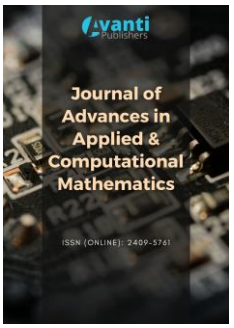
Published by Avanti Publishers

Journal of Advances in Applied & Computational Mathematics

ISSN (online): 2409-5761



Adaptive Dynamic Programming and Its Application to Economic Dispatch in Microgrid: A Brief Overview

Zitao Chen[1], Quanbin Deng[1] and Kairui Chen[2,*]

[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 51006, China

ARTICLE INFO

Article Type: Review Article

Keywords:
Microgrid
Optimal control
Economic dispatch
Adaptive dynamic programming

Timeline:
Received: November 11, 2021
Accepted: February 02, 2022
Published: May 24, 2022

Citation: Chen Z, Deng Q, Kairui C. Adaptive Dynamic Programming and Its Application to Economic Dispatch in Microgrid: A Brief Overview. J Adv App Comput Math. 2022; 9: 13-31.

DOI: https://doi.org/10.15377/2409-5761.2022.09.2

ABSTRACT

Both adaptive dynamic programming and other intelligent algorithms can solve the economic dispatch problem in the microgrid. Adaptive dynamic programming can reduce the computational burden, which the intelligent algorithms suffer from, by using function approximation structure to approximate performance index function. In recent years, it has been also widely used in economic dispatch in the microgrid. In this article, we introduce some recent research trends within the field of adaptive dynamic programming based economic dispatch. Adaptive dynamic programming is firstly reviewed. Then, the current research works about adaptive dynamic programming based economic dispatch are summarized and compared. Furthermore, we point out some topics for future studies.

*Corresponding Author
Email: kraychen@139.com
Tel: +86-138-2446-9545

1. Introduction

The microgrid has received more and more attention as an indispensable part of the construction of smart grids. Modular distributed energy sources are integrated into the microgrid, such as fuel cells, wind, solar, storage devices, and loads, which improve grid reliability and supply sustainable and quality electric power. However, it is the suitable economic dispatch that ensures normal operations and optimality. Hence, the economic dispatch of the microgrid is vital [1]. However, challenges are brought with the topology of the microgrid becoming more and more complex. Computational intelligence technique is necessary for energy management in the future [2].

The economic dispatch of the microgrid can be formulated as an optimization problem with some constraints. The optimization can be classified into static and dynamic optimization. Generally, the traditional method, such as linear programming, and intelligent method, such as particle swarm optimization, belong to static optimization. The constraints of static optimization are presented by algebraic equalities and inequalities. Authors in [3] propose a price-based power scheduling scheme for a community-scale microgrid to maximize the expected benefit while minimizing the operating cost. It should be noted that accurate mathematical models are required in these methods, but the uncertainties brought by the renewable energies and load demands lead to the loss of accuracy. Furthermore, the performance of these methods depends on the predictive error of uncertainty. To solve this problem, the authors in [4], energy storage units, and responsive loads are studied through analyzing their uncertainty natures, and a Monte Carlo simulation-based stochastic optimization method was proposed to account for the uncertainties. But computational pressure is rather large. The static optimization algorithm takes effect on the energy dispatch. However, they all suffer from high computational costs with the increase of scheduling horizons. On the other hand, these algorithms are not able to adapt to the change in user behavior patterns due to their static nature. Moreover, an accurate prediction for the residential load profile with high volatility is difficult, which will affect the optimality of these algorithms.

Reference [5] proposes a new class of fractional-order six-neuron bi-directional associative memory neural network containing multiple delays and proves the existence, uniqueness and boundedness of the neural network solution. Reference [6] proposed a two-bit-triggered control and built an improved fuzzy logic system to prove the boundedness of all system signals. Reference [7] established a fractional Oregonator model including time delay to describe the relationship between different chemical components. Reference [8] proposed a new fractional-order delayed financial crisis contagion model.

Feedback control theory is the means for developing human-engineered systems to guarantee performance and reliability [9, 10]. The optimal feedback control scheme discusses a dynamic optimization problem. The constraints are presented by differential equations or difference equations. Dynamic programming is useful to solve the optimal control sequence in the finite horizon problem. Authors in [11] present a probabilistic constrained approach to model the microgrid system and use dynamic programming to find the optimal day-ahead scheduling. However, dynamic programming is not very practical in real engineering problems. It is not robust, and the curse of dimension greatly increases the computational burden.

Adaptive dynamic programming (ADP), firstly proposed by Werbos, is a useful tool to design optimal controllers offline or online. The ADP algorithm aims to solve an optimal control law instead of a control sequence. Once the optimal control law is calculated, the control value at each time step can be directly computed. For this reason, the ADP algorithm does not suffer from the curse of dimension. Werbos proposed the basic framework of the ADP in [12]. Before 2008, the researchers in ADP community developed the algorithms based on the four basic structures proposed by Werbos, which are Heuristic Dynamic Programming (HDP), Action Dependent Heuristic Dynamic Programming (ADHDP), Dual Heuristic Dynamic Programming (DHP), Action Dependent Dual Heuristic Dynamic Programming (ADDHP) [13]. In this period, the ADP was used to deal with a challenging problem, such as the controlling problem of the inverted pendulum [14]. Until 2008, the authors in [15] proposed a zero-initialization value iteration algorithm firstly with convergence proof. From then on, the iterative forms of ADP receive great attention from people. In [16], the authors developed the value iteration algorithm for the general nonlinear

system, implemented by DHP. The authors in [17] proposed the semi-definitive-initialization based value iteration. The utility function is not required to be quadratic form. Furthermore, the authors in [18] developed a local value algorithm, which updates the subset of the state space instead of the whole state space. Policy iteration is another form of iterative ADP algorithm, which required an initial admissible control. The authors in [19] firstly developed a policy iteration for the discrete-time nonlinear system with convergence proof. The initial admissible control can be also obtained by the HDP structure. Reference [20] developed an adaptive self-triggering tracking control method.Reference [21] proposed an event-triggered prescribed build-time consistent adaptive compensation control method for a class of uncertain nonlinear systems with actuator faults. The ADP algorithms mentioned above have been applied to many problems, such as zero-sum games [22], multi-agent systems [23, 24], temperature control of water gas shift reaction [25], etc.

Characterized by strong abilities of self-learning and adaptivity, ADP is also considered to be the potential to solve the economic dispatch problem. For the first time, authors in [26] proposed an ADP based algorithm to make optimal dispatch decisions for residential EMS. In [27] 7 and [28], the authors consider the combination of renewable energy, extending the mathematical model. These algorithms establish the basic framework of the ADP-based EMS, implementing their algorithm with two feedforward neural networks to learn the optimal energy schedule. However, only with enough times of trials can the optimal dispatching solution be solved. ADP is applied to more complex scenes by other researchers. Considering the different user behaviors among four seasons, the authors in [29, 30] design multiple controllers for the EMS in various seasons. To overcome the uncertainty in the microgrid, the authors in [31 and 32] models the dynamics of the microgrid as a Markov process, proposing an optimal controller design method in the sense of expectation. The effectiveness of ADP-based EMS is verified in multiple nodes testbed. Furthermore, ADP-based EMS is also applied to appliance level control [33]. The above ADP-based method is consistent, implemented by the actor-critic structure. However, they search for the optimal parameters of the controller by trials and errors, which is sensitive to the initial state. Moreover, the convergence property of the ADP algorithm is still not discussed in a theoretical sense, which restricts its application. To improve and perfect the theoretical base of the ADP-based EMS, the authors in [42] firstly proposed a theoretical framework, named dual Q-Learning algorithm, analyzing the properties of the algorithm. Then, the ADP-based algorithm for EMS, which takes renewable energy into account, is proposed in turn [34, 35]. The convergence property ensures the effectiveness and reliability of the ADP algorithm. However, most of the convergence proof requires restricted assumption for the outer parameter, i.e. the periodicity of the load curve and electricity rate. Their performance of them may not satisfactory under the highly fluctuating load profile. It should be pointed out that the implementation of the theoretical framework is not limited to the feedforward neural network. In [36], the authors use the echo state network as the parametric structure instead of the feedforward neural network. The authors in [37] approximate the iterative value function by fuzzy structure.

As an algorithm with self-learning and self-adaptive capabilities, ADP has great potential in energy dispatching management of smart grids and can solve various energy dispatching management problems in smart grids. However, there is currently no article that systematically and comprehensively addresses this direction, which motivates our work. Therefore, we make a systematic review of ADP-based smart grid energy management in the past literature.

Even though ADP has been used to the economic dispatch for some research, the systematic comparison of these algorithms in the perspectives of the methodology is rare. This paper aims to summarize the state-of-the-art ADP algorithms and their application to the economic dispatch problem. The paper is organized as follows. In Sect. 2, we briefly describe the optimal control problem, and the dynamic programming algorithm is reviewed. In Sect. 3, the iterative ADP algorithms and their implementation are reviewed. In Sect. 4, the ADP based economic dispatch algorithms are summarized and compared. Section 5 gives some comments for future study. In Sect. 6, we have the conclusion.

2. Nonlinear Optimal Control Theory

In this section, the discrete-time optimal control problem is formulated, and a well-known dynamic programming algorithm is briefly reviewed, which is the fundamentals of the iterative ADP algorithm. Also, the so-

called curse-of-dimension problem is demonstrated so that the motivation of ADP can be clearly understood. For more details about optimal control theory, a reader can refer to [38].

2.1. State Feedback Control Scheme

A general feedback control system can be shown in Fig. (**1**).

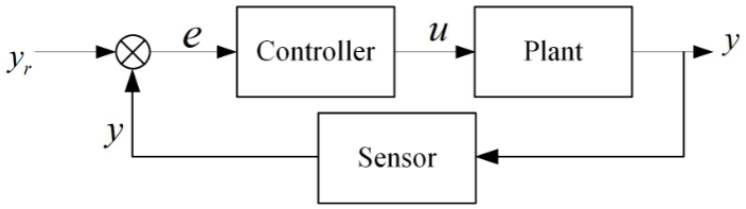


Figure 1: General feedback control scheme.

The feedback control scheme is depicted in Fig. (**1**). Adaptive dynamic programming aims to design an optimal feedback control law, which minimizes the user-defined performance index function. In the basic ADP theory, the reference input y_r is set to be zero, and the problem is an optimal regulator problem. State feedback is mostly considered in optimal control because state variables can well present the inside of the system. Seldom literature discuss output feedback control for general nonlinear system with ADP algorithm [39.

2.2. Problem Formulation

The mathematical model of the controlled plant is the state space equations defined as the form as eq. (1)

$$\dot{x}_t = \overline{F}(x_t, u_t), t \in [0, \infty) \tag{1}$$

where x_t is the state vector, defined within a compact set Ω, i.e. $x_t \in \Omega \subset R^n$,$u_t$ is the control vector, and $\overline{F}(.)$denotes the continuous-time system function. Most literature discusses deterministic nonlinear time-invariant dynamical systems because they cover most of the application areas. For the purpose of digital implementation, numerical computation, such as Runge-Kutta method, is used for the discretization of a system (1). A deterministic discrete-time time-invariant nonlinear system is defined as

$$x_{k+1} = F(x_k, u_k), k = 0,1,2,\ldots \tag{2}$$

where x_k is the state vector at a time k, $u_k \in R^m$ is the control vector. $F(.)$is the discrete-time system function. The optimality of the controller is defined by the user-defined performance index function as eq. (3)

$$J(x_0, \underline{u}_0) = \sum_{k=0}^{N-1} U(x_k, u_k) + \varphi(x_N) \tag{3}$$

where $U(.)$is the utility function, which is a positive function, and $\varphi(x_N)$ is the terminal performance index. $\underline{u}_0 = \{u_0, u_1, \cdots, u_{N-1}\}$ is a finite control sequence.

Equations (1) and (2) formulate the finite-horizon optimal control problem, i.e. $N < +\infty$. The problem turns into an infinite-horizon problem when$N \to \infty$. We reformulate the problem as the infinite-horizon form. Given the system model (2) and performance index function

$$J(x_0, \underline{u}_0) = \sum_{k=0}^{\infty} U(x_k, u_k) \tag{4}$$

Note that the discussion about the infinite-horizon optimal control problem should be under the following assumptions,

Assumption 1: The system function $x_{k+1} = F(x_k, u_k)$ is Lipschitz continuous for x_k and u_k.

Assumption 2: The system is absolutely controllable.

Assumption 3: $x_k = 0$is an equilibrium state of (4) under control $u_k = 0$, i.e. $F(0,0) = 0$. Furthermore, we assume $U(0) = 0$.

Remark 1: The infinite-horizon problem is essentially different from the finite-horizon one. Firstly, we not only minimize Jbut also require $J < +\infty$ the infinite horizon problem to ensure stability. The stability of the finite-horizon problem is not discussed, because N is finite, which forces the system stable. Secondly, the state feedback control sequence in infinite one is a time-invariant mapping, i.e $u_k = \mu(x_k)$, which is convenient for engineering implementation while the finite one is time-varying, i.e $u_k = \mu(x_k, N)$, because there is a super parametric N. We will mainly discuss the infinite horizon problem. Assumptions 1-3 are not necessary for the finite-horizon problem, because the system state x_k in a finite-horizon problem will not reach to ∞, that is, the system is stable whatever control policy is taken.

Remark 2: Actually, the utility function is to describe a reward for taking action $u_k = u(x_k)$. Intuitively, it can be designed for any form according to the special scenario of the application. For example, the utility function is set to a binary value for the control problem of the inverted pendulum. In other cases, quadratic forms utility function is adopted because it is brief and usually has clear physical meaning. A typical case is the LQR problem. In some real applications, the utility function is diverse. State variables and control variables may be coupled in the utility function [30].

2.3. Dynamic Programming

In this subsection, we will briefly review dynamic programming. Theoretically, dynamic programming can design an optimal controller for a complex nonlinear system. However, it is not so powerful in engineering applications since the so-called curse of dimension, which widely exists in the data science community. From a mathematical point of view, we can see the finite-horizon optimal control problem as a common optimization problem. Consider the state space equation (2) as a constraint, we have an optimization function

$$\begin{aligned} J(\underline{u}_0) = U(x_0, u_0) &+ U(F(x_0, u_0), u_1) + \cdots \\ &+ U(F\ (F\ (\ldots F(x_0, u_0) \ldots)\ , u_{N-2})\ , u_{N-1}) \\ &+ U(F\ (F\ (\ldots F(x_0, u_0) \ldots)\ , u_{N-1})) \end{aligned} \tag{5}$$

this is the extremum problem of multivariate function. The necessary condition of optimality is that the partial derivatives are equal to zero, i.e.

$$\begin{aligned} \frac{\partial J}{\partial u_0} &= 0 \\ \frac{\partial J}{\partial u_1} &= 0 \\ &\vdots \\ \frac{\partial J}{\partial u_{N-1}} &= 0 \end{aligned} \tag{6}$$

The solution space of Eq. (6) is a combination space. It is nearly impossible to solve nonlinear equations. However, we can find that we can solve the equations from bottom to top, which reduces the computational pressure. The reason why it works is that the solution of the ith equation is also of the $i-1$th equation ($i = 2,3,\cdots,N$). That is the famous "principle of optimality". Actually, the computational pressure in each time step is the same. This principle can be expressed as the famous Bellman equation

$$J^*(x_k) = \min_u \{U(x_k, u) + J^*(x_{k+1})\} \tag{7}$$

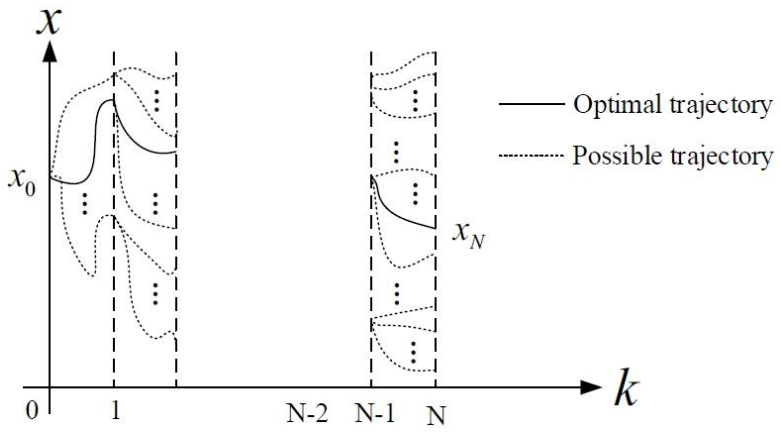


Figure 2: Forward-in-time method.

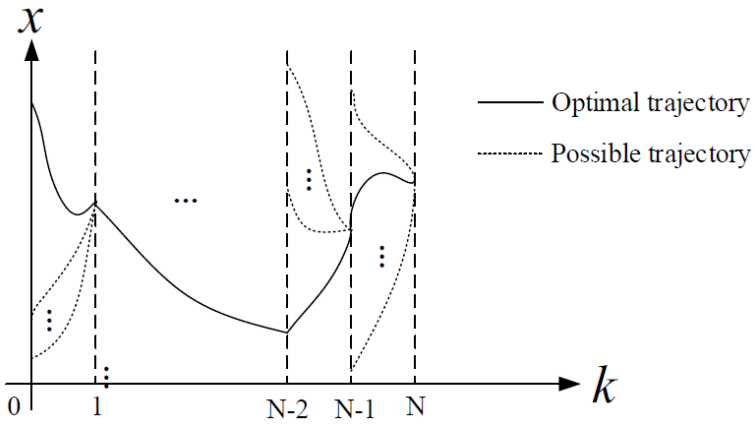


Figure 3: Forward-in-time method.

Figure (**2** and **3**) respectively show the process of searching the optimal trajectory forward in time and DP algorithm, from which we can see the forward-in-time method the number of possible trajectories grows at an exponential rate. DP algorithm, which is a back-ward in time method, well contracts the search space, and reduce the computational pressure

Although dynamic programming is theoretically a useful tool towards optimal control problems in theoretical meaning, it may not suitable for real control system control. Firstly, dynamic programming is a backwards-in-time method, which means that it cannot run the system until the control sequence $\underline{u}_0$, which does not consider the disturbance from the environment. Secondly, the computational pressure becomes very large when the dimension of the state vector or the number of time step increase. Finally, even the dynamic of the system (environment) is deterministic, some actual systems are so complex that they cannot exactly be written by formulations.

3. Iterative ADP Algorithm

The machine will still suffer from the curse of dimension although DP algorithm has well contracted the search space. From the Bellman Equation (7), if we know the optimal performance index function at a time $k+1$, i.e.$J^*(x_{k+1})$, the optimal control policy u_k can be directly calculated. However, it is hardly possible to get $J^*(x_{k+1})$ directly, because this is not causal. ADP indirectly seek the optimal performance index function $J^*(x)$ by an iterative value function $V_i(x)$. Also, the optimal state feedback control $\mu^*(x)$ is approximated by an iterative control policy $\mu_i(x)$. For this reason, the most popular algorithm in the ADP community is called Iterative Adaptive Dynamic Programming (IADP).

Value iteration and policy iteration are the two main schemes to solve the infinite horizon optimal control problem. It is believed that the value iteration requires less computation at the cost of missing the guarantee of system stability during the iteration process, i.e. the iterative control policy $\mu_i(x_k)$ may not be stable for some i. Compared with value iteration, the initial control law $\mu_0(x_k)$ for policy iteration should be an admissible control law.

3.1. Value Iteration

Initialize. Choose any semi-positive function $V_0(.)$.

Policy Improvement Step. The iterative control law is improved by

$$\mu_i(x_k) = arg \min_u \{U(x_k, u) + V_i(x_{k+1})\} \tag{8}$$

Value function Update Step. The value function is updated by

$$V_{i+1}(x_k) = U(x_k, \mu_i(x_k)) + V_i(x_{k+1}) \tag{9}$$

In addition, many recent papers have provided initial value function $V_0(.)$and the corresponding convergence analysis associated with the algorithms developed. In [15], the authors first propose the zero initialization value function, i.e. $V_0(.)$. Meanwhile, convergence proof is given. It is noted that the utility function is $U(x_k, x_k) = Q(x_k) + u_k^T R u_k$, and the system function is limited to the affine nonlinear system. In [16], the authors develop a value iterative algorithm for a non-affine nonlinear system with the quadratic form utility function $U(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$. Furthermore, the authors in [17] propose a very general value iteration algorithm. The plant is a general nonlinear system, and the utility function is only required to be positive definite.

Figure (**4**) is the sketch map showing the iterative process. The value function $V_i(x_k)$ in limiting iteration is asymptotically approximated the optimal performance index function $J^*(x_k)$. Meanwhile, the iterative control law is asymptotically approximate the optimal control law $\mu^*(x_k)$. Specifically, $\|V_{i+1}(x_k) - J^*(x_k)\|_\infty \leq \|V_i(x_k) - J^*(x_k)\|_\infty$.

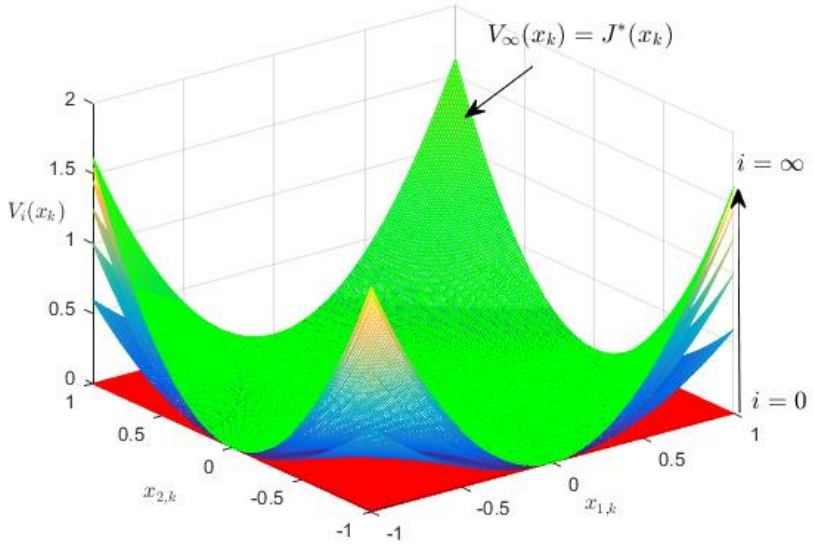


Figure 4: The sketch map of iterative value function.

3.2. Policy Iteration

Definition 1: (Admissible Control) A control policy $\mu(x_k)$ is a admissible control with respect to 4 on Ω if the state feedback control $\mu(x_k)$ not only stabilizes system 4 but also make $J(x_k)$ finite for all $x_k \in \Omega$.

Initialize. Choose any admissible control law $\mu_0(.)$.

Policy Evaluation Step. The iterative control law $\mu_i(x_k)(i = 0,1,\cdots)$ is evaluated by

$$V_i(x_k) = U(x_k, \mu_i(x_k)) + V_i(x_{k+1}) \tag{10}$$

Policy Improvement Step. The value function is updated by

$$\mu_{i+1}(x_k) = arg \min_u \{U(x_k, u) + V_i(x_{k+1})\} \tag{11}$$

The authors in [19] propose an algorithm to obtain the initial admissible control law $\mu_0(.)$. It should be noted that it is essentially a trial-and-error method. For the LQR problem, the initial admissible control law is very easy to obtain by choosing a proportional gain matrix K, s.t. The eigenvalues of $A - BK$ are all less than zero. However, for the complex nonlinear system, it is very hard to obtain the admissible $\mu_0(x_k)$.

The policy evaluation (14) is a fixed point equation. To solve the policy evaluation $V_i(.)$, we firstly assume that $V_i(x_k) = \Phi_0(x_k)$ for ith iteration, then update the $\Phi_j(x_k)$ by

$$\Phi_{j+1}(x_k) = U(x_k, \mu_i(x_k)) + \Phi_j(x_{k+1}) \tag{12}$$

Since the map $\Gamma: \Phi_j(x_k) \to \Phi_{j+1}(x_k)$ is a contract map. Let $J \to \infty$, we have

$$\Phi_\infty(x_k) = U(x_k, \mu_i(x_k)) + \Phi_\infty(x_{k+1}) \tag{13}$$

then, the policy evaluation $V_i(x_k) = \Phi_\infty(x_k)$.

Remark 3: Compared with the value iteration algorithm, policy iteration requires the iterative control policy $\mu_0(x_k)$, containing prior knowledge, as an admissible control, but does not require the positive semi-definite value function. Value iteration requires less information than policy iteration because the initial value function is arbitrary. Since policy iteration carries more information at the start of the iteration, policy iteration converges faster than value iteration in most cases.

3.3. Generalized Policy Iteration

In subsection III-A and III-B, the difference between value iteration and policy iteration is illustrated. It is worthwhile mentioning that they are consistent to some extent.

Policy Evaluation Step in generalized policy iteration. The iterative control law $\mu_i(x_k)(i = 0,1,\cdots)$ is evaluated by

$$V_i(x_k) = U(x_k, \mu_i(x_k)) + V_i(x_{k+1}) \tag{14}$$

where for $J = 1,2,\cdots, N_j - 1$,

$$\Phi_{j+1}(x_k) = U(x_k, \mu_i(x_k)) + \Phi_j(x_{k+1}) \tag{15}$$

and $V_i(x_k) = \Phi_{N_j}(x_k)$.

It should be noted that generalized policy iteration is meaningful for the real application. The number of iterations can not reach infinity.

3.4. Implementation for Iterative ADP Algorithm

In the previous part of this paper, we review the value iteration and policy iteration algorithms. It was assumed in the convergence proof that the iterative control law and value function update equations can be exactly solved

at each iteration. Actually, these equations are difficult to solve for complex nonlinear systems. Given that, for implementation purposes, function approximation structures are used to approximate the iterative solutions.

According to the problem formulated in section II, we know that parametric structures are required. Firstly, the system function is sometimes unknown or so complex that we can not write it. So a model network is required to identify the system model based on an online or offline data set. Secondly, we cannot calculate the performance index function $J^*(x_k)$ at time k because it is an infinite series. Therefore, a critic network is also required to approximate the $J^*(x_k)$. Finally, the optimal control $\mu(x_k)$ is also needed to be approximated by a neural network to ensure its optimality.

3.4.1. Function Approximation Structures

Neural networks, fuzzy structure, quadratic form, etc. are widely used in the implementation of the ADP algorithm. The most used parametric structure is a feedforward network with the activation function $\phi_{c,i}$ and $\phi_{a,i}$ is defined as the tanh function. All the parameters in the three neural networks are undetermined. The model network should be well trained firstly because it is the plant in the control system. Then the performance index function should be approximated by the critic network. Finally, the action network can be adjusted according to the approximated performance index function. The model network is like the four limbs of the athlete. While the critic network is the referee, and the action network is the brain of the athlete.

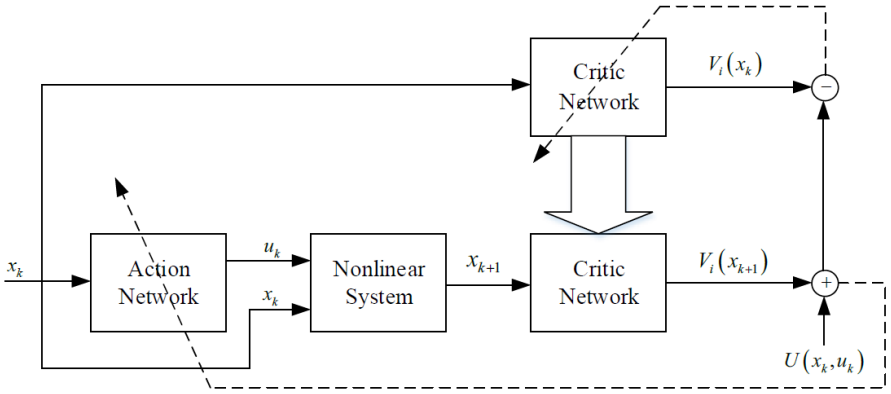


Figure 5: The HDP structure diagram.

Now we define the model network as

$$\hat{x}_{k+1} = W_{m2\phi_m}(W_{m1z_k} + b_m) \tag{16}$$

where $z_k = [x_k, u_k]^T$.

The parametric structures for iterative value function, i.e. critic network, can be consistently defined as

$$\hat{V}_i(x_k) = W_{c,i\phi_c}(x_k) \tag{17}$$

Similarly, the action network can be consistently defined as

$$\hat{\mu}_i(x_k) = W_{a,i\phi_a}(x_k) \tag{18}$$

where $W_{c,i}$,$W_{a,i}$ are undetermined coefficients. It should be noted that the approximation error is unavailable, i.e.

$$V_i(x_k) = \hat{V}_i(x_k) + \varepsilon_{c,i}(x_k) \tag{19}$$

and

$$\mu_i(x_k) = \hat{\mu}_i(x_k) + \varepsilon_{a,i}(x_k) \tag{20}$$

where $\varepsilon_{c,i}(x_k)$ and $\varepsilon_{a,i}(x_k)$ denote the approximation errors.

In some literature [40], a quadratic form structure is also used as the parametric structure. The activation function is a vector with $\frac{n(n+1)}{2}$ dimensions

$$\phi(x) = [x_1^2, x_1x_2, \dots, x_1x_n, x_2^2, x_2x_3, \dots, x_2x_n, \dots, x_{n-1}x_n, x_n^2]^T \tag{21}$$

the weight vector $W \in R^{\frac{n(n+1)}{2}}$

The connection between the three networks is shown in Fig. (**5**), which is the HDP (Heuristic Dynamic Programming) structure.

We take VI algorithm as an example to illustrate the training strategy. Before the iteration of the critic network and action network, we train the model network offline with pre-collected data set.

Remark 4: In a model-based optimal control problem, we will train the model network before training another network, and then keep its weights unchanged. Therefore, we need off-line data. In a model-free optimal control problem, the model network is not required, i.e. ADHDP(Action Dependent Heuristic Dynamic Programming) discussed later.

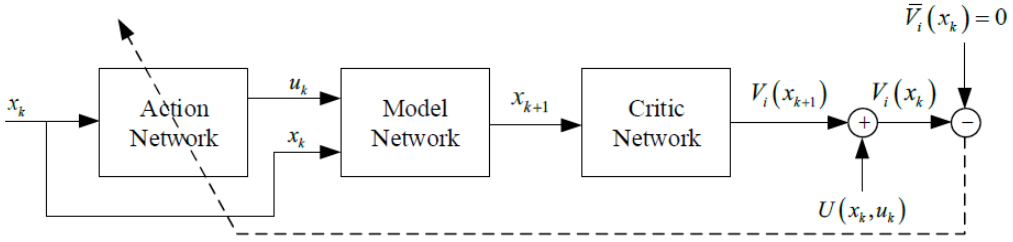


Figure 6: Action network training process.

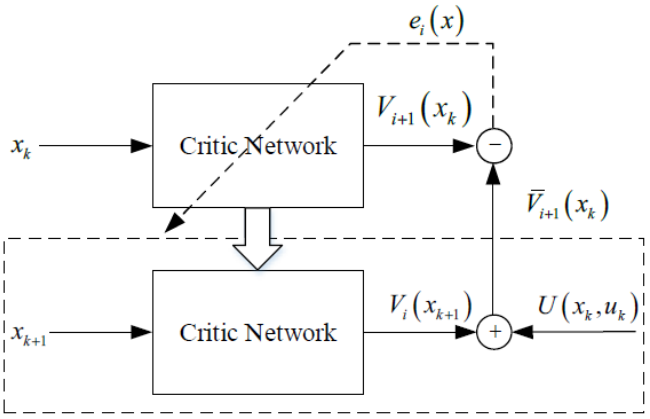


Figure 7: Critic network training process.

Then, we start the iteration. The optimal performance index function is unavailable, resulting in the failure in training critic network by traditional supervise learning. Firstly, initialize the critic network satisfy that $\hat{V}_0(x) \geq 0, \forall x \in \Omega \subset R^n$, which is a semi-positive definite function we guess. The approximated iterative control policy $\hat{u}_0(x)$ is obtained by tuning the weights of the action network, shown in Fig. (**6**).

$$W_{a,i}^{j+1} = W_{a,i}^{j} - \Delta W_{a,i}^{j} \tag{22}$$

the $\hat{u}_0(x)$ approximates $u_0(x)$ with bounded error

$$\varepsilon_{a,0}(x) = u_0(x) - \hat{u}_0(x) \tag{23}$$

Afterwards, the critic network would be trained, shown in Fig. (**7**).

the updating rules of action network weights is

$$W_{c,i}^{j+1} = W_{c,i}^{j} - \Delta W_{c,i}^{j} \tag{24}$$

the $\hat{V}_1(x)$ approximates $V_1(x)$ with bounded error

$$\varepsilon_{c,1}(x) = V_1(x) - \hat{V}_1(x) \tag{25}$$

For $i = 1,2,\cdots$, the target of action network and critic network are

$$\bar{\mu}_i(x) = \arg\min_u U(x_k, u) + V_i(\hat{x}_{k+1}) \tag{26}$$

$$\bar{V}_{i+1}(x_k) = U(x_k, \hat{u}_i(x_k)) + \hat{V}_i(\hat{x}_{k+1})$$

Then, the iterative control policy and iterative value function can be updated and satisfy that

$$\hat{u}_i(x) = \bar{u}_i(x) + \varepsilon_{c,i}(x), i = 1,2 \ldots \tag{27}$$

$$\hat{V}_i(x) = \hat{V}_i(x) + \varepsilon_{c,i}(x), i = 2,3 \ldots$$

Remark 5: Different from the theoretical analysis in subsection 3.1, there exists unavoidable error in the real application of the iterative ADP algorithm. In most cases, the numerical error is so small that it will not make the result bad. To make sure that the ADP algorithm will converge with the finite error at each iteration, the authors in [38] proposed a finite-error-bound ADP algorithm, helping to control the error in using the iterative ADP method.

Remark 6: The NN implementation for iterative ADP algorithm in this subsection is essentially an offline method, more details of which will be discussed later. The difference between the offline algorithm and the online algorithm is whether we iterate at time k or time $k+1$. In the offline algorithm, we can estimate the state trajectory at a time $k+1$, i.e. $\hat{x}_{k+1}$ through the model network. An online algorithm is a model-free scheme since we can collect the system trajectory online. The model network is not adopted in the online algorithm so we can get the state until time $k+1$.

3.4.2. Offline and Online Implementation

ADP algorithm can be implemented offline or online. In the offline algorithm, a state variable set

$$\chi = \{x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(P)}\} \tag{28}$$

is chosen. The sample $x_k^{(1)}$ is uniformly distributed in the compact set Ω. According to 26, the target of the action network and critic network can be calculated, then the problem is supervised learning at each iteration. Batch mode or non-batch mode for weights tuning can be chosen depending on a special case. Theoretically, offline implementation ensures synchronous convergence between the critic network and the action network.

Rather than offline implementation, online ADP also receives attention. On one hand, the offline method is mainly based on system identification theory or big data, which requires lots of data. However, the computational pressure of such data sometimes is also unbearable, while online implementation reduces the computational cost. On the other hand, online tuning the weights of the approximator may really realize the adaptability of the system. In an online ADP algorithm, the system state pairs (x_k, x_{k+1}) will be used to iterate at time $k+1$. So we desire that after the system runs for some time, the weights of the critic network and action network will converge.

The online algorithm updates each iteration step i at each time step k, which implies $i = k$, while the offline algorithm requires both iteration index i and time index k and update at each iteration step i but not each real-time step k.

In both offline and online algorithms, we must make sure the stability of the neural network. The proof indicates that we can only ensure stability when fixing the weight of the first layer in NN. In the present literature, authors choose an arbitrary weight matrix in the first layer. Here we want to note that the powerful ability of neural networks is from its flexibility in tuning basis function, so only tuning the second layer will limit the approximate ability of the neural network unless the arbitrary weight matrix is luckily chosen. In some cases, it even performs worse than the quadratic form parametric structure, which indicates that the structure of the parametric structure is also as important as the stability of the neural network. In the offline algorithm, the data set will help NN learn well than the online case, but it is not convergent in mathematical meaning. In engineering practice, people assume that it converges if the data set is large enough and the state trajectory is acceptable. In summary, there is a gap between the theory and application of ADP. Given that, it is meaningful to develop the related initialization algorithm of the neural network to make sure the convergence of NN for the iterative ADP algorithm, especially in online cases.

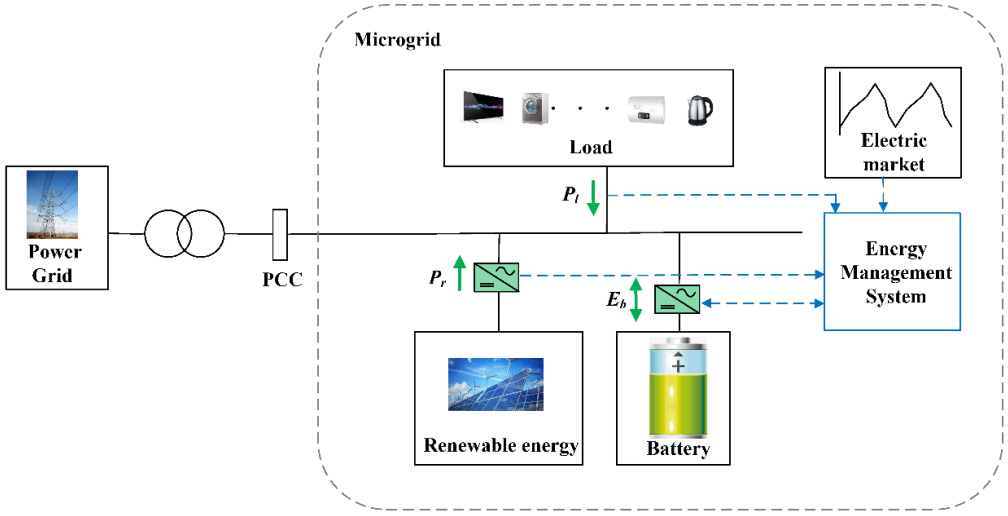


Figure 8: The structure of microgrid.

4. ADP With Application to Economic Dispatch

A smart grid is a very stomatic and highly nonlinear system. It consists of multiple power stations, step-up substations, long-distance transmission, step-down distribution stations, power load, etc. The microgrid is a new small-scale power system, which can work on grid-connected mode and isolated island mode, depending on whether it connects to the PCC (Point of Common Coupling). There are many distributed equipment in the microgrid, including distributed generators, storage systems, etc. Also, the loads can be classified into many classes such as industrial load, household load, etc. The energy flow among these elements in microgrids should be well scheduled. In this section, we will review the ADP based microgrid control technique.

4.1. Structure of Microgrid

A general structure of the smart microgrid system is shown in Fig. (**8**), which is comprised of a utility grid, a battery storage system, solar photovoltaic (PV) panels, wind-turbine generators, and loads. The central controller manages energy flow. The direction of electricity transmission (bidirectional/unidirectional) is shown in Fig. (**8**). Wind turbines and photovoltaic produce electrical energy. Then the energy will be sent to loads, storage systems, and the utility grid in priority order. Batteries are used to store and buffer electrical energy. Excessive energy will

be transmitted to the utility grid at a lower price. The core issue of controlling the energy scheduling system is making an optimal control policy, which can coordinate all the subsystems. We assume the microgrid runs in a grid-connected mode and actively participates in the real-time electricity market to take advantage of the real-time electricity prices. The interval for each time step is 1 hour.

4.2. Summary of Current Literature

Table **1** summarizes the current literature about the ADP based economic dispatch problem. The Notations throughout this paper are defined as follows

k time index

$P_{G,k}$ power of the utility grid

$P_{L,k}$ load demand

$P_{R,k}$ the output power of the renewable energy

$E_{B_Q,k}$ energy of the Qth battery

$E_{B_Q,\min}$ minimum energy of the Qth battery

$E_{B_Q,\max}$ maximum energy of the Qth battery

$P_{B_Q,k}$ power of the Qth battery

$P_{B_Q,\min}$ minimum power of the Qth battery

$P_{B_Q,\max}$ maximum power of the Qth battery

$P_{B,k}$ the total battery power

λ the periodic of $P_{L,k}$, $P_{G,k}$and C_k

$\bar{F}$ uncontrollable system function

N_T scheduling horizon

According to the power balance relationship, we have

$$P_{G,k} = P_{L,k} - P_{B,k} - P_{R,k} \tag{29}$$

the total battery power is the sum of each single battery

$$P_{B,k+1} = \sum_{Q=1}^{n_b} P_{B_Q,k} \tag{30}$$

The dynamic of each the Qth single battery is modelled by

$$E_{B_Q,k+1} = E_{B_Q,k} - \eta_Q(P_{B_Q,k}, T)P_{B,k} \tag{31}$$

and

$$EB_Q, k_{B_Q,max}{}_{B_Q,min} \tag{32}$$

The battery power is limited by its rated discharging/charging power

$$PB_Q, k_{B_Q,max}{}_{B_Q,min} \tag{33}$$

The utility function in different papers is listed in Tbl. 1. They are all time-varying due to the electricity price C_k varies with time k.

Equations (29)-(33) represents the deterministic part of the mathematical model in the economic dispatch problem. The literature in Tbl. 1 is all based on them.

However, the variables are not the same in all the literature. The deterministic part, $E_{B_Q,k}$ are defined as the state variables, and $P_{B_Q,k}$ are the control variables. For the convenience of analysis, we divide the current literature into three types.

Type 1 literature mainly represents rather early research work. In [28], the electricity price C_k, the renewable energy $P_{R,k}$ and the load profile $P_{L,k}$ are the other three state variables. Then, the mathematical model in the form as

$$x_{k+1} = \bar{F}(x_k, u_k) \tag{34}$$

and the utility function $U(x_k, u_k)$ is time-invariant. It should be emphasized that the property of $\bar{F}$ is different from F. Since $P_{R,k}$, $P_{L,k}$ and C_k are not influenced by the battery discharging/charging power, they are not controllable. Therefore, the iterative adaptive dynamic programming for the controllable system is invalid. Two neural networks are used for the implementation of the ADP algorithm proposed in [28]. The modification of parameters in the action network and critic network is based on the trial-and-error based method. The system starts to run from an initial state x_0 to the terminal time of the scheduling horizon N_T. At each time step, the action network and critic network are updated. An episode is defined as the states run from x_0 to x_{N_T}. The optimal parameters are solved by carrying out enough episodes. In other literature of type 1 marked in Tbl. 1, such as the authors extend the scene to the multi-houses case. The training process of neural networks depends on historical data. Hence, the generalization ability of the controller may be reduced when the uncontrollable part is very uncertain.

The type 2 literature is firstly proposed by the authors in [42]. The iterative adaptive dynamic programming scheme is used for the economic dispatch for residential buildings. Different from the type 1 literature, $P_{R,k}$ and $P_{L,k}$ are defined as the external parameters of the system. Hence, the system function is time-varying instead of the time-invariant one. The dynamic electricity price is the time-varying parameter in the utility function. Then, the mathematical model in the form as

$$x_{k+1} = F(x_k, u_k, P_{R,k}, P_{L,k}) \tag{35}$$

and

$$U(x_k, u_k, k) = x_k^T Q_k x_k + u_k^T R_{u_k} \tag{36}$$

For the general time-varying system $x_{k+1} = F(x_k, u_k, k)$, the optimal controller is also time-varying, i.e. $u_k^* = \mu_k^*(x_k)$. It is nearly impossible to solve the optimal controller for the general time-varying system. Hence, in the type 2 literature, except for [45], the authors make the periodic assumption for the external parameters, i.e. $P_{R,k+\lambda} = P_{R,k}$,$P_{L,k+\lambda} = P_{L,k}$ and$C_{k+\lambda} = C_k$. Hence, it is proven that the iterative control law converges to an optimal periodic control sequence, i.e. $\{u_0^*(x_k), u_1^*(x_k), \cdots, u_{\lambda-1}^*(x_k)\}$. To implement the iterative algorithm for the time-varying system under the periodic assumption, two iterative loops are introduced in type 2 literature. The authors also extend their algorithms to a more complex case. In [43, a multi-battery system is considered. To reduce the problem caused by the high dimension state vector, a virtual battery @ is defined to present the worse case of all the batteries, which simplifies the computation. To limit the battery power, the term related to the control variable is modified as $\int_0^{P_{B_{Q,k}}\eta_Q{}^{(T,P_{B_{Q,k}})}} (\Psi^{-1}(s))^T \, ds$, where $\Psi(\cdot)$ is a monotonic odd function with its first derivative bounded by a constant M. The discount factor is defined as a parametric variable to overcome the disturbance of the periodic load profile. The authors in type 2 paper establish the mathematical model of the economic dispatch, which is easy to give the convergence proof. Therefore, the methods proposed in type 2 papers are more reliable. However, the theoretical analysis will become more complex with the model becoming complicated.

Table 1: Summary of current literature.

Reference	Year	Load Profile	State Variables	Time-Varying Parameters	Controllable
[26]	2013	nonperiodic	$E_{B,k}, P_{L,k}, C_k$	N	N
[27]	2013	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[28]	2013	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[29]	2015	nonperiodic	$E_{B,k}, P_{L,k}, C_k$	N	N
[42]	2015	periodic	$E_{B,k}, P_{G,k}$	Y	Y
[43]	2015	periodic	$E_{B_Q,k}, P_{G,k}$	Y	Y
[34]	2016	periodic	$E_{B_Q,k}, P_{G,k}$	Y	Y
[44]	2016	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[35]	2017	periodic	$E_{B,k}, P_{G,k}$	Y	Y
[36]	2017	periodic	$E_{B,k}, P_{G,k}$	Y	Y
[45]	2017	quasi-periodic	$E_{B,k}, P_{G,k}$	Y	Y
[46]	2017	periodic	$E_{B,k}, P_{G,k}$	Y	Y
[30]	2018	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[31]	2018	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[32]	2018	nonperiodic	$E_{B,k}, P_{L,k}, C_k, P_{R,k}$	N	N
[37]	2019	periodic	$E_{B,k}, P_{G,k}$	Y	Y

The type 3 literature is similar to the reinforcement learning technique. The authors in [31 and 32] define the microgrid system as a Markov decision process. And they define the optimal control law in the sense of expectation. The type 3 algorithms reduce the dependency of optimality on the forecast information.

It is worthwhile pointing out that the parametric structure can greatly affect the result of algorithms. In most of the literature, a feedforward network is adopted to approximate the iterative value function and iterative control law. However, other neural networks may work better than traditional feedforward networks, such as recurrent neural networks, deep residual networks.

4.3. Dynamic of Uncertain Part

It is worthwhile mentioning that the dynamic of the load is not considered in most of the current literature. Actually, the load profile used in the iterative algorithm is the load consumption in the future, which is required to be predicted. Hence, the prediction error is unavoidable, which will have an effect on the performance of the ADP algorithm.

There are different mathematical models to describe the behavior of users. In [37], an exponential smoothing model [47, 48] is used to predict the load profile and electricity price in the future. The next-day load consumption and price are predicted by

$$\hat{P}_{L,d}(\tau) = \omega \hat{P}_{L,d-1}(\tau) + (1-\omega) P_{L,d-1}(\tau) \tag{37}$$

and

$$\hat{C}_{r,d}(\tau) = \omega \hat{C}_{r,d-1}(\tau) + (1-\omega) C_{r,d-1}(\tau) \tag{38}$$

where $\hat{P}_{L,d-1}(\tau)$ and $\hat{C}_{r,d-1}(\tau)$ are predictions for the previous day, and $P_{L,d-1}(\tau)$ and $c_{r,d-1}(\tau)$ are actual value in the previous day, and $\tau \in \{0,1,2,\cdots,\lambda-1\}$ is hour index. ω is the smooth parameter.

The feedforward network-based forecast scheme is the most widely used in the literature. Given a time series of active load, we can model the load profile by a feedforward network

$$\hat{P}_{L,k+1} = \hat{g}(P_{L,k}, P_{L,k-1}, \cdots, P_{L,k-N_P}) \tag{39}$$

where $\hat{g}$ denotes the neural network and $N_P + 1$ is the number of historical data.

The exponential smoothing model and feedforward network-based forecasted model is rather simple. It should be noted that for a quasi-periodic load profile, the prediction error is small because the change of load consumption is slow. However, it may be invalid when the load profile is highly stochastic.

The methods proposed in type 1 and type 2 literature greatly rely on the result of load forecasting. Hence, a more effective load forecast algorithm for the external parameters with high volatility is necessary to be developed.

5. Comments and Topics for Future Studies

ADP is a powerful tool to solve the difficult optimal control problem of complex systems. Current literature reveals the potential of ADP in dealing with the economic dispatch problem. However, existing literature about ADP mainly discusses optimal control of nonlinear time-invariant systems. The economic dispatch problem, it brings new challenges to the traditional ADP algorithm. Firstly, the iterative ADP algorithm for the partly controllable system should be developed to support the analysis of the ADP based economic dispatch design. Secondly, the periodic assumption may restrict the application in an engineering problem. The residential load profile is quite stochastic and has very high volatility.

Given that, in order to apply the iterative ADP algorithm to the economic dispatch problem with highly volatile loads, there are three directions that can be the focus in the future.

Firstly, the mathematical model of economic dispatch can be specified. The existing models only consider the power balance. The voltage, current, and reactive power can be also considered in the mathematical model. Meanwhile, the battery model is also simplified in the existing literature. The output of renewable energy is assumed to be the maximum. All these may be oversimplified. In the future, it is very meaningful to establish hardware in loop experiments to test the effectiveness of the iterative ADP when applying to the real world.

Secondly, it is the parametric structure that plays a very important role in the ADP algorithm. Until now, there is seldom literature discussing the proper parametric structure for the economic dispatch system. The suitability is more significant than the complexity of the function approximation structure. If a neural network is used as the parametric structure in the LQR problem, it may not have perfect performance as using the quadratic form structure. Since the existence of approximation error, the complex structure may lead to failure in weights convergence.

Finally, the error between the infinite horizon and finite horizon problem is worth discussing. Only the authors in [49] discuss this topic. However, the system in [49] is time-invariant. Until recently, there is some lack of knowledge about the ADP algorithm for finite-horizon economic dispatch. Usually, we just need to focus on the scheduling horizons, such as one day, one week, and one month. Furthermore, the prediction error may become vary largely in the far future. A more effective predictive algorithm is also needed to be developed.

6. Conclusion

In this paper, we review the state-of-the-art online ADP based economic dispatch algorithms. The relationship between the optimal control framework and the ADP algorithm is revealed. By reviewing the ADP algorithms, the

advantage of the ADP based economic dispatch method is illustrated. A comparison of the existing literature is given, and we point out three directions for the ADP based economic dispatch method.

In future work, we will consider more practical factors on the basis of the three directions of the ADP-based economic dispatch method proposed in this paper, build a more realistic smart grid economic dispatch management system model, collect more relevant data, and conduct more In-depth research makes this research more practical and valuable.

References

[1] Werbos PJ. Computational intelligence for the smart grid-history, challenges, and opportunities. IEEE Computational Intelligence Magazine, 2011; 6(3): pp. 14-21. https://doi.org/10.1109/MCI.2011.941587

[2] Venayagamoorthy GK. Dynamic, stochastic, computational, and scalable technologies for smart grids. IEEE Computational Intelligence Magazine, 2011; 6(3): pp. 22-35. https://doi.org/10.1109/MCI.2011.941588

[3] Nguyen DT, Le LB. Optimal bidding strategy for microgrids considering renewable energy and building thermal dynamics. IEEE Transactions on Smart Grid, 2014; 5(4): pp. 1608-1620. https://doi.org/10.1109/TSG.2014.2313612

[4] Talari S, Yazdaninejad M, Haghifam M. Stochastic-based scheduling of the microgrid operation including wind turbines, photovoltaic cells, energy storages and responsive loads. IET Generation, Transmission Distribution, 2015; 9(12): pp. 1498-1509. https://doi.org/10.1049/iet-gtd.2014.0040

[5] Changjin X, *et al*. Further exploration on bifurcation of fractional-order six-neuron bi-directional associative memory neural networks with multi-delays. Applied Mathematics and Computation, 2021; 410: 126458. https://doi.org/10.1016/j.amc.2021.126458

[6] Chen Z, Wang J, Ma K, *et al*. Fuzzy adaptive two-bits-triggered control for nonlinear uncertain system with input saturation and output constraint. International Journal of Adaptive Control and Signal Processing, 2020; 34(4): 543-559. https://doi.org/10.1002/acs.3098

[7] Changjin X, *et al*. Bifurcation Dynamics in a Fractional-Order Oregonator Model Including Time Delay. Match-Communications in Mathematical and in Computer Chemistry, 2022; 87(2): 397-414. https://doi.org/10.46793/match.87-2.397X

[8] Changjin X, *et al*. Bifurcation control strategy for a fractional-order delayed financial crises contagions model. AIMS Mathematics, 2022; 7(2): 2102-2122. https://doi.org/10.3934/math.2022120

[9] Lewis FL. Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE circuits and systems magazine, 2009; 9(3): pp. 32-50. https://doi.org/10.1109/MCAS.2009.933854

[10] Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. IEEE Control Systems Magazine, 2012; 32(6): pp. 76-105. https://doi.org/10.1109/MCS.2012.2214134

[11] Nguyen TA, Crow ML. Stochastic optimization of renewablebased microgrid operation incorporating battery operating cost. IEEE Transactions on Power Systems, 2016; 31(3): pp. 2289-2296. https://doi.org/10.1109/TPWRS.2015.2455491

[12] Werbos P. Advanced forecasting methods for global crisis warning and models of intelligence. General System Yearbook, 1977; pp. 25-38.

[13] Wei Q, Song R, Li B, Lin X. Self-Learning Optimal Control of Nonlinear Systems. Springer, 2018; 103: https://doi.org/10.1007/978-981-10-4080-1

[14] Si J, Wang Y-T. Online learning control by association and reinforcement. IEEE Transactions on Neural networks, 2001; 12(2): pp. 264-276. https://doi.org/10.1109/72.914523

[15] Al-Tamimi A, Lewis FL, Abu-Khalaf M. Discrete-time nonlinear hjb solution using approximate dynamic programming: Convergence proof. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008; 38(4): pp. 943-949. https://doi.org/10.1109/TSMCB.2008.926614

[16] Wang D, Liu D, Wei Q, Zhao D, Jin N. Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. Automatica, 2012; 48(8): pp. 1825-1832. https://doi.org/10.1016/j.automatica.2012.05.049

[17] Wei Q, Liu D, Lin H. Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. IEEE Transactions on cybernetics, 2015; 46(3): pp. 840-853. https://doi.org/10.1109/TCYB.2015.2492242

[18] Wei Q, Lewis FL, Liu D, Song R, Lin H. Discrete-time local value iteration adaptive dynamic programming: Convergence analysis. 2018; 48(6): pp. 875-891. https://doi.org/10.1109/TSMC.2016.2623766

[19] Liu D, Wei Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Transactions on Neural Networks and Learning Systems, 2013; 25(3): pp. 621-634. https://doi.org/10.1109/TNNLS.2013.2281663

[20] Wang J, Zhang H, Ma K, Liu Z, Chen CLP. Neural Adaptive Self-Triggered Control for Uncertain Nonlinear Systems With Input Hysteresis. IEEE transactions on neural networks and learning systems, 2021; https://doi.org/10.1109/TNNLS.2021.3072784

[21] Wang J, Gong Q, Huang K, *et al*. Event-triggered Prescribed Settling Time Consensus Compensation Control for a Class of Uncertain Nonlinear Systems with Actuator Failures, IEEE transactions on neural networks and learning systems, 2021; https://doi.org/10.1109/TNNLS.2021.3129816

[22] Xue S, Luo B, Liu D. Event-triggered adaptive dynamic programming for zero-sum game of partially unknown continuous-time nonlinear systems. IEEE Transactions on Systems, Man, and Cybernetics, 2019; pp. 1-11.

[23] Vamvoudakis KG, Lewis FL, Hudas G. Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality. Automatica, 2012; 48(8): pp. 1598-1611. https://doi.org/10.1016/j.automatica.2012.05.074

[24] Abouheaf M, Lewis FL, Vamvoudakis KG, Haesaert S, Babuska R. Multi-agent discrete-time graphical games and reinforcement learning solutions. Automatica, 2014; 50(12): pp. 3038-3053. https://doi.org/10.1016/j.automatica.2014.10.047

[25] Wei Q, Liu D. Data-driven neuro-optimal temperature control of water-gas shift reaction using stable iterative adaptive dynamic programming. IEEE Transactions on Industrial Electronics, 2014; 61(11): pp. 6399-6408. https://doi.org/10.1109/TIE.2014.2301770

[26] Huang T, Liu D. A self-learning scheme for residential energy system control and management. Neural Computing and Applications, 2013; 22(2): pp. 259-269. https://doi.org/10.1007/s00521-011-0711-6

[27] Boaro M, Fuselli D, De Angelis F, Liu D, Wei Q, Piazza F. Adaptive dynamic programming algorithm for renewable energy scheduling and battery management. Cognitive Computation, 2013; 5(2): pp. 264-277. https://doi.org/10.1007/s12559-012-9191-y

[28] Fuselli D, De Angelis F, Boaro M, Squartini S, Wei Q, Liu D, Piazza F. Action dependent heuristic dynamic programming for home energy resource scheduling. International Journal of Electrical Power & Energy Systems, 2013; 48: pp. 148-160. https://doi.org/10.1016/j.ijepes.2012.11.023

[29] Xu Y, Liu D, Wei Q. Action dependent heuristic dynamic programming based residential energy scheduling with home energy interexchange. Energy Conversion and Management, 2015; 103: pp. 553-561. https://doi.org/10.1016/j.enconman.2015.06.048

[30] Liu D, Xu Y, Wei Q, Liu X. Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming. IEEE/CAA Journal of Automatica Sinica, 2017; 5(1): pp. 36-46. https://doi.org/10.1109/JAS.2017.7510739

[31] 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 Transactions on Sustainable Energy, 2018; 10(2): pp. 931-942. https://doi.org/10.1109/TSTE.2018.2855039

[32] 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 Transactions on Smart Grid, 2018; 10(3): pp. 2440-2452. https://doi.org/10.1109/TSG.2018.2798039

[33] Wei Q, Liao Z, Song R, Zhang P, Wang Z, Xiao J. Self-Learning Optimal Control for Ice-Storage Air Conditioning Systems via Data-Based Adaptive Dynamic Programming. in IEEE Transactions on Industrial Electronics, 2021; 68(4): pp. 3599-3608. https://doi.org/10.1109/TIE.2020.2978699

[34] Wei Q, Liu D, Liu Y, Song R. Optimal constrained self-learning battery sequential management in microgrid via adaptive dynamic programming. IEEE/CAA Journal of Automatica Sinica, 2016; 4(2): pp. 168-176. https://doi.org/10.1109/JAS.2016.7510262

[35] Wei Q, Shi G, Song R, Liu Y. Adaptive dynamic programmingbased optimal control scheme for energy storage systems with solar renewable energy. IEEE Transactions on Industrial Electronics, 2017; 64(7): pp. 5468-5478. https://doi.org/10.1109/TIE.2017.2674581

[36] Shi G, Liu D, Wei Q. Echo state network-based q-learning method for optimal battery control of offices combined with renewable energy. IET Control Theory & Applications, 2017; 11(7): pp. 915-922. https://doi.org/10.1049/iet-cta.2016.0653

[37] Zhu Y, Zhao D, Li X, Wang D. Control-limited adaptive dynamic programming for multi-battery energy storage systems. IEEE Transactions on Smart Grid, 2019; 10(4): pp. 4235-4244. https://doi.org/10.1109/TSG.2018.2854300

[38] Lewis FL, Vrabie D, Syrmos VL. Optimal control. John Wiley & Sons, 2012. https://doi.org/10.1002/9781118122631

[39] Lewis FL, Vamvoudakis KG. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2010; 41(1): pp. 14-25. https://doi.org/10.1109/TSMCB.2010.2043839

[40] Vrabie D, Lewis F. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. Neural Networks, 2009; 22(3): pp. 237-246. https://doi.org/10.1016/j.neunet.2009.03.008

[41] Liu D, Wei Q. Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. IEEE Transactions on Cybernetics, 2013; 43(2): pp. 779-789. https://doi.org/10.1109/TSMCB.2012.2216523

[42] Wei Q, Liu D, Shi G. A novel dual iterative q-learning method for optimal battery management in smart residential environments. IEEE Transactions on Industrial Electronics, 2014; 62(4): pp. 2509-2518. https://doi.org/10.1109/TIE.2014.2361485

[43] Wei Q, Liu D, Shi G, Liu Y. Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming. IEEE Transactions on Industrial Electronics, 2015; 62(7): pp. 4203-4214. https://doi.org/10.1109/TIE.2014.2388198

[44] Venayagamoorthy GK, Sharma RK, Gautam PK, Ahmadi A. Dynamic energy management system for a smart microgrid. IEEE transactions on neural networks and learning systems, 2016; 27(8): pp. 1643-1656. https://doi.org/10.1109/TNNLS.2016.2514358

[45] Wei Q, Lewis FL, Shi G, Song R. Error-tolerant iterative adaptive dynamic programming for optimal renewable home energy scheduling and battery management. IEEE Transactions on Industrial Electronics, 2017; 64(12): pp. 9527-9537. https://doi.org/10.1109/TIE.2017.2711499

[46] Wei Q, Liu D, Lewis FL, Liu Y, Zhang J. Mixed iterative adaptive dynamic programming for optimal battery energy control in smart residential microgrids. IEEE Transactions on Industrial Electronics, 2017; 64(5): pp. 4110-4120. https://doi.org/10.1109/TIE.2017.2650872

[47] Hong SH, Yu M, Huang X. A real-time demand response algorithm for heterogeneous devices in buildings and homes. Energy, 2015; 80: pp. 123-132. https://doi.org/10.1016/j.energy.2014.11.053

[48] Yang X, Zhang Y, He H, Ren S, Weng G. Real-time demand side management for a microgrid considering uncertainties. IEEE Transactions on Smart Grid, 2019; 10(3): pp. 3401-3414. https://doi.org/10.1109/TSG.2018.2825388

[49] Wang F-Y, Jin N, Liu D, Wei Q. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with -error bound. IEEE Transactions on Neural Networks, 2010; 22(1): pp. 24-36. https://doi.org/10.1109/TNN.2010.2076370