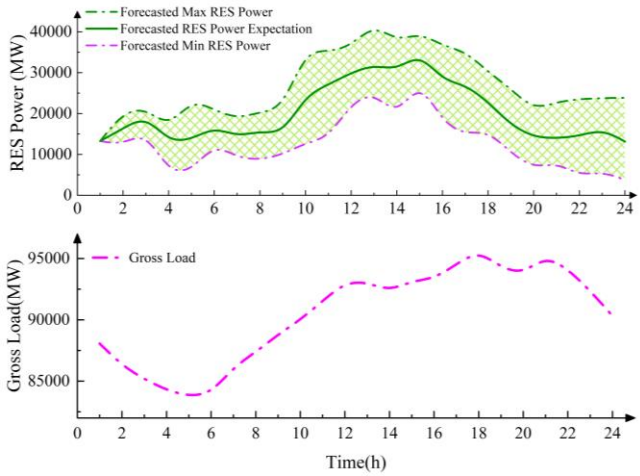

Fig. 2 (a) The forecasted RES output and (b) Gross load of the regional power grid

TABLE I. CAPACITY OF DIFFERENT TYPES OF UNITS (MW)

Units	Thermal	Hydro	RES	Total
Capacity(MW)	102643	31323	60738	194704

TABLE II PARAMETERS OF THE VES

g	Bus	$P^{G}_{j,\max}/P^{G}_{j,\min}$ (MW)	$\Delta_j^{\text{up}}/\Delta_j^{\text{down}}$
1	5	1200/600	300
2	16	2400/1200	600
3	24	2400/1200	600
4	39	1200/600	300
5	48	1920/960	480
6	140	2400/1200	600
7	168	900/450	225
8	169	2400/1200	600

A. Training Procedure

In the related literature, the above problem is usually solved by gradient-descend type algorithms, while the parameter θ_τ is regarded as the variable. The advantage of gradient-descend type algorithms is that they do not require any extra information in addition to the input and output values of the function. In our problem, $\mathcal{V}_t^\pi(a_{t-1})$ is convex, and the sub-gradient of $\mathcal{V}_t^\pi(a_{t-1})$ is derived by solving the linear programming. Thus, the problem can be solved more efficiently with the above information. By regarding $\theta(\{\theta_\tau|\tau=1,\ldots, N_\tau\})$ as constant, the optimality condition of (37) in the paper is stated as,

$$\nabla J = \frac{1}{N_\tau}\sum_{\tau=1}^{N_\tau}\frac{2(\mathcal{V}_t^\pi(a_{t-1}^\tau)-\mathfrak{V}_t^\pi(a_{t-1}^\tau;\theta_\tau))\cdot}{(\nabla\mathcal{V}_t^\pi(a_{t-1}^\tau)-\nabla\mathfrak{V}_t^\pi(a_{t-1}^\tau;\theta_\tau))}=0 \quad (1)$$

Due to the convexity of $\mathcal{V}_t^\pi(a_{t-1})$, we have

$$\mathcal{V}_t^\pi(a_{t-1}) \geq \mathcal{V}_t^\pi(a_{t-1}^\tau)+\nabla\mathcal{V}_t^\pi(a_{t-1}^\tau)(a_{t-1}-a_{t-1}^\tau) \quad (2)$$

Combining (1) and (2), $\mathfrak{V}_t^\pi(a_{t-1};\ \theta)$ is stated as follows,

$$\mathfrak{V}_t^\pi(a_{t-1};\theta)=\bigcap_{\tau=1}^{N_\tau}\mathfrak{V}_t^\pi(a_{t-1}^\tau;\theta_\tau) \quad (3)$$

$$\mathfrak{V}_t^\pi(a_{t-1}^\tau;\theta_\tau)=\min\{\eta_t:\eta_t\geq\overline{g}_t^\tau-\overline{\mu}_t^{\tau\top}x_t\} \quad (4)$$

where $\theta_\tau=\{\overline{g}_t^\tau,\overline{\mu}_t^\tau\};\overline{g}_t^\tau=\mathcal{V}_t^\pi(a_{t-1}^\tau)-\nabla\mathcal{V}_t^\pi(a_{t-1}^\tau)a_{t-1}^\tau$ is the intercept of the cut and $\overline{\mu}_t^{\tau\top}=-\nabla\mathcal{V}_t^\pi(a_{t-1}^\tau)a_{t-1}$ is the gradient of the cut. SDDPIL is a model-based reinforcement learning algorithm since the environment model $p(s_{t+1}|s_t,\ a_t)$ is known as $p(\xi_{t+1}|\xi_t)$. It is a double-pass reinforcement learning algorithm consisting of a forward pass and a backward pass. Accordingly, the detailed training procedure is stated as follows.

1) Forward Pass

The forward pass will solve the Bellman optimality equation by outer approximating the expected cost-to-go function $\mathcal{V}_{t+1}^\pi(a_t)$ with piecewise linear functions at each stage. The dynamic programming equation for $t=1, 2,\ldots, T$ is stated as follows

$$V_t^\pi(s_t)=\min_{a_t\in\mathcal{A}(s_t)} r_t(s_t,a_t)+\mathfrak{V}_{t+1}^\pi(a_t) \quad (5)$$

where $\mathfrak{V}_{t+1}^{\pi}$ is the piecewise linear approximations of $\mathcal{V}_{t+1}^{\pi}(a_t)$. The detailed formulation is stated as follows,

$$\mathfrak{V}_{t+1}^{\pi}(a_t) \coloneqq \min\{\eta_{t+1} : \eta_{t+1} \geq \overline{g}_{t+1}^{e} - \overline{\mu}_{t+1}^{e\top} x_t,\ \forall e \in \mathcal{E}_{t+1}\} \quad (6)$$

where $\overline{g}_{t+1}^{e}$ is the intercept of cut e; $\overline{\mu}_{t+1}^{e}$ is the gradient of cut e; $\mathcal{E}_{t+1}$ is the set of the cuts in stage t+1.

The task of the forward pass is to generate trajectories for updating $\mathfrak{V}_{t+1}^{\pi}$ in the backward pass. The forward pass uses a Monte Carlo sampling technique to generate $\mathcal{K}$ scenarios $(\varsigma_1,\ldots,\varsigma_s,\ldots,\varsigma_{\mathcal{K}})$ from the scenario tree. Then (5) will be solved on the $\mathcal{K}$ scenarios from $t = 1, 2,\ldots, T$ to generate $\mathcal{K}$ trajectories. For stage T, we have $\mathfrak{V}_{T+1}^{\pi}(a_T) \equiv 0$.

2) Backward Pass

The task of the backward pass is to generate supporting hyperplanes to update $\mathfrak{V}_t^{\pi}$. The backward pass will solve (5) from $t = T$ to $t = 2$ with the trajectories extracted randomly from the experience replay buffer. After solving (5) for all scenarios at stage t, the dual variables μ_t of the constraint $\mathcal{A}(s_t) = \mathcal{A}(\widehat{x}_{t-1}, \xi_t)$ and the optimal values $\overline{V}_t^{\pi}$ $(s_t(\xi_t))$ are obtained to generate the cuts, which are added to all sub-problems at stage t-1 to update $\mathfrak{V}_t^{\pi}$. The gradient $\overline{\mu}_t^{e}$ and intercept $\overline{g}_t^{e}$ of cut e are updated as follows:

$$\overline{\mu}_t^{e} = T_t^{\top} \sum\nolimits_{\xi_t \in \Xi_t} p(\xi_t \mid \xi_{t-1}) \mu_t(\xi_t) \quad (7)$$

$$\overline{g}_t^{e} = \sum\nolimits_{\xi_t \in \Xi_t} p(\xi_t \mid \xi_{t-1}) \overline{V}_t^{\pi}(s_t(\xi_t)) + \overline{\mu}_t^{e\top} T_t \widehat{x}_{t-1} \quad (8)$$

The cuts will update the $\mathfrak{V}_t^{\pi}$ to improve the approximation of the expected cost-to-go function iteratively. At the early iterations of training, the expert performance data are better than the trajectories generated in the forward pass. The expert performance data are more likely to be chosen by giving a larger probability. After several iterations, the expert performance data will be given the same probability as the trajectories generated in the forward pass. This means that in later iterations, the forward passing trajectory will be treated as important as expert performances.

B. Convergence Criterion of SDDPIL

Different from model-free methods, SDDPIL has an explicit convergence criterion which is guaranteed by the probability theory. The SDDPIL provides the lower bound (LB) by solving the first stage problem.

$$LB = V_1^{\pi}(s_1) = \min\nolimits_{a_1 \in \mathcal{A}(s_1)} Q_1^{\pi}(s_1, a_1) \quad (9)$$

The upper bound (UB) is provided by calculating the expectation of the return of several episodes. The UB is a confidence interval for the random sampling.

$$UB = [Z - z_{\alpha/2} \frac{\sigma}{\sqrt{\mathcal{K}}}, Z + z_{\alpha/2} \frac{\sigma}{\sqrt{\mathcal{K}}}] \quad (10)$$

$$Z = \frac{1}{\mathcal{K}} \sum\nolimits_{m=1}^{\mathcal{K}} \sum\nolimits_{t=1}^{T} r_{t,m}(s_t, a_t) \quad (11)$$

$$\sigma = \sqrt{\frac{1}{\mathcal{K}-1} \sum\nolimits_{m=1}^{\mathcal{K}} (\sum\nolimits_{t=1}^{T} r_{t,m}(s_t, a_t) - Z)^2} \quad (12)$$

where $z_{\alpha/2}$ is the $(1 - \alpha)$ quantile of the standard normal distribution. Finally, the flowchart of the SDDPIL algorithm is shown in Table I.