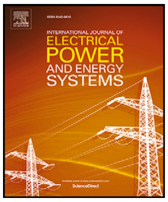
Electrical Power and Energy Systems 142 (2022) 108359

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes



Approximate dynamic programming with policy-based exploration for microgrid dispatch under uncertainties☆

Avijit Das [a], Di Wu [a,*], Zhen Ni [b]

[a] Pacific Northwest National Laboratory, Richland, WA 99352, USA
[b] Florida Atlantic University, Boca Raton, FL 33431, USA

ARTICLE INFO

Keywords:
Approximate dynamic programming
Battery energy storage systems
Control policy design
Microgrid
Stochastic optimization

ABSTRACT

Approximate dynamic programming (ADP) is a promising approach for power system scheduling and dispatch under uncertainties. This paper presents an innovative ADP-based dispatch method for a microgrid with intermittent renewable generation, battery energy storage systems, and controllable distributed generators. The proposed ADP algorithm is based on a double-pass value iteration approach and takes advantage of the underlying properties of the microgrid dispatch problem. In the forward pass, decision variables are updated moving forward in time using an ϵ-greedy strategy to balance exploitation and exploration. In particular, an approximate optimization method is proposed to speed up exploitation. In addition to random exploration, a policy is designed to guide the algorithm to explore some promising solution space in a probabilistic manner. In the backward pass, the value function is updated moving backward in time using the trajectory of states, decisions, and outcomes of the sample path in the forward pass. The proposed method is evaluated through numerical experiments in both deterministic and stochastic environments. Case study results show that the proposed method demonstrates improved performance in both optimization gap and computation time in comparison to conventional methods.

1. Introduction

The recent development of distributed energy resources, including renewable generation (RG), battery energy storage systems (BESS), and dispatchable distributed generators (DG), makes them valuable assets in microgrids [1]. These emerging resources not only provide economic benefits, but also help to improve system resilience [2]. On the other hand, the variability and uncertainty associated with RG present challenges to system operation, requiring advanced methods for microgrid dispatch [3]. Scenario-based stochastic programming (SBSP) is a popular approach for microgrid scheduling and dispatch under uncertainties [4]. Scenarios can be generated using the roulette wheel mechanism or Monte Carlo simulation [5]. The performance of SBSP methods highly depends on the training set of scenarios and the associated probabilities [6]. Often, a large number of scenarios are required to capture the stochasticity well, making the problem computationally intensive and sometimes even infeasible to solve [7]. Scenario reduction techniques can help simplify the problem [8], but may neglect low-probability yet high-impact scenarios. In addition, the existing SBSP approaches are limited to fixed scheduling plans and may not respond to unexpected variations in real-time.

The stochastic microgrid dispatch can also be formulated as dynamic programming (DP) problems with uncertainties [9]. For example, an optimization framework based on stochastic DP is developed in [10] to simultaneously address uncertainties in loads and prices as well as risk consideration and energy hub operational constraints. Because of the widely known “curse of dimensionality”, classical DP has limited application to practical stochastic engineering problems [11]. Approximate dynamic programming (ADP) is a broad umbrella for a modeling and algorithmic strategy for solving large and complex stochastic sequential decision-making problems approximately [12]. Central to ADP is making decisions based on value function approximation (VFA)—a collection of function representations, techniques, and methods aimed at providing a scalable and effective approximation to exact value functions [13]. In ADP algorithms, the temporal difference with forward-in-time and/or backward-in-time fashion is used to update values and decision variables through an iterative process [14].

☆ This work was supported in part by the U.S. Department of Energy (DOE), Office of Electricity through the Energy Storage program and in part by the National Science Foundation under Grants 1949921 and 2047064. Pacific Northwest National Laboratory is operated for the U.S. DOE by Battelle Memorial Institute under Contract DE-AC05-76RL01830.

* Corresponding author.
E-mail address: di.wu@pnnl.gov (D. Wu).

https://doi.org/10.1016/j.ijepes.2022.108359
Received 19 February 2022; Received in revised form 18 April 2022; Accepted 19 May 2022
Available online 31 May 2022

Based on the path taken to search for an optimal policy, there are two categories of ADP algorithms: value iteration and policy iteration [15]. Policy iteration algorithms explicitly estimate the value of the current policy to some level of accuracy within an inner loop and the estimated value function is used to compute a new, improved policy within an outer loop. On the other hand, value iteration algorithms iteratively search for the approximated value function [16].

Various ADP algorithms have been proposed during the past few years for optimal scheduling and dispatch of energy storage and microgrid. Some of them are based on the policy iteration strategy. For example, an ADP approach is developed in [17] to generate an optimal bidding strategy for RG paired with storage participating in day-ahead markets, where least squares policy evaluation is used to estimate the weights of a polynomial VFA. In [18], an ADP algorithm is proposed to solve high dimensional optimization problems for an integrative home energy system consisting of energy storage and electric vehicles. The authors in [19] propose a dynamic energy management mechanism for economical operation of microgrids in real-time, where a deep recurrent neural network is employed to learn the value function.

Value iteration is the most widely used method and often tends to be the most natural way of updating an estimate of the value of being in a state. As a result, a large amount of literature on ADP-based storage and microgrid dispatch is based on value iteration. Many of them are pure forward-pass algorithms. Just to name a few, the authors in [20] propose an ADP approach with a look-up table for VFA to dispatch BESS in an islanded microgrid, considering potential impacts on battery life. In the ADP-based microgrid dispatch presented in [21], a new method is proposed to calculate the sample observation of the marginal value of the energy stored in the battery, and thereby to update the slope of the piecewise linear function. A data-driven hybrid method using model predictive control and ADP is proposed in [22] to study real-time stochastic operation of microgrids, where the empirical knowledge obtained from the historical data used for offline training and piecewise linear functions are used to approximate value functions. To address long-term load growth uncertainty and short-term power fluctuation, the authors in [23] propose an ADP-based flexible distribution system expansion planning model, where the value table is initialized using historical observations. In [24], an ADP algorithm is proposed for real-time dispatch of an integrated heat and power system, where the high dimensional state variables are aggregated into the state of charge (SOC) of the battery and available heat in heat storage. Double-pass ADP algorithms have also been proposed for microgrid and storage dispatch. For example, a computationally efficient smart home energy management system is proposed in [25], considering stochastic energy consumption and photovoltaics (PV) generation models over a horizon of several days. In [26], a double-pass ADP based on a structured lookup table is proposed for optimal scheduling of wind paired with storage and benchmarked against the optimal solution on a library of deterministic and stochastic problems.

When the states and actions become large and problems become complicated without preferrable convex or concave structure, existing ADP methods may become less efficient [27]. The empirical knowledge obtained from the historical data can help to improve the learning performance [28]. However, this approach may require extra efforts to collect the historical data, and the learning performance of the ADP algorithm may highly be influenced by the expert demonstration. One promising solution is to develop customized ADP taking advantage of the underlying properties of the problems to improve performance. In fact, knowledge and information about problem structure and characteristics have been used in designing analytical approximation functions to map states to actions without solving an embedded optimization problem. For example, a policy function approximation (PFA) method is proposed in [29] to solve the stochastic dispatch problem of a grid-connected microgrid, where developing analytical approximation functions using the problem characteristics was the key concept. In [30], a PFA algorithm is proposed based on the problem knowledge obtained from the data-driven supervised learning approach to develop an effective control mechanism for the PV storage systems. Although these approximation functions or heuristic rules alone may not perform well for complex problems [31], incorporating these approximate control policies into ADP could help enhance learning efficiency.

This paper proposes an innovative ADP-based optimal dispatch of a remote microgrid consisting of RG, BESS, and DG. In this problem, multiple BESS and DG assets need to be optimally dispatched at each time step to minimize the operation cost over time without knowing exactly the future load and RG. The multi-time-period sequential decision-making problem is modeled as a Markov decision process (MDP). The proposed ADP follows the double-pass value iteration strategy with decision variables updated using an ϵ-greedy strategy in the forward pass to balance exploration and exploitation. The main contributions of this paper are twofold:

- The optimal dispatch problem is nonlinear in general, making the optimization in exploitation challenging to solve. To speed up exploitation, instead of solving the optimization problem exactly, a rule-based dispatch method is proposed to more efficiently and approximately explore the solution space.
- The underlying properties of the microgrid dispatch problem are explicitly utilized in the algorithm design. To enhance exploration capability, in addition to exploring decisions randomly, a customized policy is designed to guide the algorithm to explore some promising decisions and thereby to learn the value function faster and more accurately.

The rest of this paper is organized as follows. Section 2 presents ADP preliminaries. The model description and problem formulation are provided in Section 3. Section 4 presents the proposed ADP approach. The proposed method is illustrated and evaluated through case studies in Section 5. Finally, the conclusion and future work are offered in Section 6.

2. Preliminaries

A sequential decision-making problem in deterministic and stochastic environments can be formalized as an MDP [32]. In an MDP, at each discrete time step t, a decision-maker or an agent measures the state S_t of the environment, and takes a decision x_t from the feasible set χ_t. The decision results in cost $C(S_t, x_t)$ and the system arrives at a new state S_{t+1} following the transition function $S^M(\cdot)$:

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}), \tag{1}$$

where W_{t+1} is the exogenous information that becomes available between t and $t+1$. The goal is to find the optimal policy π that minimizes the cumulative sum of cost over the course of interaction:

$$\min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t=0}^{T} \gamma^t C(S_t, X^\pi(S_t)) \right\}, \tag{2}$$

where $\mathbb{E}\{\cdot\}$ is the expectation operator, γ is a discount factor, T denotes the planning horizon, and $X^\pi(S_t)$ is a policy function that returns a decision x_t. We need an expectation in (2) because the information variable W_{t+1} is random at times before $t+1$. Both classical and approximate DP specifically focus on using Bellman's equation, which can be written in the expectation form:

$$V_t(S_t) = \min_{x_t \in \chi_t} [C(S_t, x_t) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1} | S_t, x_t)\}], \tag{3}$$

where $V_t(S_t)$ is the value function or cost-to-go function of state S_t, which captures the expected value of being in a state at t and following an optimal policy forward. The embedded expectation makes (3) computationally expensive to solve when the state, action, and information spaces become large and multi-dimensional. To circumvent the embedded expectation and overcome the curse of dimensionality, the post-decision state S_t^x is introduced to represent the state instantly after the current decision x_t is made, but before the arrival of any new information [15]. According to the definition, the post-decision state S_t^x is a deterministic function of S_t and x_t: $S_t^x = S^{M,x}(S_t, x_t)$, where $S^{M,x}(\cdot)$ is the transition function, which can be obtained by removing W_{t+1} from $S^M(\cdot)$. Thus, the post-decision state updates the physical state based on the action taken and keeps the information state the same as S_t [27].

The value function of the post-decision state captures the expected downstream costs:

$$V_t^x(S_t^x) = \mathbb{E}\{V_{t+1}(S_{t+1})|S_t, x_t\}. \tag{4}$$

Hence, Bellman's equation can be written in a post-decision formulation:

$$V_t(S_t) = \min_{x_t \in \mathcal{X}_t} [C(S_t, x_t) + \gamma V_t^x(S_t^x)]. \tag{5}$$

Solving the optimization with embedded expectation in (3) requires making a decision considering all outcomes and is often computationally intractable. On the other hand, the expectation operator is replaced by the value function of the post-decision state in (5), and the deterministic optimization can be used to make decisions. The optimality equation around the post-decision state can be derived as

$$V_{t-1}^x(S_{t-1}^x) = \mathbb{E}\left\{ \min_{x_t \in \mathcal{X}_t} (C(S_t, x_t) + \gamma V_t^x(S_t^x)) \middle| S_{t-1}^x \right\}. \tag{6}$$

The expectation is outside of the minimization operator, which offers a computational advantage. Mechanisms can be designed to update the estimate of the post-decision value function based on observations from individual sample paths, which refer to particular sequences of exogenous information.

Based on these optimality equations, different value iteration algorithms can be developed to compute or approximate the value function, from which an optimal policy is then derived. All these algorithms involve solving the optimality equations using estimates of the downstream values iteratively to learn the post-decision value function. Let n denote the iteration index, S_t^n and x_t^n denote the state and decision, respectively, at iteration n, and $\bar{V}_t^{n-1}(\cdot)$ denote the estimated value associated with a post-decision state at iteration $n-1$. Following (5), the decision expected to minimize the value of being at state S_t^n can be expressed as

$$x_t^n = \arg\min_{x_t^n} (C(S_t^n, x_t^n) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t^n))). \tag{7}$$

In a double-pass approach, the sample realization of the value of being in state S_t^n can be updated using (8) in the backward pass based on the sample path in the forward pass:

$$\hat{v}_t^n = C(S_t^n, x_t^n) + \hat{v}_{t+1}^n. \tag{8}$$

Note that with a realized W_t, $\hat{v}_t^n$ is not only a sample of the value of being in state S_t^n, but also a sample of the value of being in post-decision state $S_{t-1}^{x,n}$. Therefore, we can use the observations to update the estimate of post-decision value function iteratively with respect to n using (9) based on the stochastic gradient algorithm [15] and thereby gradually approximate the expectation in (6):

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1 - \alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha \hat{v}_t^n , \tag{9}$$

where $\alpha_{n-1} \in [0, 1]$ is the step-size that is used to smooth the function approximation. The smoothing is required to capture the uncertainties in $\hat{v}_t^n$ and approximately calculate the expectation in (6). The goal is to use the sample observations ($\hat{v}_t^n$) to approximate the mean of the distribution from which the observations are being drawn. Once the approximate post-decision value function $\bar{V}_t$ is obtained for all time steps, given a state at a time step, a decision can be made by solving (10):

$$x_t = \arg\min_{x_t} (C(S_t, x_t) + \bar{V}_t(S_t^x)). \tag{10}$$

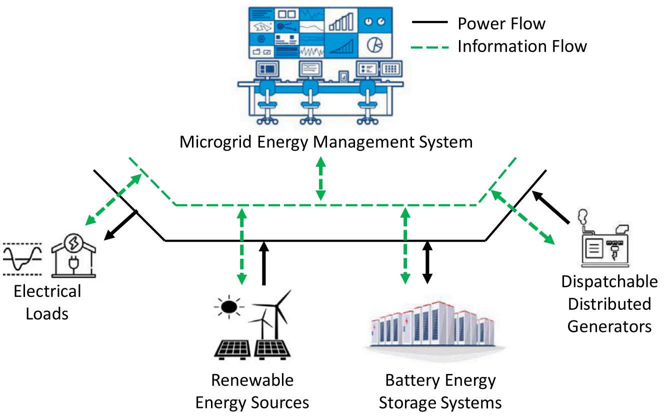

Power Flow
Information Flow
Microgrid Energy Management System
Electrical Loads
Renewable Energy Sources
Battery Energy Storage Systems
Dispatchable Distributed Generators

Fig. 1. Schematic diagram of the microgrid with power and information flows.

Note that obtaining the approximated value function is the most time-consuming part, and the time required to solve (10) is relatively ignorable.

3. Model description and problem formulation

In this paper, we consider a remote microgrid that consists of various distributed energy resources, including distributed RG such as PV and wind, BESS assets, and dispatchable DGs such as diesel engines. A schematic diagram of the microgrid with power and information flows is provided in Fig. 1. As can be seen, there are bi-directional information flows between the microgrid resources and the operator, so the microgrid operator collects resource information and sends back the dispatch decisions. In the given microgrid model, the power flows are unidirectional except for BESSs because of their charging and discharging operations. The microgrid resources need to be optimally scheduled and dispatched over a look-ahead time horizon to minimize the cumulative operation cost subject to system- and component-level constraints in the presence of uncertainties in RG and load. This section first describes models used for the microgrid dispatch problem and then presents an MDP formulation of the problem.

3.1. Microgrid dispatch problem

Let M and J denote the numbers of BESSs and DGs, respectively, m and j denote the index of BESS and DG, respectively, and t denote the time step index. The objective is to minimize the expected microgrid operation cost over a finite horizon of time T considering uncertainties from RG and load:

$$\min \mathbb{E}\left[\sum_{t=1}^{T} C_t\right], \tag{11}$$

subject to the power balance at the system level:

$$\sum_{m=1}^{M} p_{t,m}^{\text{batt}} + \sum_{j=1}^{J} p_{t,j}^{\text{dg}} + r_t - p_t^{\text{dump}} = l_t, \quad \forall t, \tag{12}$$

where $p_{t,m}^{\text{batt}}$ is the BESS charging/discharging power (positive when discharging), $p_{t,j}^{\text{dg}}$ is the power output of DG j, r_t is the RG output, p_t^{dump} is the dumped RG, and l_t is the system load. Both $p_{t,j}^{\text{dg}}$ and p_t^{dump} are nonnegative. For simplicity, it is assumed that there is always sufficient power to meet the load and therefore there is no unserved load. In cases where unserved load needs to be considered, the dumped renewable can be replaced by the imbalance power to represent either dumped renewable or unserved load. The objective function consists of three components: BESS degradation cost, DG generation cost, and penalty on dumped renewable, as expressed in (13):

$$C_t = \sum_{m=1}^{M} C^{\text{batt}}_{t,m} + \sum_{j=1}^{J} C^{\text{dg}}_{t,j} + \pi p^{\text{dump}}_t , \tag{13}$$

where π is a penalty price for the dumped renewable. The penalty term is introduced to avoid the dumped renewable whenever possible. The individual component models, BESS degradation cost $C^{\text{batt}}_{t,m}$, and DG generation cost $C^{\text{dg}}_{t,j}$ are detailed as follows.

3.1.1. Battery energy storage system

BESS can be used to shift load and store excessive generation from renewable energy, and thereby help reduce microgrid operation cost. A BESS can be modeled as a scalar linear system that resembles simplified dynamics of energy state parameterized by charging and discharging power limits, energy state limits, and efficiencies [33]. The dynamics of SOC can be expressed as

$$s_{t+1,m} = s_{t,m} - \frac{\Delta e_{t,m}}{E_m}, \quad \forall m, \forall t, \tag{14}$$

where $s_{t,m}$ is the BESS SOC at time t, E_m is the usable energy capacity, and $\Delta e_{t,m}$ is the change of energy stored in the battery m at time step t, as expressed in (15):

$$\Delta e_{t,m} = \begin{cases} \eta^-_m p^{\text{batt}}_{t,m} \Delta t, & \text{if } p^{\text{batt}}_{t,m} \le 0 \\ \dfrac{p^{\text{batt}}_{t,m}}{\eta^+_m} \Delta t, & \text{otherwise} \end{cases} \quad \forall m, \forall t, \tag{15}$$

where η^-_m and η^+_m are the charging and discharging efficiencies of BESS m, respectively, and Δt is the time step size. The BESS power is constrained by

$$-P^-_m \le p^{\text{batt}}_{t,m} \le P^+_m, \quad \forall m, \forall t, \tag{16}$$

where P^-_m and P^+_m are the maximum charging and discharging power, respectively. The SOC is constrained by

$$\underline{s}_m \le s_{t+1,m} \le \bar{s}_m, \quad \forall m, \forall t, \tag{17}$$

where $\underline{s}_m$ and $\bar{s}_m$ are the minimum and maximum SOC limits, respectively.

An aging or degradation cost is introduced to capture the loss of life associated with different charging and discharging operations. The battery degradation cost per kWh of discharged energy can be calculated based on the cost of a BESS and the lifetime energy throughput [34]. Therefore, the BESS degradation cost can be expressed as

$$C^{\text{batt}}_{t,m} = \begin{cases} g_m p^{\text{batt}}_{t,m} \Delta t, & \text{if } p^{\text{batt}}_{t,m} > 0 \\ 0, & \text{otherwise} \end{cases} \quad \forall m, \forall t, \tag{18}$$

where g_m is the battery degradation cost in \$/kWh for BESS m.

3.1.2. Dispatchable DG

At any time t, the output of a DG depends on its ON/OFF status and is limited by the minimum loading level and rated power [28]:

$$\begin{cases} k^{\text{dg}}_j p^{\text{rated}}_j \le p^{\text{dg}}_{t,j} \le p^{\text{rated}}_j, & \text{if } d_{t,j} > 0 \\ p^{\text{dg}}_{t,j} = 0, & \text{otherwise} \end{cases} \quad \forall j, \forall t, \tag{19}$$

where k^{dg}_j is the minimum load level as a percentage of rated power p^{rated}_j of DG j, and $d_{t,j}$ is the state of DG j that counts the time of continuous ON (if >0) or OFF (if <0) operation.

The DG generation cost can be modeled as a quadratic function of power output when a DG is ON:

$$C^{\text{dg}}_{t,j} = \begin{cases} a_j (p^{\text{dg}}_{t,j})^2 + b_j p^{\text{dg}}_{t,j} + c_j, & \text{if } d_{t,j} > 0 \\ 0, & \text{if otherwise} \end{cases} \quad \forall j, \forall t, \tag{20}$$

where a_j, b_j, and c_j are the coefficients of the quadratic function for DG j. Let $y_{t,j}$ denote an integer variable that indicates the switching operation of DG j at the end of time step t: 1 for startup (switching from OFF to ON), −1 for shutdown (switching from ON to OFF), and 0 for maintaining the same ON/OFF status. The feasible switching operation of the DG is determined considering the startup/shutoff time constraints:

$$\begin{cases} y_{t,j} \in \{-1, 0\}, & \text{if } d_{t,j} \ge T_{g,\text{ON}} \\ y_{t,j} \in \{1, 0\}, & \text{if } d_{t,j} \le -T_{g,\text{OFF}} \\ y_{t,j} = 0, & \text{otherwise} \end{cases} \quad \forall j, \forall t, \tag{21}$$

where $T_{j,\text{ON}}$ and $T_{j,\text{OFF}}$ are the minimum on and off time of DG j, respectively.

The DG transition function is given by (22):

$$d_{t+1,j} = \begin{cases} 1, & \text{if } y_{t,j} = 1 \\ -1, & \text{if } y_{t,j} = -1 \\ d_{t,j} + 1, & \text{if } y_{t,j} = 0 \ \& \ d_{t,j} \ge 1 \\ d_{t,j} - 1, & \text{if } y_{t,j} = 0 \ \& \ d_{t,j} \le -1 \end{cases} \quad \forall j, \forall t. \tag{22}$$

The DG startup and shutdown costs can be readily expressed as a function of the switching operation, but are excluded from this paper for simplicity without affecting the nature of the microgrid dispatch problem from the perspective of ADP algorithm design.

3.1.3. Renewable generation and load

RG is assumed to be operated in the maximum power point tracking (MPPT) mode to maximize the environmental and economic benefits unless the excess power cannot be taken by the load and BESSs [35]. The RG output depends on weather conditions and cannot be accurately forecasted. The actual RG power output in MPPT mode r_t is the sum of the forecast $\hat{r}_t$ and random forecast error ε_r, and should always be non-negative and capped at the rated power r^{rated} [36]:

$$r_t = \min\{\max\{\hat{r}_t + \varepsilon_{r,t}, 0\}, r^{\text{rated}}\}. \tag{23}$$

When the excess power cannot be stored in BESSs, RG power needs to be dumped, as can be captured by p^{dump}_t.

Similar to the RG, a random forecast error is introduced to capture the uncertainty in load forecast:

$$l_t = \min\{\max\{\hat{l}_t + \varepsilon_{l,t}, l_{\min}\}, l_{\max}\}, \tag{24}$$

where l_t and $\hat{l}_t$ are the actual and forecast load, respectively, ε_l is the forecast error, $l_{\min}$ and $l_{\max}$ are the minimum and maximum load, respectively.

3.2. MDP formulation

The microgrid dispatch problem can be formulated as an MDP following the canonical model in [37]. First, we need to build up a set of state variables to capture all information that is necessary and sufficient to make decisions, calculate costs, and simulate process over time. In this problem, the set of state variables includes both physical and information states:

$$S_t = (\underbrace{s_{t,m},\ d_{t,j},}_{\text{physical}}\ \underbrace{r_t,\ l_t}_{\text{information}}). \tag{25}$$

The set of decision variables is

$$x_t = (p^{\text{batt}}_{t,m}, p^{\text{dg}}_{t,j}, y_{t,j}, p^{\text{dump}}_t), \quad x_t \in \chi_t. \tag{26}$$

A decision must be made from the feasible space χ_t defined by (12), (14)–(17), (19), (21), (23), and (24). The corresponding cost $C(S_t, x_t)$ is determined according to (13), (18), and (20). The set of exogenous information is the forecast errors:

$$W_t = (\varepsilon_{r,t}, \varepsilon_{l,t}), \tag{27}$$

Algorithm 1 The proposed ADP algorithm.

```
Initialization
while $n < N$ do
    Choose a sample path $\omega^n$
    for $t = 1 : T$ do                                      ▷ Forward Pass
        if $rand > \epsilon_1$ then                         ▷ Exploitation
            Solve (7) using the proposed approx. opt.
        else                                                ▷ Exploration
            if $rand \le \epsilon_2$ then
                Execute the proposed policy:
                $x_t^n = X^{\text{expl}}(S_t|\theta)$
            else
                Take a decision $x_t^n \in \chi_t$ randomly
            end if
        end if
        Post-decision state: $S_t^{x,n} = S^{M,x}(S_t^n, x_t^n)$
        if $t < T$ then
            Next state: $S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$
        end if
    end for
    for  $t = T : 1$ do                                     ▷ Backward Pass
        Compute $\hat{v}_t^n$ using (8) with $\hat{v}_{T+1}^n = 0$.
        Update VFA $\bar{V}_{t-1}^n(S_{t-1}^{x,n})$ using (9)
    end for
    $n \leftarrow n + 1$
end while
Return the post-decision value function $(\bar{V}_t^N)_{t=1}^T$
```

which becomes available at time t. Given the current state S_t, the decision x_t, and the exogenous information W_{t+1}, the system arrives at the next state S_{t+1} following the transition function $S_{t+1} = S^M(S_t, x_t, W_{t+1})$, which is defined by (14) and (15) for BESS SOC, (22) for DG ON/OFF status, (23) for RG output, and (24) for system load. Note that RG output and load are independent of state and decision variables and only depend on the exogenous information and the forecast that is given and remain unchanged throughout the planning horizon.

4. Proposed approximate dynamic programming approach

This paper presents an innovative ADP approach for the microgrid dispatch problem. The proposed approach follows a double-pass value iteration process using a lookup table (also referred to as tabular function) with discretized states for VFA. Note that when microgrid size becomes extremely large with hundreds to thousands of distributed energy resources, the lookup table method may not be sufficient and can be replaced by parametric and non-parametric models (e.g., neural network-based models).

- The forward pass updates the microgrid dispatch decisions moving forward in time using the ϵ-greedy strategy to balance exploitation and exploration. The underlying properties of the microgrid dispatch problem are explicitly utilized for designing (i) an approximate optimization to speed up exploitation and (ii) a customized policy to enhance exploration capability, and thereby to learn the dispatch policy over iteration in a more efficient manner.
- The backward pass updates the value function moving backward in time using the trajectory of states, decisions, and outcomes of the sample path in the forward pass.

The proposed method is detailed in Algorithm 1. To start the algorithm, a number of initial values and parameters need to be set, including the value table, maximum iteration number, initial states, exploration parameters, and parameters used in the policy-based exploration. Note that the post-decision values table is a matrix of estimated value with respect to discretized state (row) and time step (column). For RG and load, as forecasts are given and not updated over time, there is only one value for these two post-decision states, which are their forecasts. Therefore, the combinations of discretized states degenerate to the combinations of BESS SOC and DG ON/OFF status. Often, the policy-based decision-making technique comes with parameters that need to be tuned properly to achieve acceptable performance. The tuning parameters for the proposed policy-based exploration are $\theta = (\theta_h, \theta_l)$, where θ_h and θ_l represent net load thresholds that activate BESS discharging and charging, respectively.

At each iteration n, a sample path ω^n is first generated to represent a realization of exogenous information. Forward pass and backward pass are then executed in sequence:

- In the forward pass, at each time step t, a random number is first generated and compared with ϵ_1 to determine whether decisions are updated by exploitation or exploration. If exploitation is selected, a decision is made by solving (7) using the proposed approximate optimization. If exploration is selected, another random variable is generated and compared with ϵ_2 to determine whether to take a decision randomly or using the proposed policy based on current states. The post-decision state and the state at the next time step are then obtained using transition functions, where $W_{t+1}(\omega^n)$ denotes the realization of the exogenous information at time step $t+1$ contained in sample path ω^n.
- In the backward pass, the sampled realizations $\hat{v}^n$ are calculated using (8) and the post-decision value functions $\bar{V}^n$ are updated using (9).

This procedure repeats until the maximum iteration number N is reached. The final output is a trained post-decision value table for making dispatch decisions in real time by solving (10). The proposed approximate optimization for exploitation and policy for exploration are detailed as follows.

4.1. Approximate optimization for exploitation

Eq. (7) can be solved exactly by discretizing all decisions and exploring all possible combinations at a high computational cost. A method is proposed herein to solve the optimization approximately and thereby speed up the exploitation. The main idea is to first find all feasible BESS power solutions based on the current SOC and feasible SOC at the next time step. A rule-based dispatch is then designed to more efficiently and approximately explore DG operations for each feasible BESS power solution. The DG dispatch strategy is proposed to avoid high computation costs to solve the given microgrid dispatch problem. The proposed approach is applicable for microgrids with DGs, and it guarantees the dispatch solution. Detailed steps are provided as follows.

1. First, given the current BESS SOC $s_{t,m}$, we find all feasible BESS SOC at the next time step. A feasible SOC must satisfy (17) to meet the SOC limits. In addition, considering the power limits, a feasible SOC also needs to satisfy (28), which can be derived from (14)–(16):

$$s_{t,m} - \frac{P_m^+ \Delta t}{\eta_m^+ E_m} \le s_{t+1,m} \le s_{t,m} + \frac{\eta_m^- P_m^- \Delta t}{E_m}. \tag{28}$$

For each feasible BESS SOC $s_{t+1,m}$, the corresponding BESS power can be calculated using (29):

$$p_{t,m}^{\text{batt}} = \begin{cases} \frac{(s_{t,m} - s_{t+1,m}) E_m}{\Delta t \eta_m^-}, & \text{if} \;\; s_{t,m} \le s_{t+1,m} \\ \frac{(s_{t,m} - s_{t+1,m}) E_m \eta_m^+}{\Delta t}, & \text{if} \;\; s_{t,m} \ge s_{t+1,m}. \end{cases} \tag{29}$$

Algorithm 2 Optimal DG dispatch.

```
Initialization:
Define a set $\mathcal{A}$ to list all DGs that are ON. Let $\mathcal{A}_i$ denote the i-th permutation of $\mathcal{A}$ to represent a dispatch order, and $N_p$ denote the number of permutation of $\mathcal{A}$. Initialize $C(S_t, x_t) = \infty$.
for $i = 1 : N_p$ do
    Set the required power: $p^a = l_t - r_t - \sum_{m=1}^{M} p^{\text{batt}}_{t,m}$
    for $g = 1 : |\mathcal{A}|$ do
        Determine power output of DG $j = \mathcal{A}_i(g)$:
            $p^{\text{dg},i}_{t,j} = \min(\max(k^{\text{dg}}_j p^{\text{rated}}_j, p^a_t), p^{\text{rated}}_j)$
        Update the required power: $p^a \leftarrow p^a - p^{\text{dg},i}_{t,j}$
    end for
    Calculate $p^{\text{dump},i}_t$ based on (12)
    Calculate $C^i(S_t, x_t)$ using (13)
    if $C^i(S_t, x_t) < C(S_t, x_t)$ then
        $C(S_t, x_t) = C^i(S_t, x_t)$, $p^{\text{dg}}_{t,j} = p^{\text{dg},i}_{t,j}$, and $p^{\text{dump}}_t = p^{\text{dump},i}_t$
    end if
end for
```

2. We then determine optimal DG outputs for each feasible BESS power solution. For DGs that are OFF, we simply set their power output and generation cost to zero. The remaining DGs need to be optimally coordinated to serve the net load minus BESS power. This can be achieved by searching different dispatch schemes for the lowest cost solution following a rule-based dispatch, as detailed in Algorithm 2.
3. In the value table, there are multiple states with the same BESS SOC combinations but different DG ON/OFF combinations at the next time step. The one with the least cost-to-go is chosen. The corresponding switching operations can be determined based on the current DG status and transition function given by (22).
4. Following steps 1–3, we obtain a pool of screened feasible decisions and the corresponding post-decision states. Using the current estimates from the value table, a decision can then be made by solving (7) to complete the exploitation.

4.2. Proposed policy for exploration

The design of the policy function ($X^{\text{expl}}(S_t|\theta)$) is the key to enhancing the exploration capability of the algorithm. A properly designed policy function can guide the ADP algorithm to explore some promising space in addition to making a decision randomly and thereby to learn the value function faster and more accurately. For the microgrid dispatch problem, BESS operation is critical to minimize the operation cost and therefore is determined first. Once the BESS operation is determined, the remaining decisions in x_t can be made following steps 2–3 in Section 4.1. When determining BESS operation, the proposed policy determines charging/discharging status first and then BESS power level, which are described as follows.

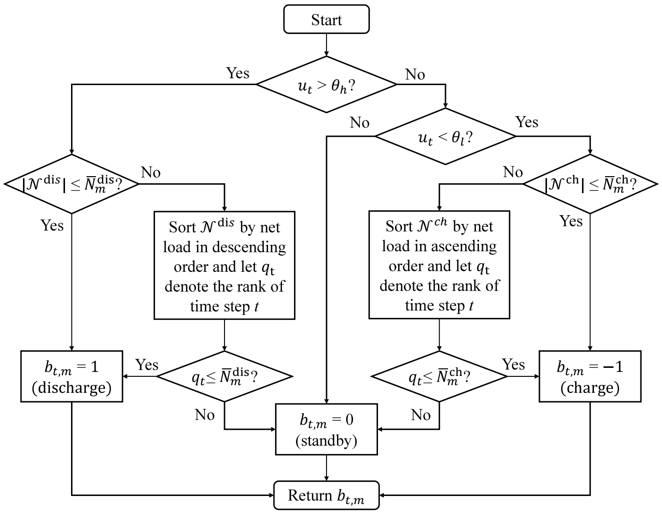

Start
$u_t > \theta_h$?
Yes
No
$u_t < \theta_l$?
No
Yes
$|\mathcal{N}^{\text{dis}}| \leq \overline{N}^{\text{dis}}_m$?
No
Yes
$|\mathcal{N}^{\text{ch}}| \leq \overline{N}^{\text{ch}}_m$?
No
Yes
Sort $\mathcal{N}^{\text{dis}}$ by net load in descending order and let q_t denote the rank of time step t
Sort $\mathcal{N}^{ch}$ by net load in ascending order and let q_t denote the rank of time step t
$b_{t,m} = 1$ (discharge)
Yes
$q_t \leq \overline{N}^{\text{dis}}_m$?
No
$b_{t,m} = 0$ (standby)
$q_t \leq \overline{N}^{\text{ch}}_m$?
Yes
No
$b_{t,m} = -1$ (charge)
Return $b_{t,m}$

Fig. 2. Flowchart for determining BESS operation status.

- *BESS charging/discharging status*

  We introduce two thresholds θ_h and θ_l for the net load, which is $u_t = l_t - r_t$. Because u_t increases with load and decreases with RG, its comparison with the thresholds can be used to indicate whether the net load is relatively high or low, which can be used to assist BESS charging/discharging decision-making. Herein, t is called a high-load time step if $u_t > \theta_h$ and a low-load time step if $u_t < \theta_l$. Note that the charging/discharging status should not be determined only based on the net load in comparison to the thresholds at the current time step. The temporal interdependency of BESS operation and the SOC constraint should also be considered. Without reasonably considering the temporal interdependency, BESS may be poorly dispatched, increasing the cumulative microgrid operation cost. For example, if we discharge a BESS whenever the net load exceeds the higher threshold θ_h, we may later find that the BESS runs out of energy and cannot be discharged when the net load is even higher and discharging is more valuable. Therefore, potential future needs should also be taken into account when determining the current BESS operating status.

  The proposed policy follows a simple strategy to determine BESS operating status. If the current time step t is a high-load time step, we consider putting BESS in discharging or standby mode, depending on whether a BESS has enough energy to be discharged for all future high-load time steps before a low-load time step becomes available for charging. Given the current SOC of BESS m, assuming the BESS is discharged at the rated power, the maximum number of time steps for discharging $\overline{N}^{\text{dis}}_m$ can be calculated. Let t_c be the first low-load time step after t within the planning time horizon, and $\mathcal{N}^{\text{dis}}$ denote a set that contains all high-load time steps between t and t_c.

  – If $|\mathcal{N}^{\text{dis}}| \leq \overline{N}^{\text{dis}}_m$, there is enough energy in BESS m to be discharged for all high-load time steps between t and t_c. Therefore, we set $b_{t,m} = 1$, which indicates that BESS m is discharged at time step t.
  – If $|\mathcal{N}^{\text{dis}}| > \overline{N}^{\text{dis}}_m$, we sort $\mathcal{N}^{\text{dis}}$ by net load in descending order and let q_t denote the rank of time step t. BESS m is discharged if $q_t \leq \overline{N}^{\text{dis}}_m$ and on standby otherwise.

  If the current time step t is a low-load time step, charging or standby decisions can be made similarly and are omitted here to conserve space. This procedure is summarized in Fig. 2.
- *BESS power level*

  Once BESS operating status is determined, the power level is set to zero for all BESSs with $b_{t,m} = 0$ and we only need to determine the power level for those that are not in standby mode.

  If t is a high-load time step, BESSs to be discharged ($b_{t,m} = 1$) are first sorted by degradation cost g_m in ascending order to determine discharging power level to serve the net load. In this way, a BESS with a lower degradation cost has a higher priority to be discharged. The discharging power is the minimum of three values: (i) the rated power, (ii) the power corresponding to an SOC equal to the lower limit at the end of time step, and (iii) the remaining net load, as expressed in (30):

$$p^{\text{batt}}_{t,m} = \min\left(P^{+}_{m}, \frac{(s_{t,m} - \underline{s}_{m})E_m\eta^{+}_{m}}{\Delta t}, p^{a}_{t}\right). \tag{30}$$

Algorithm 3 Policy for determining BESS discharging power level.

```
Initialization:
Set the required power: $p^a = l_t - r_t$. Generate a queue $\mathcal{D}$ by sorting BESSs with $b_{t,m} = 1$ by degradation cost $g_m$ in ascending order.
for $i = 1 : |\mathcal{D}|$ do
    $m = \mathcal{D}(i)$
    Determine $p_{t,m}^{\mathrm{batt}}$ using (30)
    Update the required power: $p^a \leftarrow p^a - p_{t,m}^{\mathrm{batt}}$
end for
```

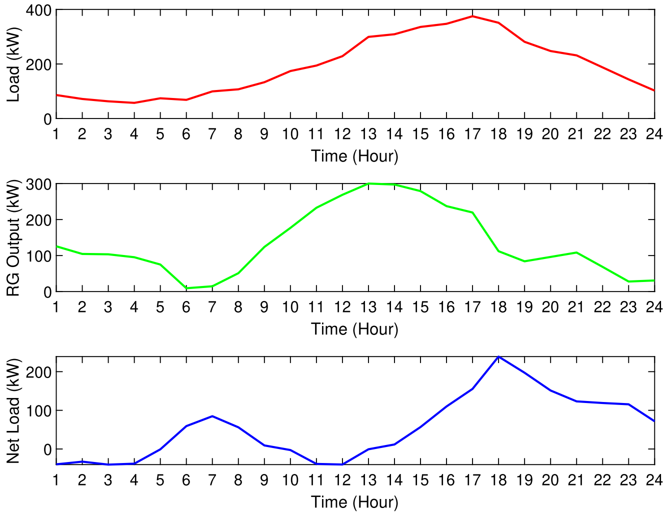


Fig. 3. Microgrid load, renewable generation, and net load.

The dispatch policy is detailed in Algorithm 3. If t is a low-load time step, the charging power can be determined similarly, with (30) in the algorithm replaced by (31):

$$p_{t,m}^{\mathrm{batt}} = \max\left(-P_m^-, \frac{(s_{t,m} - \underline{s}_m)E_m}{\Delta t \eta_m^-}\right). \tag{31}$$

5. Case study and simulation results

Case studies have been designed and carried out to validate and evaluate the proposed ADP approach in both deterministic and stochastic environments. To better demonstrate the performance in terms of optimization gap and computation time, we compared the proposed ADP with four existing algorithms: (i) classical DP [15], (ii) policy-iteration-based ADP [27], (iii) double-pass ADP [38], and (iv) forward-pass ADP [28].

The test system is a remote microgrid with two BESSs and three DGs in addition to PV and wind generation. A typical residential load profile in Phoenix, Arizona, from [39] was scaled to represent a residential microgrid load with a peak of 375 kW. The rated power of PV and wind are 150 kW and 200 kW, respectively, with their normalized output profiles generated using the System Advisory Model [40]. Fig. 3 plots the microgrid load, total power output from PV and wind, and the corresponding net load on a typical summer day, which is used for numerical experiments in this paper. Table 1 lists the DG parameters, which were adopted from [41] and slightly modified to match the system load. The parameters $T_{j,\mathrm{ON}}$ and $T_{j,\mathrm{OFF}}$ are assumed to be 1 h for all DGs. The BESS parameters were obtained and derived from [42], and are listed in Table 2. In particular, the 2-h BESS 1 and 6-h BESS 2 correspond to lithium-ion and vanadium redox flow battery technologies, respectively.

The parameters used for the ADP algorithms are listed in Table 3.

Table 1
DG parameters.

	Range (kW)	a_j (\$/kW^2)	b_j (\$/kW)	c_j (\$)
DG 1	[10,60]	0.00024	0.0267	0.38
DG 2	[20,60]	0.00052	0.0152	0.65
DG 3	[50,200]	0.00042	0.0185	0.40

Table 2
BESS parameters.

	Energy capacity (kWh)	Rated power (kW)	Round-trip efficiency (%)	Degradation cost ($)
BESS 1	100	50	83.7	0.069
BESS 2	240	40	68	0.070

Table 3
ADP parameters.

Param.	Value	Param.	Value
$\epsilon_1^{\mathrm{init}}$	0.7	$\epsilon_1^{\mathrm{final}}$	0.05
ϵ_2	0.5	α	0.5
θ_l	0 kW	θ_h	120 kW

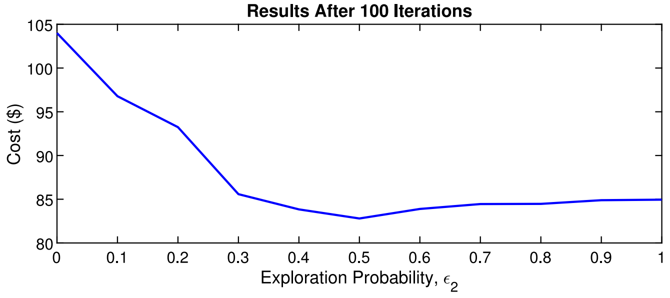


Fig. 4. Microgrid operation cost after 100 iterations for different ϵ_2 values.

- Inspired by the decayed ϵ-greedy strategy in [43], a decaying ϵ_1 was used to balance between exploitation and exploration. In particular, ϵ_1 is initialized at 0.7 ($\epsilon_1^{\mathrm{init}}$) and divided by 1.7 every 20 iterations until it reaches 0.05 ($\epsilon_1^{\mathrm{final}}$).
- The parameter N is a heuristic parameter that controls when to terminate the iteration. Theoretically, increasing the value of the exploration parameter ϵ_1, more iteration is required to achieve the same level of solution accuracy, but the average time per iteration decreases. In practice, N should be determined based on desired solution accuracy and computation time. Note that Algorithm 1 runs offline. Therefore, when the solution accuracy is not satisfied, the user can continue the iteration to obtain improved solutions. For the system studied in this paper, the proposed approach shows around 1% of the optimization gap at 100 iterations, and we set $N = 100$ for the deterministic case study.
- We use the grid search method to find a suitable ϵ_2 for the given microgrid dispatch problem and set ϵ_2 to be 0.5. The microgrid operation cost over different exploration rates is plotted in Fig. 4. As can be seen, the operation cost is maximum when there is no policy-based exploration. With the increment of ϵ_2, the microgrid operation cost drops significantly as the policy-based exploration shares knowledge of some promising solution spaces with the algorithm. The plot shows that the operation cost keeps increasing after $\epsilon_2 = 0.5$, which indicates that a balance between policy-based and random explorations is required to obtain a promising approximate solution.
- The thresholds used in the policy-based exploration are heuristic. We just need some net load values at which the difference in

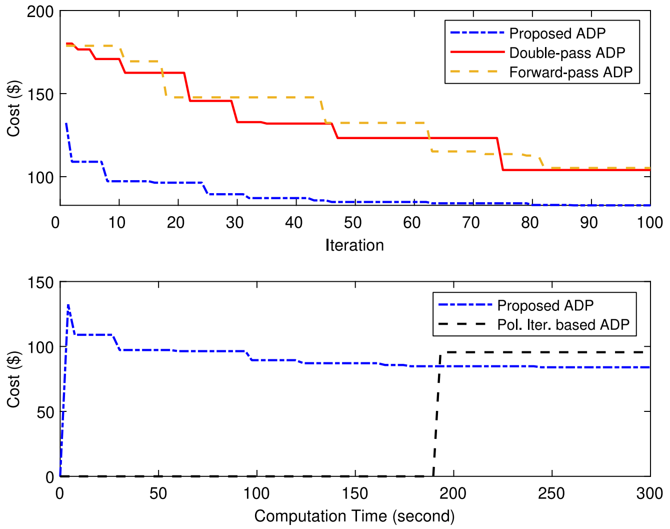


Fig. 5. Convergence performance of the proposed versus conventional ADP approaches.

DG marginal cost can overcome BESS losses and degradation cost. These values can be determined by analytical or iterative searching methods.

- Finally, a fixed step-size of $\alpha = 0.5$ was used to strike a balance between observations and update the value functions in the backward pass.

We use the same ϵ-greedy strategy for decision-making while training the existing ADP approaches. For the policy-iteration-based ADP approach, we use a neural network to approximate the value function with a sample size of 10. All numerical experiments were performed in MATLAB $R2020b$ on a computer with an Intel Core i7-8665U 1.90 GHz CPU and 16 GB RAM.

5.1. Deterministic environment

In this simplified case, it is assumed that we have a perfect forecast of system load and RG with uncertainties ignored. The deterministic problem can be solved exactly using the classical DP to serve as a benchmark. ADP algorithms are mainly used to handle uncertainties, but can also be applied to deterministic problems. In addition to computation time, another important performance metric for ADP is the optimization gap in percentage, which is defined as

$$Gap = \frac{|V - V^*|}{V^*}, \qquad (32)$$

where V^* is the exact value or cost and V is an estimate obtained using an ADP algorithm.

The performance results using the classical DP, the proposed ADP, and the conventional ADP approaches are summarized in Table 4, where the optimization gap is calculated based on the exact cost obtained using the classical DP. As can be seen, it takes 805 min (more than 13 h) to solve the microgrid dispatch problem exactly using the classical DP. With 100 iterations, the proposed ADP solves the problem in 6 min (more than 130 times faster than the classical DP), with an optimization gap of around 1%. On the other hand, the existing policy-iteration-based ADP approach shows a 2.5 times larger optimization gap with 67 times longer computation time. With the same number of iterations, the conventional double-pass and forward-pass ADP approaches result in much higher costs, showing optimization gaps of more than 27%. Note that the computation time reported in the table for the ADP approaches is the time required for training, and online decision-making time is almost the same for all the ADP approaches, at around 1 s.

Table 4
Performance comparison for deterministic microgrid dispatch.

Approach	Cost ($)	Gap	Time (min)
Classical DP	81.9	–	805
Proposed ADP	82.8	1.1%	6
Policy-iteration-based ADP	84.3	2.9%	403.8
Double-pass ADP	104	27%	6
Forward-pass ADP	105.2	28.5%	5.3

To better see the convergence process, the cost obtained from the proposed ADP approach is compared with other ADP approaches in terms of number of iterations and computation time. The performance of the proposed ADP approach is compared with the double-pass and forward-pass ADP approaches in terms of the number of iterations due to their competitive training time. On the other side, the computation time of the existing policy-iteration-based ADP approach is considerably higher than the proposed ADP approach, and they are compared in terms of computation time. The results are plotted in Fig. 5. As can be seen, the proposed policy-based exploration strategy significantly improves learning efficiency, approaching the value function faster and more accurately compared to the double-pass and forward-pass ADP approaches. The proposed ADP approach can be trained and used to get a solution much faster than the existing policy-iteration-based ADP approach and provides the minimum operation cost in the comparative study. Key observations are highlighted as follows:

- Compared with the double-pass and forward-pass ADP approaches, the proposed ADP with customized exploration exhibits significantly improved performance even at the beginning of the iteration process. After one iteration, the optimization gap of the proposed method is 58% lower than the conventional ADP approaches without customized exploration. This difference increases to 90% after eight iterations.
- In general, the value or cost improvement speed decreases as the optimization gap becomes smaller. With the conventional ADP approaches, after 75 iterations, the cost reduces to around $104, representing more than a 27% optimization gap. The performances of conventional ADP approaches are also evaluated by increasing the number of iterations. We observe that even at such a large gap, the cost curve becomes flat for many iterations, and only another 12% reduction is achieved after another 925 iterations. On the other hand, even with a smaller value after eight iterations, the optimization gap decreases at a much faster speed using the proposed method and arrives at 1% after another 92 iterations.
- In comparison between the proposed ADP and policy-iteration-based ADP approaches, the existing policy-iteration-based ADP approach takes 48 times more computation time to output a solution (zeros in initial time steps represent no solution). In 5 min of training, the proposed ADP approach shows a 14% improvement in operation costs.

Dispatch results from the proposed ADP are plotted in Fig. 6, including DG and BESS power as well as BESS SOC. As can be seen, DGs are only dispatched in the early morning and from evening to midnight, when RG is not enough to meet the load. BESSs are charged when the net load is negative, and are mainly discharged around the evening time when the net load is the highest to maximize the benefits of energy shifting, considering losses and degradation cost. In particular, because DG 1 is the most cost-effective among the three DGs, it is always dispatched first and operated at or close to the rated power when it is ON. For the same power output, DG 2 is the most expensive but with a lower minimum loading level compared with DG 3, and therefore can be dispatched when the desired power output is beyond the operating range of DG 3. DG 3 is only dispatched from evening to midnight, when the net load is high enough. BESS 1 has a lower degradation cost and a

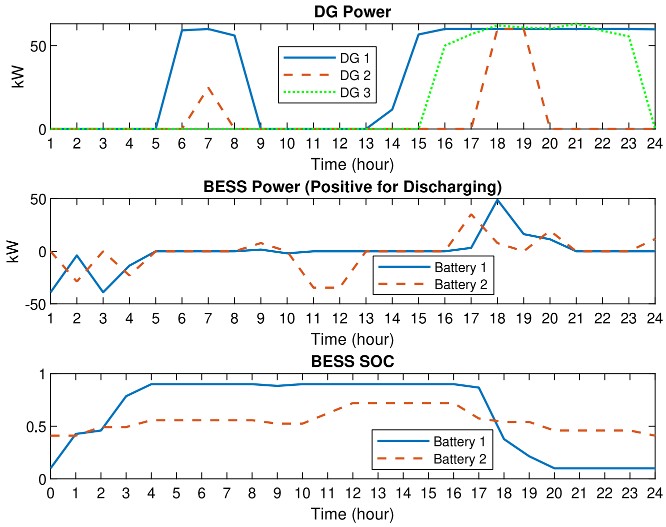


Fig. 6. Microgrid dispatch results with the proposed ADP approach.

Table 5
Distributions of forecast errors.

No.	Load	Renewable
1	$U(-3,3)$	$U(-5,5)$
2	$N(0,3)$	$U(-5,5)$
3	$U(-3,3)$	$N(0,1)$
4	$N(0,1.5)$	$N(0,2)$

higher round-trip efficiency, but shorter duration. It is typically used for intraday energy shifting for the microgrid. With a larger energy capacity, BESS 2 can be used for both intraday and interday energy shifting, as long as the benefits are large enough compared to the energy losses and degradation cost. With the limited energy to shift on the example day, BESS 1 is utilized first. It is charged until the maximum SOC is reached at the beginning of the day and discharged around the evening time. BESS 2 is charged with the remaining excess energy from RG and the energy capacity is not fully used on the example day. Note that BESSs can also be charged using DG power in addition to RG if the benefits from energy shifting are high enough compared to losses and degradation cost, which is not the case in this particular example. With the conventional ADP, BESSs and DGs are poorly dispatched, as indicated by the high optimization gaps. For example, some RG is dumped instead of being used to charge BESSs. In some hours, one BESS is discharging while the other one is charging, causing unnecessary losses. These results are not very meaningful and therefore are omitted here to conserve space.

5.2. Stochastic environment

The proposed and conventional ADP algorithms were also evaluated and compared in a stochastic environment, considering uncertainties from load and RG. The load and RG forecasts are given in Fig. 3. We designed four test cases with different probability distribution functions and parameters for load and RG forecast errors, as listed in Table 5, where $U(g,h)$ represents a uniform distribution between $g\%$ and $h\%$ of the forecast value, and $N(p,q)$ represents a normal distribution with a mean value of $p\%$ and a standard deviation $q\%$ of the forecast value. These distributions were used to generate samples for both training and testing. Due to the high computation cost, we exclude the policy-iteration-based ADP approach for comparison in this case study. For training, 2000 iterations were used for the proposed and conventional ADP approaches to generate the post-decision VFA. For testing, 500 simulations were performed in each case.

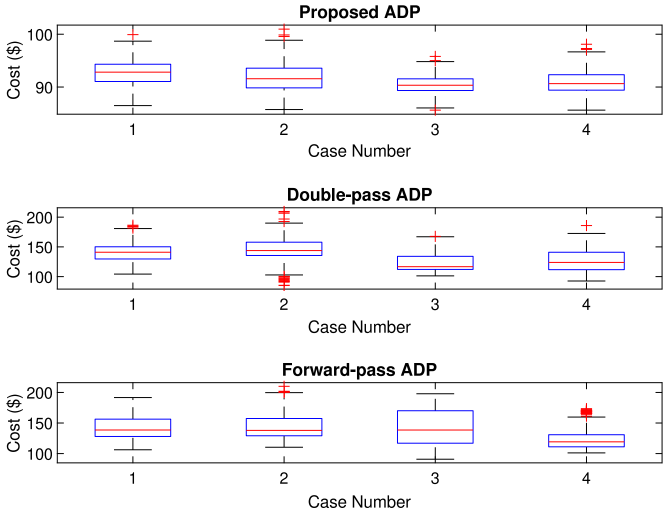


Fig. 7. Box plots of microgrid operation costs for the proposed ADP, double-pass ADP, and forward-pass ADP approaches.

The computation time is about the same for both algorithms. Fig. 7 provides the box plots of the microgrid operation cost to compare the proposed ADP with the conventional ADP approaches in a statistical manner. As can be seen, the proposed ADP results in lower microgrid operation cost than the conventional ADP approaches in all four cases. With the proposed ADP, the mean values of the cost are between $90 and $93. With the conventional ADP approaches, these values are between $121 and $145, about 40–50% higher. It is quite obvious that the proposed method shows much better performance compared with the conventional ADP without the customized exploration.

6. Conclusion and future work

This paper presented an innovative ADP approach with policy-based exploration for microgrid dispatch under uncertainties. Based on the underlying properties of the microgrid dispatch problem, we proposed an approximate optimization to speed up exploitation and a policy-based exploration to enhance exploration capability, and thereby to learn the value function faster and more accurately. The proposed method was validated and evaluated through case studies in both deterministic and stochastic environments. The results showed that the proposed approach outperforms the existing ADP approaches in terms of both optimization gap and solution time. One area of future work is to design parametric and nonparametric models for value function approximation and incorporate them into the proposed ADP. Another interesting research direction is to develop ADP based on the policy iteration strategy for microgrid and battery energy storage system dispatch.

CRediT authorship contribution statement

Avijit Das: Methodology, Software, Validation, Formal analysis, Writing – original draft. **Di Wu:** Conceptualization, Methodology, Formal analysis, Writing – original draft. **Zhen Ni:** Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Arboleya P, Gonzalez-Moran C, Coto M, Falvo MC, Martirano L, Sbordone D, Bertini I, Pietra BD. Efficient energy management in smart Micro-Grids: ZERO grid impact buildings. IEEE Trans Smart Grid 2015;6(2):1055–63. http://dx.doi.org/10.1109/TSG.20152392071.

[2] Wu D, Ma X, Huang S, Fu T, Balducci P. Stochastic optimal sizing of distributed energy resources for a cost-effective and resilient microgrid. Energy 2020;198:117284. http://dx.doi.org/10.1016/j.energy.2020117284.

[3] Liang H, Zhuang W. Stochastic modeling and optimization in a microgrid: A survey. Energies 2014;7(4):2027–50. http://dx.doi.org/10.3390/en7042027.

[4] Hajiamoosha P, Rastgou A, Bahramara S, Bagher Sadati SM. Stochastic energy management in a renewable energy-based microgrid considering demand response program. Int J Electr Power Energy Syst 2021;129:106791. http://dx.doi.org/10.1016/j.ijepes.2021106791.

[5] Nguyen TA, Crow M. Stochastic optimization of renewable-based microgrid operation incorporating battery operating cost. IEEE Trans Power Syst 2016;31(3):2289–96. http://dx.doi.org/10.1109/TPWRS.20152455491.

[6] Schulze T, McKinnon K. The value of stochastic programming in day-ahead and intra-day generation unit commitment. Energy 2016;101:592–605. http://dx.doi.org/10.1016/j.energy.201601090.

[7] Mavromatidis G, Orehounig K, Carmeliet J. Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach. Appl Energy 2018;222:932–50. http://dx.doi.org/10.1016/j.apenergy.201804019.

[8] Henrion R, Römisch W. Problem-based optimal scenario generation and reduction in stochastic programming. Math Program 2018;1–23. http://dx.doi.org/10.1007/s10107-018-1337-6.

[9] Moradi H, Esfahanian M, Abtahi A, Zilouchian A. Optimization and energy management of a standalone hybrid microgrid in the presence of battery storage system. Energy 2018;147:226–38. http://dx.doi.org/10.1016/j.energy.201801016.

[10] Moazeni S, Miragha AH, Defourny B. A risk-averse stochastic dynamic programming approach to energy hub optimal dispatch. IEEE Trans Power Syst 2019;34(3):2169–78. http://dx.doi.org/10.1109/TPWRS.20182882549.

[11] Forootani A, Iervolino R, Tipaldi M. Applying unweighted least-squares based techniques to stochastic dynamic programming: Theory and application. IET Control Theory Appl 2019;13(15):2387–98.

[12] Bertsekas DP. Reinforcement learning and optimal control. Athena Scientific Belmont, MA; 2019.

[13] Sammut C, Webb GI. Encyclopedia of machine learning. Springer Science & Business Media; 2011, http://dx.doi.org/10.1007/978-0-387-30164-8_870.

[14] Yu H, Bertsekas DP. Convergence results for some temporal difference methods based on least squares. IEEE Trans Automat Control 54(7):1515–31. http://dx.doi.org/10.1109/TAC.20092022097.

[15] Powell WB. Approximate dynamic programming: solving the curses of dimensionality. second ed.. John Wiley & Sons; 2011, http://dx.doi.org/10.1002/9781118029176.

[16] Pietrabissa A, Priscoli FD, Di Giorgio A, Giuseppi A, Panfili M, Suraci V. An approximate dynamic programming approach to resource management in multi-cloud scenarios. Internat J Control 2017;90(3):492–503. http://dx.doi.org/10.1080/0020717920161185802.

[17] Löhndorf N, Minner S. Optimal day-ahead trading and storage of renewable energies—an approximate dynamic programming approach. Energy Syst 2010;1(1):61–77. http://dx.doi.org/10.1007/s12667-009-0007-4.

[18] Li H, Zeng P, Zang C, Yu H, Li S. An integrative DR study for optimal home energy management based on approximate dynamic programming. Sustainability 2017;9(7):1248. http://dx.doi.org/10.3390/su9071248.

[19] Zeng P, Li H, He H, Li S. Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning. IEEE Trans Smart Grid 2019;10(4):4435–45. http://dx.doi.org/10.1109/TSG.20182859821.

[20] Das A, Ni Z. A computationally efficient optimization approach for battery systems in islanded microgrid. IEEE Trans Smart Grid 2018;9(6):6489–99. http://dx.doi.org/10.1109/TSG.20172713947.

[21] Shuai H, Fang J, Ai X, Tang Y, Wen J, He H. Stochastic optimization of economic dispatch for microgrid based on approximate dynamic programming. IEEE Trans Smart Grid 2019;10(3):2440–52. http://dx.doi.org/10.1109/TSG.20182798039.

[22] Li Z, Wu L, Xu Y, Moazeni S, Tang Z. Multi-stage real-time operation of a multi-energy microgrid with electrical and thermal energy storage assets: A data-driven MPC-ADP approach. IEEE Trans Smart Grid 2022;13(1):213–26. http://dx.doi.org/10.1109/TSG.20213119972.

[23] Sun Q, Wu Z, Gu W, Zhu T, Zhong L, Gao T. Flexible expansion planning of distribution system integrating multiple renewable energy sources: An approximate dynamic programming approach. Energy 2021;226:120367. http://dx.doi.org/10.1016/j.energy.2021120367.

[24] Xue X, Ai X, Fang J, Yao W, Wen J. Real-time schedule of integrated heat and power system: A multi-dimensional stochastic approximate dynamic programming approach. Int J Electr Power Energy Syst 2022;134:107427. http://dx.doi.org/10.1016/j.ijepes.2021107427.

[25] Keerthisinghe C, Verbič G, Chapman AC. A fast technique for smart home management: ADP with temporal difference learning. IEEE Trans Smart Grid 2018;9(4):3291–303. http://dx.doi.org/10.1109/TSG.20162629470.

[26] Salas DF, Powell WB. Benchmarking a scalable approximate dynamic programming algorithm for stochastic control of grid-level energy storage. INFORMS J Comput 2017;30(1):106–23. http://dx.doi.org/10.1287/ijoc.20170768.

[27] Jiang DR, Pham TV, Powell WB, Salas DF, Scott WR. A comparison of approximate dynamic programming techniques on benchmark energy storage problems: Does anything work? In: IEEE symposium on adaptive dynamic programming and reinforcement learning. 2014, p. 1–8. http://dx.doi.org/10.1109/ADPRL.20147010626.

[28] Shuai H, Fang J, Ai X, Wen J, He H. Optimal real-time operation strategy for microgrid: An ADP-based stochastic nonlinear optimization approach. IEEE Trans Sustain Energy 2019;10(2):931–42. http://dx.doi.org/10.1109/TSTE.20182855039.

[29] Powell WB, Meisel S. Tutorial on stochastic optimization in energy–part II: An energy storage illustration. IEEE Trans Power Syst 2016;31(2):1468–75. http://dx.doi.org/10.1109/TPWRS.20152424980.

[30] Keerthisinghe C, Chapman AC, Verbič G. Energy management of PV-storage systems: Policy approximations using machine learning. IEEE Trans Ind Inform 2019;15(1):257–65. http://dx.doi.org/10.1109/TII.20182839059.

[31] Powell WB. From reinforcement learning to optimal control: A unified framework for sequential decisions. In: Vamvoudakis K, Wan Y, Lewis F, Cansever D, editors. Handbook of Reinforcement Learning and Control. Studies in systems, decision and control, Cham: Springer; 2021, p. 29–74. http://dx.doi.org/10.1007/978-3-030-60990-0_3, Handbook of reinforcement learning and control.

[32] Puterman ML. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons; 2014, http://dx.doi.org/10.1002/9780470316887.

[33] Wu D, Jin C, Balducci P, Kintner-Meyer M. An energy storage assessment: Using optimal control strategies to capture multiple services. In: Proceedings of the IEEE power and energy society general meeting. Denver, CO; 2015, p. 1–5. http://dx.doi.org/10.1109/PESGM.20157285820.

[34] Wu D, Ma X. Modeling and optimization methods for controlling and sizing grid-connected energy storage: A review. Curr Sustain/Renew Energy Rep 2021;8(2):123–30. http://dx.doi.org/10.1007/s40518-021-00181-9.

[35] Pourbehzadi M, Niknam T, Aghaei J, Mokryani G, Shafie-khah M, Catalão J. Optimal operation of hybrid AC/DC microgrids under uncertainty of renewable energy resources: A comprehensive review. Int J Electr Power Energy Syst 2019;109:139–59. http://dx.doi.org/10.1016/j.ijepes.201901025.

[36] Jiang DR, Powell WB. An approximate dynamic programming algorithm for monotone value functions. Oper Res 2015;63(6):1489–511. http://dx.doi.org/10.1287/opre.20151425.

[37] Powell WB, Meisel S. Tutorial on stochastic optimization in energy–part I: Modeling and policies. IEEE Trans Power Syst 2016;31(2):1459–67. http://dx.doi.org/10.1109/TPWRS.20152424974.

[38] Mes MRK, Rivera AP. Ch. approximate dynamic programming by practical examples. In: Approximate dynamic programming by practical examples. Springer International Publishing; 2016, p. 63–101. http://dx.doi.org/10.1007/978-3-319-47766-4_3.

[39] Commercial and residential hourly load profiles for all TMY3 locations in the united states. National Renewable Energy Laboratory; 2014, http://dx.doi.org/10.25984/1788456, http://data.openei.org/submissions/153.

[40] Freeman J, Blair N, Guittet D, Boyd M, Mirletz B, et al. System Advisor Model, Available: https://sam.nrel.gov/.

[41] Wu D, Yang T, Stoorvogel AA, Stoustrup J. Distributed optimal coordination for distributed energy resources in power systems. IEEE Trans Autom Sci Eng 2017;14(2):414–24. http://dx.doi.org/10.1109/TASE.20162627006.

[42] Mongird K, Viswanathan V, Alam J, Vartanian C, Sprenkle V. Grid energy storage technology cost and performance assessment. Tech. rep. DOE/PA-0204, Pacific Northwest National Laboratory; 2020.

[43] Sajedian I, Lee H, Rho J. Double-deep Q-learning to increase the efficiency of metasurface holograms. Sci Rep 2019;9:10899. http://dx.doi.org/10.1038/s41598-019-47154-z.