A Multi-agent reinforcement learning algorithm with fuzzy approximation for Distributed Stochastic Unit Commitment

Ghorbani, Farzaneh
University of Zanjan, Zanjan, Iran
f.ghorbani@znu.ac.ir

Afsharchi, Mohsen
University of Zanjan, Zanjan, Iran
afsharchi@znu.ac.ir

Derhami, Vali
Yazd University, Yazd, Iran
vderhami@yazd.ac.ir

Abstract

This paper proposes a novel multi-agent unit commitment model under Smart Grid (SG) environment to minimize the demand satisfaction error and production cost. This is a distributed solution applicable in non-deterministic environments with stochastic parameters intending to solve Distributed Stochastic Unit Commitment (DSUC) problem. We use multi-agent reinforcement learning (RL) in which agents learn as independent learners to cooperatively satisfy the demand profile. The learning mechanism proceeds using a reward signal, which is given based on the performance of the entire system as well as the impact of the joint action of the agents. The learning agent utilizes a novel multi-agent version of Fuzzy Least Square Policy Iteration (FLSPI) as a model-free RL algorithm to approximate Q-function. Based on this approximation, the agent makes the best decision to achieve the goals while considering the constraints governing the system. Uncertainty sources in our definition of the problem are fluctuations in the predicted demand function, random productions of clean energy generators and the possibility of accidental failure in power generators. Training for one time interval (i.e. one season or one year) consisting of several time intervals (i.e. days) can be simultaneously conducted by one trial in our method. We have conducted our experiment in two different frameworks. These frameworks are defined based on the problem complexity in terms of the number of generators, the uncertainties in the environment and the system constraints. The results show that the learning agent learns to satisfy the demand profile as well as other constrains.

1 Introduction

In the present era, the supply of electricity in its traditional form is carried out based on the Primary Energy Sources (PES) to meet the industrial demands. PES are operationally expensive and generation of electricity by these resources causes air pollution and environmental consequences. Additionally, centralized production and long-distance transmission lead to low reliability. Efforts to resolve these challenges have led to birth of a new power grid called Smart Grid. In addition, distributed generation of power, as one of the most important smart grid goals, employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies [9]. This smartness offers various benefits such as higher reliability, less unpredictable outages, less human error, less energy losses, higher transmission and distribution capacity and promotion of the use of low-cost and renewable energy resources such as wind turbines and solar panels, while upgrading the power generation and distribution infrastructures [4].

One of the main substructures of power grid is microgrid, which is a small-scale power supply network consisting of low-capacity renewable energy generators, residential electrical consumers (e.g., home appliances), and energy storage devices [2, 11]. Microgrids are aware of the local energy supply, the demand profile and can trade energy with other microgrids and connecting power plants [4]. In the smart grid, microgrids can sell extra energy to other microgrids to reduce the dependence on the power plant and save the long-distant energy transmission loss [27].

With time-varying renewable energy production, production of a set of electrical generators needs to be coordinated to achieve some common target: to match the energy demand at minimum cost or maximize revenues from energy production. This coordinated optimization process is called Unit Commitment (UC) [20]. In an uncertain real world environment, this problem becomes Stochastic Unit Commitment, which has been studied by various researchers [8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 24, 25, 26].

In [11], authors take advantage of linear programming to select optimal system capacities and schedule of operation for the stochastic unit commitment (SUC) problems in a microgrid. The method is tested in a microgrid with three consumers and three CHP (Combined Heat and Power), wind and photo-voltaic energy generators, storages of thermal and electricity, management systems for communications and energy as well as other components. The proposed method in [24] uses dynamic programming to solve UC problems when demand is not certain. Any renewable energy is considered in this paper. Descent algorithm is used for Stochastic Storage Problems in [17]. No renewable energy exists in the defined problem, and continuous parameters are discretized to use by the method. Authors in [15] deploy a multi-agent method to solve UC problems with several types of agent consisting of a facilitator, generators and mobile agents. Generator agents and mobile agents negotiate with each other. Experiments are down in a simple test-bed consisting of three

controllable generators, a single facilitator and two types of mobile agents. Deterministic unit commitment problems (DUC) are solved in [18] by solving subproblems using dynamic programming. To examine the method, several controllable generators are used. In [14], a multi-agent reinforcement learning method on the base of Q-learning is used to introduce a method to unfold dynamic economic emissions dispatch problem. Stochastic Games framework is considered to show the problem as a sequential decision-making process. Soltani et.al. in [21] consider the multi-objective problem of unit commitment in presence of uncontrollable energy sources (i.e. wind and solar power). In this work, generators failures are not included and the solution focuses on the cost and emissions minimizing. Demand uncertainty caused by demand-side response is discussed in [25]. This is a method with focus on the price-elasticity of power demand.

Logenthiran et.al. in [12] describe a three-step method to find a solution of thermal unit commitment problems in a microgrid in island mode. It uses Lagrangian relaxation and genetic algorithm and is tested in a system consisting of PV (Photo Voltaic) and wind energy, several thermal units and battery banks. Authors proposed a multi-agent method for the microgrid in island mode in reference [13]. Agents in several types such as load and microgrid levels, storage and microgrid managements, coordinator, database and power world simulator are used in the mentioned paper. The work in [16] focuses on the SUC problem with variable demand and generators' outputs. The problem is investigated as a factored Markov decision process model, and an approximate algorithm is proposed. This method tries to balance cost of operation and risk of blackout. Wang et.al. in [26] investigate the UC problem with the volatile renewable energy of wind considering security of the system. The problem of wind energy is also under investigated in [19] considering the failures of the network, generators and transmission lines. Distributed gradient descent is used to propose a method to solve SUC problem in a distributed manner in [8]. Renewable energy is considered in this paper.

Almost in all of the above-mentioned researches, the problem is modeled centrally or the decision-making agents (i.e. generators) share information to solve the problem [8]. However, for various reasons such as cybernetic attacks and market competition, information sharing is not feasible in real-world environments [8]. In addition, the stochastic nature of the problem such as the demand function fluctuations and the random amount of generated clean energy has been ignored at the most researches. Generally, what we consider in this work has not been reported so far.
In this paper, we train controllable generators to learn to meet the demand profile of the micro-grid in a cooperative manner while satisfying the existing constraints. We propose a multi-agent reinforcement learning method for problems with continuous state-action spaces to solve the Stochastic Unit Commitment in a fully distributed way. Thus, the problem we tackle is Distributed Stochastic Unit Commitment problem (DSUC). Our contribution

can be summarized as follows:

- The agents solve the problem without sharing much information and help provide more security in the power grid: We assume that agents do not share information of their policies and decisions due to the potential cyber-attacks aiming at spiteful control of electricity flow.
- The agents learn to satisfy the demand function despite its unpredicted fluctuations: Unpredictable fluctuations of the demand function is something that may occur in power grids making the environment uncertain so that the learning task is more challenging. By learning this random variation, the grid will not be interrupted by power failure and thus its reliability will not be reduced.
- The agents learn to comply with clean energy generators: The presence of uncontrollable clean energy generators influenced by the weather condition is another source of uncertainty that can be handled by our method.
- The agents learn to satisfy the demand function associated with a time interval: This is not a one-time solution of UC problem that will be repeated at time intervals. The agents can be trained to work for several time periods; for instance a month, a season, or even a year.
- The agents work in the continuous state and action spaces: The generator production is not necessarily selected from a discrete set. Therefore, this solution is a continuous time solution and is able to satisfy a continuous demand function.
- The total performance of the system is based on the received reward from the environment and the general state of the system. Therefore, unlike many existing multi-agent learning methods, increasing the number of agents will not increase the time and space complexity of the proposed method.
- The proposed algorithm has a theoretical foundation, high learning speed with a very low error rate in learning.

The remainder of this paper is organized as follows: Section 2 presents some preliminaries to express the proposed method. Section 3 contains the proposed multi-agent algorithm to solve DSUC problem. In Section 4, we use two frameworks to test and explain the results of the experiments.

2 PRELIMINARY CONCEPTS

This section provides a brief overview of the concepts that we need throughout this paper.

2.1 Reinforcement Learning

The main idea behind the reinforcement learning is that the rewarded behavior is likely to be repeated, whereas a behavior, which is punished is less likely to recur [23]. Thus, an agent learns from the received environmental feedback by two different signals: state signal indicates the agent state in the environment, and the reward signal shows feedback of the environment to determine the desirability of the agent state. The agent tries to maximize its long-term utility. By Reinforcement Learning (RL) methods, in the state s, an agent takes the action a, goes to the state s' and receives the reward r. The agent updates its state value function $V(s)$ or its state-action value function $Q(s,a)$, showing the long-term usefulness of the state s or usefulness of the action a in the state s, respectively. This could be seen in Relations (1) and (2), which is called Bellman equation [5] where S is the state space, A is the action set and γ $(0 \leq \gamma < 1)$ is the discount factor.

$$V_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty}(\gamma^k R_{t+k+1})|S_t = s] \tag{1}$$

where $\mathbb{E}_\pi[.]$ and R_t mean of expected value of $[.]$ and received reward in time t.

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \\ &\mathbb{E}_\pi[\sum_{k=0}^{\infty}(\gamma^k R_{t+k+1})|S_t = s, A_t = a] \end{aligned} \tag{2}$$

2.2 Multi-Agent Reinforcement Learning (MARL)

A multi-agent system is a loosely coupled network of problem-solving entities (agents) working together to find answers to problems beyond the individual capabilities or knowledge of each entity (agent) [22]. Like other intelligent entities, agents act based on the utility in any state of environment. In the presence of other agents, uncertainty and a general utility model, a problem can be modeled as an Multi-agent Markov Decision Process (MMDP) in which a joint action at any state consists of individual action performed by all the agents [6]. Let the system be fully observable to each agent, then an MMDP is defined as a tuple $M = \langle \beta, A, S, P, R \rangle$ where β is a set of m agents, every agent $i \in \beta$ has a finite set of actions A_i and the joint action space $A = A_1 \times \cdots \times A_m$ is made of the elements $\langle a_1, \cdots, a_m, \rangle$, $a_i \in A_i$. In addition, S is the state space, $P : S \times A \times S \to [0,1]$ is the dynamic of the system and $R : S \to \Re$ is a bounded reward function with a real value.

2.3 Unit Commitment problem

Unit commitment (UC) is an optimization problem used to determine the operation schedule of the generating units at every hour interval with varying loads under different constraints and environments [20]. The optimization

problem tries to find the best solution to satisfy demand load while considering the grid constraints. Among these constraints are the limited capacity of the power generators, limited battery capacity and minimizing production cost. The Stochastic Unit Commitment is a special case of the unit commitment problem that due to the nature of the random and unpredictable production of clean and renewable energy generators and also random fluctuations in the demand profile, uncertainty is introduced to the original problem [8]. The basis of our model is cooperative multi-agent systems. It should be emphasized that the grids in our formulation include a combination of renewable energy resources and PES generators (i.e. agents). Clearly, only the production of PES generators can be controlled and the amount of the produced clean energy appears as an uncertainty source, making the learning process more challenging. In the following, we model the UC and SUC problem according to our modeling.

Definition 2.1 *Unit Commitment (UC): A Unit commitment problem is defined by a tuple* (C, N, S, A, L, F) *where* C *is the number of controllable generators in the micro-grid,* N *is the number of time steps in a time period (for example, one day),* S *is the joint state space of controllable generators. In the exact word,* $S = (S_1, S_2, ..., S_C)$ *which* S_i *is the state space of* i*th agent (i.e. controllable generator) and* $S_i(t) \in \{0, 1\}$ *is its status in time step* t*,* A *is the joint action space of the controllable generators where* $A = (A_1, A_2, ..., A_C)$ *where* A_i *is the action space of the* i*th agent (i.e. each action* $a_i(t)$ *could be a change in energy production in time step* t*).* L *is the set of demanded load values according to the time steps over a period of time, therefore,* $L = (l_1, l_2, ..., l_N)$ *where* l_t *is the demand load in time step* t*.* F *is the set of constraints of the agents and the whole system. Thus,* $F = \cup_{i \in C} F_i \cup F_s$ *where* F_i *and* F_s *are constraints for the agent* i *and the whole system (i.e. global constraints).*

The goals in an UC problem are: [8]:

- Finding the status of any generator which is either on or off, in any time step.
- Determining the amount of any generator production in every time step.
- Ensuring that generators, with the determined status and production, satisfy the demanded load considering minimum cost and the other constraints.

Definition 2.2 *Stochastic Unit Commitment (SUC): A Stochastic Unit Commitment problem is defined by a tuple* $(U, C, N, S, A, L, F, E, O)$ *where* $C, N, S, A,$
L, F *are as definition 2.1.* U *is the number of renewable energy generators and* E *is the joint production space of clean energy generators (i.e. uncontrollable generators).* $E = (E_1, ..., E_U)$ *where* E_j *is production space of the* j*th uncontrollable generator and* $e_j(t)$ *is its production in time step* t*.*

Due to the presence of fluctuations in the demand function as well as generators' random production, SUC will be a complicated problem. We consider each decision-making point in SUC as an intelligent autonomous agent enabling them to cooperate and solve the problem. We model the problem as a distributed constraint optimization problem where agents optimize the constraints while satisfying the demand function. The optimization is carried out via constraint learning.

2.4 Fuzzy Least Square Policy Iteration

Most of the existing reinforcement learning approaches are proposed for problems with discrete state and action spaces, while most of the real world problems have large or continuous state and action spaces. Fuzzy Least Square Policy Iteration (FLSPI) [10] is among the few methods proposed for the problems with large or continuous state and action spaces. This method has acceptable learning speed and accuracy in single-agent environments and has a theoretical basis. By defining the basis functions using a zero-order Takagi-Sugeno fuzzy system, FLSPI makes Least Square Policy Iteration (LSPI) applicable for large and continuous spaces. This is a policy iteration (PI) based algorithm having two phases: policy evaluation and policy improvement. FLSPI uses the fuzzy system as an approximator to partition the state space and define the appropriate membership functions. The fuzzy rules will be defined based on this partitioning afterward. Consequences of the rules are made of the combination of the weighted candidate actions. Candidate actions are selected from the agent's action space and is used to generate the final continuous action. In each step of the algorithm, depending on the weight of the actions and the action selection method, an action is selected from each rule. The final action will be obtained from the weighted summation of these selected actions.

To have the formal definition of FLSPI, we assume that the state space is an m dimensional space in which its ith dimension is partitioned to d_i parts and l candidate actions are selected from the agent action space. Now, based on the problem definition, u rules are defined, which the ith rule is as follows:

$$\begin{gathered}\text{If } x_1 \text{ is } L_{i_1} \text{ and ... and } \ x_m \text{ is } L_{i_m} \text{ Then} \\ (o_{k_1} \text{ with weight } w_{i_1} \text{ or ... or } o_{k_l} \text{ with weight } w_{i_l})\end{gathered} \tag{3}$$

where L_{i_j} is the ith defined membership function for jth dimension of the state space. Using the defined fuzzy rules, the basis functions are defined as:

$$\phi(s,a) = \left[\overbrace{0...\mu_1(s)...0}^{m}\overbrace{0...\mu_2(s)...0}^{m}\ldots\overbrace{0...\mu_u(s)\ldots 0}^{m}\right]^T \tag{4}$$

Cardinality of $\phi(s,a)$ is equal to $u \times l$. Corresponding to each rule, the firing strength related to the mentioned state is located at the location of the selected

candidate action. FLSPI uses the defined updating rules of LSPI.

$$A{=}A{+}\phi\left(s,a\right)\left(\phi\left(s,a\right){-}\gamma\phi\left(s^{'},\pi\left(s^{'}\right)\right)\right)^{T} \tag{5}$$

$$b = b + \phi(s,a)r \tag{6}$$

Matrices A and b are used to update the weight vector, w.

$$Aw{=}b \tag{7}$$

The weight vector is used to update the action-value function to approximate the optimal policy.

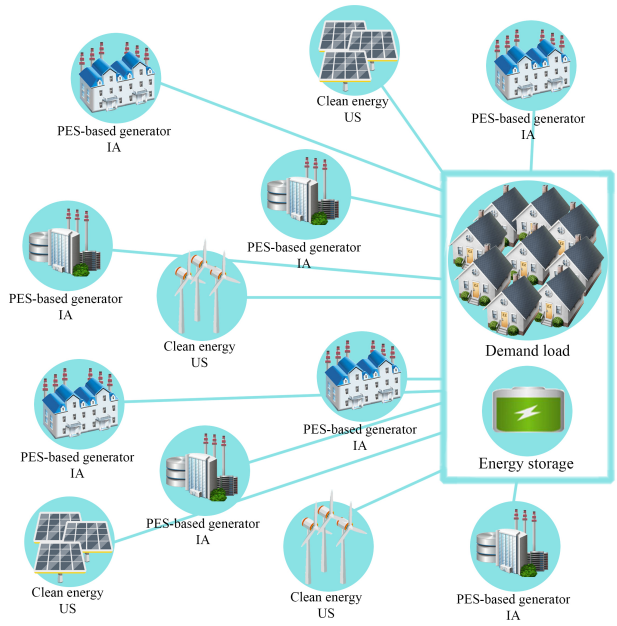
$$\widehat{Q}^{\pi}{=}\mathbf{\Phi}w = \sum_{i=1}^{R}\mu_i(s)w_{ii^{+}} \tag{8}$$

where R is number of fuzzy rules.

3 Proposed Method

Micro-grids play an important role in smart grids. A micro-grid is an electrical system including multiple loads and distributed energy resources that can be operated in parallel with the broader utility grid or a small, independent power system.

It increases reliability with distributed generation and efficiency with reduced transmission length, and it is easier to be integrated with alternative energy sources [1]. In addition, since micro-grid is a localized distributed network with sources and loads, it can be managed by distributed intelligent agents. In a multi-agent system, making the best decision depends on the other agents' decisions. Therefore, in most of the multi-agent learning methods, agents use the joint action learning strategy [7]. This needs information sharing that does not meet the reliability requirements of the smart grid where the exchange of information puts them at the risk of eavesdropping and cybernetic attacks and enable energy market speculators to abuse this information. In this paper, we propose a distributed solution for the SUC problem, based on a reinforcement learning method called Fuzzy Least Square Policy Iteration (FLSPI). The high-speed convergence, the existence of mathematical analysis, fewer adjustable parameters than other RL methods as well as acceptable performance in large or continuous spaces are among the advantages of this method. Our solution acts based on the received reward from the environment and is an independent action learner; therefore, the agents do not share information. To explain more, each agent approximates the system status based on the demand load, energy stored in the battery and the reward received at each time step implicitly. Based on this approximation, it selects the best possible action. We consider the state space as a three dimensional space: the amount of energy produced by the agent, demand load and the energy stored in the battery. Increasing or decreasing the



IA: Intelligent Agent, US: Uncontrollable source

Figure 1: Distributed generation in a micro-grid.

amount of the produced energy makes our action space. Assuming continuous state and action spaces provides more flexibility to determine the best value for the amount of energy that each agent must produce. By selecting an action and applying it, a reward signal will be given to the agent based on the behavior of the other agents, dynamic of the whole system and the system constraints (i.e. succeeding in demand satisfaction, minimizing production costs and so on). According to the joint action of the agents, the state of each agent will change to a new state including the energy that must be generated in the next time step, demand load and battery storage. Eventually, the agent learns the best behavior, allotting to it the highest accumulative reward. Like every RL modeling, learning is carried out based on the received reward. Therefore, how to define rewards in the UC problem is highly important. The details of our reward function are presented in 4.1. In the following, we explain our method more technically. At first, we partition the state space in all of its three dimensions and define the appropriate membership functions. As already mentioned, the first dimension is the range of the energy that the agent produces, i.e. $[0, E_{max}]$, where E_{max} is the maximum power production capacity of the learning agent. The second dimension is the range of demand fluctuations, i.e. $[0, D_{max}]$, where D_{max} is the maximum demand load for a specific time interval. The third dimension is the battery capacity, i.e. $[0, B_{max}]$, where B_{max} is the maximum capacity of the battery (maximum energy that could be

stored in the battery). Now, using a set of candidate actions, we can define fuzzy rules. This set is selected from all the possible actions of the agent. We define the continuous action space as $[-A_{max}^{dec}, A_{max}^{inc}]$, where A_{max}^{dec} and A_{max}^{inc} are the maximum increase and decrease in power production in each time step. Now, we define the fuzzy rules using the membership functions for the continuous state space and the weighted candidate actions of the continuous action space. The number of these rules is equal to $f_1 \times f_2 \times f_3$, where f_i is the number of the ith dimension's partitions. For example, if the state space is partitioned into three parts in each dimension, the number of fuzzy rules will be equal to 27. Therefore, by assuming that the number of candidate actions is equal to a, the rules are defined as follows:

$$\begin{gathered} \text{If } x_1 \text{ is } L_{i_1} \ \ \textit{and } x_2 \text{ is } L_{i_2} \text{ and } x_3 \text{ is } L_{i_3} \text{ Then} \\ (o_1 \text{ with weight } w_{j_1} \text{ or ... or } o_a \text{ with weight } w_{j_a}) \end{gathered} \tag{9}$$

where $1 \leq i_k \leq f_k$, $1 \leq k \leq 3$ and L_{i_k} is the membership function of i_k.
In other words, in every state, firing strength of the current state membership degree is placed in the position j^+ of the basis function vector. This is the position of the selected candidate action in the process of selecting the best action. Thus, the jth basis function of the state s is as follows:

$$[0 \ ... \ \mu_j(s) \ ... \ 0] \tag{10}$$

where $\mu_j(s)$ is the firing strength of jth rule for state s and is placed in position j^+.
Selecting the best candidate actions of each rule is done based on the candidate actions' weights and action selection method (e.g. using an exploration term), at every time step. This is because, selecting a suitable action for a state should be based on its state-action value. Based on the FLSPI method, the state-action values are dependent on the weight vector.

$$Q^{\pi}(s,a) = \phi(s,a)^T w \tag{11}$$

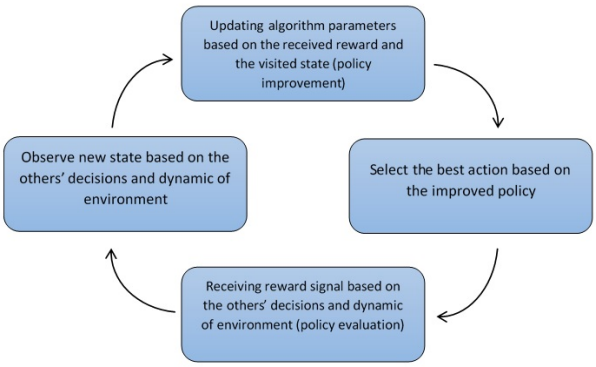


Figure 2: Block diagram of proposed method

Therefore, what is needed to obtain the best decision is to find the weight of the candidate actions based on the desirability of them depending on the reward received by the agents. This is the policy evaluation phase. In this phase, a reward signal is given to the agent, based on the system constraint compliance and demand satisfaction. We will explain this reward in Section 4.1. Obviously, the reward received from the environment and the agent's next state are not only dependent on the agent's action, but also depend on the actions performed by all agents in the environment, unpredicted changes in demand function and random amount of the produced clean energy. Total effect of these events changes battery storage to a new state and determine the next time step demand. Only the first part of the agent state (i.e. the amount of the agent production) is solely dependent on the agent's action. Figure 2 shows the associated diagram.

Here, we must calculate the weight of each candidate's action for each rule based on the reward received from the final action and the current state of the agent. Now, the final action should be computed based on the selected candidate actions with the coefficients of the firing strength related to the current state. To update the weight vector, matrices A and b are used. These matrices are updated using Relations (5) and (6), respectively. Then, the weight vector will be updated in Relation (7). This is the policy improvement phase.The process continues until the specified condition is met (for example achieving the goal or after a fixed number of iterations).

If we follow the normal process of the single-agent version of FLSPI, the next state of the agent is not predictable by just determining the current state and action. It depends on many other factors; therefore, parameter updating should be postponed until the environment changes to a new state.
It should be noted that due to the impact of the environment dynamics and other agents behavior in the agent's next state (i.e. to update matrix A) and the

Algorithm 1 Proposed method

Input: p_1, p_2 and p_3 : Number of the partitions for three dimension state space (generator power, demand and battery), $\{o_1, ..., o_1\}$: candidate action set (the value of decrease or increase), γ : Discount factor, initial matrices A_0, b_0 and w_0.

Output: π : policy (w : weight vector), the amount of energy changes in each time step

1: Observe initial state s_0

2: Select a suitable action o_{jj^+} from each rule based on the actions' weights and determined action selection strategy

3: Calculate amount of production change

$$a_1(s_1) = \sum_{i=1}^{u} \mu_i(s_1) o_{jj^+} \tag{12}$$

4: Apply a_1, observe s_2 and receive reward r_1 (based on the all agent actions and dynamic of system).

5: **repeat**

6: $t \leftarrow t + 1$

7:

$$A_t = A_{t-1} + \phi(s_{t-1}, a_{t-1})(\phi(s_{t-1}, a_{t-1}) - \gamma\phi(s_t, \pi(s_t)))^T \tag{13}$$

8:

$$b_t = b_{t-1} + \phi(s_{t-1}, a_{t-1}) r_{t-1} \tag{14}$$

9: Solve

$$\frac{1}{t} A_t W_t = \frac{1}{t} b_t \tag{15}$$

10: Select a suitable action o_{jj^+} from each rule based on the actions' weights and determined action selection strategy

11: calculate amount of production change

$$a_t(s_t) = \sum_{i=1}^{u} \mu_i(s_t) o_{jj^+} \tag{16}$$

12: Apply a_t, observe s_{t+1} (based on the all agent actions) and receive reward r_t.

13: **until** Adapt condition is met.

received reward (i.e. to update matrix b), the agent learns how to select its best action according to the system dynamic. Apparently, the agent approximates the policy of the others implicitly, and uses this approximation in selecting the best possible action in the common environment. The algorithm is presented in Algorithm 1.

3.1 Discussion

The only added complexity by the multi-agent version of FLSPI algorithm in compare to its single-agent version is the increase in the dimension of the state space from one to three. Let f_i be the partition number of the ith dimension in the state space. The number of the fuzzy rules is equal to $f_1 \times f_2 \times f_3$. If the number of the selected candidate action is equal to n_a, then in the single-agent version, we have $|A| = (f_1 \times n_a)^2$ and $|b| = f_1 \times n_a$. This is while in the proposed multi-agent version, we have $|A| = (f_1 \times f_2 \times f_3 \times n_a)^2$, $|b| = f_1 \times f_2 \times f_3 \times n_a$. It is important to note that the imposed complexity is independent of the number of the agents and does not increase by the number of agents and remains unchanged.

4 Experimental Setup

In this section, we first outline the frameworks for the test grids and then provide the settings and definitions of the parameters and finally present the results based on the Matlab simulations.

4.1 Test Environment

In this paper, we use two frameworks to test our solution for the DSUC problem. These frameworks are defined based on the problem complexity in terms of the number of generators, the uncertainties in the environment and the system constraints. Controllable generators are considered intelligent agents where they will be trained to make decision autonomously in a distributed manner. Renewable energy generators, power storages and demand functions are other components of these systems. In both frameworks, the state and action spaces are assumed continuous. Furthermore, the training time steps could be increased to increase the accuracy of the learned policy in exchange for time complexity. Therefore, it is necessary to balance the needed accuracy and time complexity in a determined period of time. This is possible by experiment.

Definition 4.1 *Framework 1: Here, we assume that there is a micro grid with two controllable and one uncontrollable generators. Random production of clean energy such as wind and solar energy confronts the environment with uncertainty. In addition, unpredicted demand function fluctuations also add more uncertainty to the environment. Generators have some limitations such as the maximum production capacity as well as the amount of increase and decrease*

in energy production at a specified time step. Such constraints are appropriately defined for any controllable generators. On the contrary, constraints such as battery capacity and demand satisfaction are subject to the general constraints defined for all agents. Therefore, the reward will be defined based on the violation from the production power range for the learning generator, the success of demand satisfaction, and compliance with the capacity of the battery.

Definition 4.2 *Framework 2: In this framework, the number of generators increases to ten, including seven controllable and three uncontrollable generators. The uncertainty in the demand function is also considered. Due to the failure possibility of generators in the real world problem, we also consider this issue in this framework. It is assumed that some generators may fail, with a random probability in each time step, and after a random number of time steps, they will be repaired. In this case, to have a reliable system, the remaining generators should compensate the lack of the failed generators, with the minimum error of demand satisfaction as well as with minimum imposed cost. Therefore, to define the reward signal in this framework, we also consider the production cost defined as a penalty for agents.*

Table 1 show the cost function based on the amount of production (Cost function), the maximum production power (Maximum power) and the allowed range of production changes in each time step (Maximum change), for the three types of generators.

Generators	Cost function	Maximum power (kW)	Maximum change (kW)
generator1	$5.13x^2 - 10.19x + 29.53$	10800	3000
generator2	$5x^2 - 10x + 29.72$	6300	2000
generator3	$4.94x^2 - 9.92x + 29.794$	5400	1700

Table 1: Generators with different features

4.2 Experimental results

We partitioned each dimension of the continuous three dimensional state space into three parts and define a triangular membership function (as a simple and usual membership function) corresponding to each part (same as Figure 3). The upper bound for the first, second and third dimensions of the state space is equal to the maximum production power of the learning generator, the maximum storage capacity of the battery and the maximum demand load, respectively. We use two first generators presented in Table 1 with one clean energy generator with random production. This amount of energy is modeled as the absolute value of the normal probability density function as pointed out in [3, 28] with mean and standard deviation equal to 0kW and 500kW. Such a generator has a maximum production power that we assume it as twice the standard deviation of the density function. Random fluctuations of the

demand function are also modeled by the normal probability density function with the mean equal to 0kW and the standard deviation of 600kW. In other words, it is possible that demand load at any time step be less or more than the predicted demand by a random value. In this experiment, the maximum battery capacity is defined equal to 3800kW. This capacity is selected based on the minimum value with a good performance. The candidate action sets for the first and second generators are defined as $\{-3000, -1500, 0, 1500, 30000\}$ and $\{-2000, -1000, 0, 1000, 2000\}$, respectively. These sets are chosen based on some experiments in which we tried to reduce the complexity. Since quantities less than 1kW are not noticeable in the power production, we consider the smallest change equal to 1kW (i.e. using round function). In addition, we set the discount factor to $\gamma = 0.95$ and use ϵ-greedy action selection method in all the experiments. ϵ-greedy is an action selection method that allows the algorithm to choose a random action with the probability of ϵ and the action with the highest weight with the probability of $1 - \epsilon$.

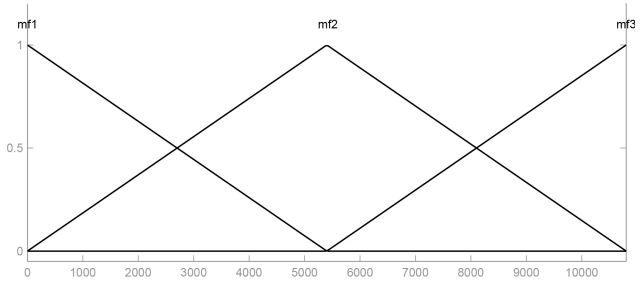


Figure 3: Triangular membership functions

Figure 4 shows the demand function for the first framework. The definition of demand function is based on the general form of consumption. Generating the random values will cause many oscillatory changes in the demand function and will cover all demand functions in the determined range. Therefore, the agent learns to satisfy the demand for a specific range (e.g. one year) with only one trial.

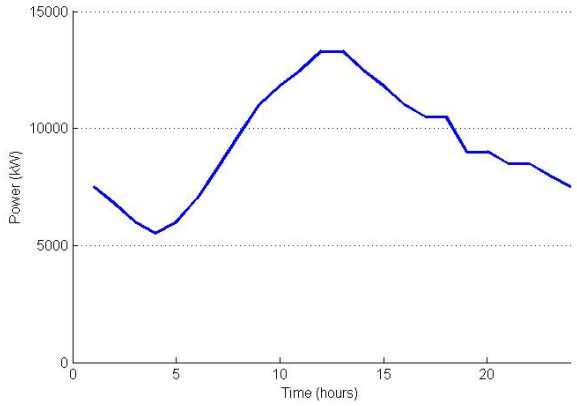


Figure 4: Demand function for framework 1 with 3 generators

The learning process will end after 30 successful episodes (i.e. demand and other hard constraint satisfaction) or after 1000 consecutive episodes. The agent also has an opportunity of 300 episodes in each trial to reach the goal. This experiment has been performed 50 trials independently. Here, each time period is divided into 24 time steps. Based on the required accuracy, the length of time intervals could be decreased to any desirable extent. In each step, the reward signal is defined with $0.5 + \frac{1}{1+error}$ for the case that the generator production is in the allowed range and 0 otherwise. *error* is the difference between the produced energy and the demand load.
After training the agent, the derived policy of each trial is tested on 50 different demand functions, which their values are within the defined range in the training phase. Table 2 presents the results of this experiment for the training and test phases.

Mean episode to learn	Mean error 1	Mean error 2	Mean error 3	Mean error 4
64.9	86.383	43.776	12.988	62.695

Table 2: Mean errors for different scenarios in the first framework

We use different values for grid parameters, in order to study their impacts on the algorithm performance. One can see that in the first test in which its parameters are similar to the training phase, the demand satisfaction error (Mean error 1) is very low and is equal to 86,383kW, which is less than 1%, compared to the maximum possible error (i.e. 9996kW). In an effort to reduce this error, we used some other ideas. In the second test, we assumed that the demand function has the standard deviation of 400kW. As it is observed, if the random variation of the demand function at the learning phase exceeds its actual value, then the results in the test phase will improve (Mean error 2).

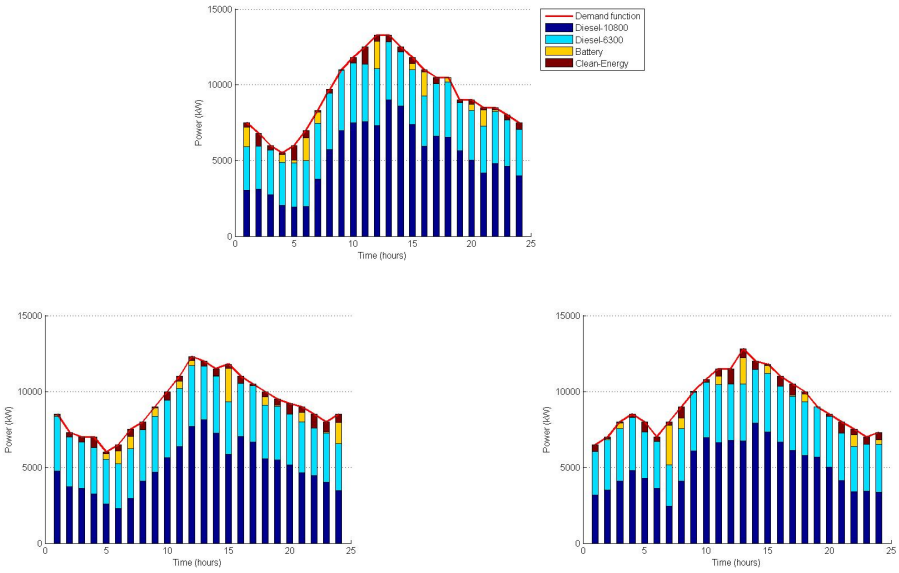


Figure 5: Different demand functions that agents learned to satisfy for the first framework.

Then, we assumed that the demand function does not have any unpredictable fluctuation and the only uncertainty factor is the random production of clean energy. Despite the existence of clean energy generator (i.e. uncertainty sources), it is seen that the average error (Mean error 3) is very low and is close to zero. Finally, we assumed that the parameters of the demand functions and clean energy were the same as the training phase, but we increased the battery capacity to 4500kW. As can be seen, the average error (Mean error 4) has fallen but not to the expected value. Therefore, the main factor of error is the random changes of the demand function and clean energy. By increasing the fluctuation domain of the demand function in the learning phase, we can reduce the error rate at the test phase. In many trials of this test, despite the existence of uncertainty in the environment, the controllable generators have learned to satisfy the demand functions without any error. Error is caused by sudden and severe fluctuations in predicted demand function and produced clean energy, which is apparent in mean error values. To conclude, the proposed algorithm has a high degree of flexibility in learning a range of different functions in the stochastic and non-deterministic domains, and the results demonstrate the efficiency of the method. For a better understanding, Figure 5 presents the results of satisfaction of three different demand functions at different time intervals. It should be noted that these functions are just examples of the numerous demand functions that the agents are able to satisfy without any or with a very small error after just one training trial.

As shown, these three functions (Figures 5a, 5b and 5c) have different peak hours and their fluctuations are also different. In these diagrams, the production rate of all three generators along with the amount of battery consumption is shown. At some hours, some extra energy is generated, which is stored in the battery and can be consumed later when the generators are not able to exactly satisfy the demand. This helps the agent to implicitly approximate the others agents' policies when unpredicted changes exist in the demand function and the clean energy production.
Now, we explain the results in the second framework. In this framework, seven controllable and three uncontrollable generators are used. Controllable generators are selected from Table 1: Three generators of type 1, two generators of type 2 and two generators of type 3. Three uncontrollable generators are set to work with the normal probability distribution function with the standard deviation of 500kW, 600kW and 700 kW. For the demand function, we use the normal probability distribution function with the standard deviation of 700kW. For all normal distributions, we set the mean to zero. According to the definition given at the beginning of this section, what distinguishes the second framework from the first framework is as follows:

- The increased number of controllable generators as intelligent agents, making the problem more difficult to solve, particularly without information sharing,
- The increased number of uncontrollable clean energy generators imposing more uncertainty on the environment,
- The increased range of the demand function fluctuation, increasing the uncertainty,
- The possibility of generators failure and their outage from the energy production process,
- The demand satisfaction with the optimal cost

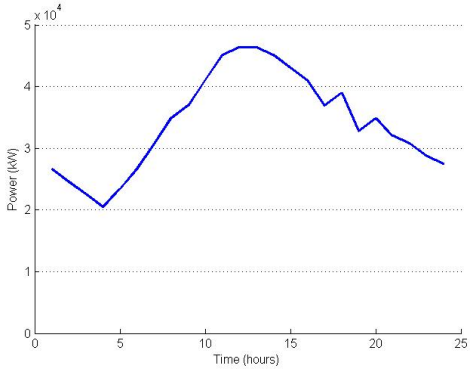


Figure 6: Demand function for framework 2 with 10 generators

Figure 6 shows the basic demand function for the second framework. This is more difficult to learn than the demand function of the first framework in Figure 4. However, in general, due to the unpredicted fluctuations, the demand functions in the learning and test phases are more complex than what shown in Figure 6.
Table 3 presents the results for this experiment. The used parameters are as before. The average number of the needed episodes for learning (each episode is equal to one day) is equal to 55.08, indicating that the learning speed is high. The average error derived from the test, with parameters as the training parameters, is slightly lower than the average of the maximum possible error (i.e. 34747kW) in demand satisfaction, which is 0.36%. Decreasing the standard deviation parameter for demand fluctuations to 500kW improves the test results. By eliminating the fluctuations of the demand function, the environment uncertainty factors are limited to the random values of the clean energy produced by the three uncontrollable generators. This further reduces the mean error of the demand satisfaction (equal to 0.005%, which can be considered almost zero). Increasing the capacity of the battery to 13500kW, while the parameters of random functions related to demand fluctuations and clean energy remain as the training phase, increases the average error to demand satisfaction, which is not desirable. Again, it can be concluded that decreasing the uncertainty in the environment results in decreasing the error of the demand satisfaction.
In this framework, the agents also learn to satisfy a defined range of different demand functions cooperatively, just with one training trial and without information sharing in parallel. Figures 7a, 7b and 7c present an example containing three different functions, which are satisfied by only one learning trial.

Mean episode to learn	Mean error 1	Mean error 2	Mean error 3	Mean error 4
55.08	125.016	105.648	84.028	139.002

Table 3: Mean errors for different scenarios in the second framework

Despite the fact that the complexity of the problem is increased, the agents are able to satisfy the different demand functions in the distributed manner and without information sharing. Diagrams are based on the controllable generators' production, produced clean energy and battery consumption. As mentioned earlier, when the total production is more than the demand load, surplus energy is stored in the battery until it is picked up when needed.

Using the proposed method, the agents are able to satisfy the demand function, in spite of failure of some generators. This is based on the condition that the amount of demand is not more than the sum of the generating capacity of the generators in the process of producing energy. Figure 8 show two examples of offsetting the unpredicted failures of some generators by others. In Figure

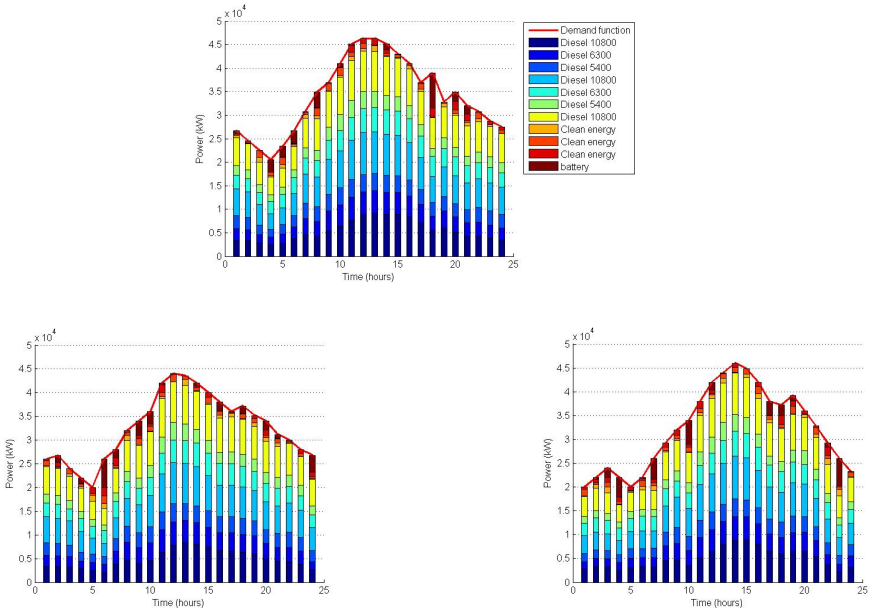


Figure 7: Different demand functions that agent learned to satisfy in the second framework.

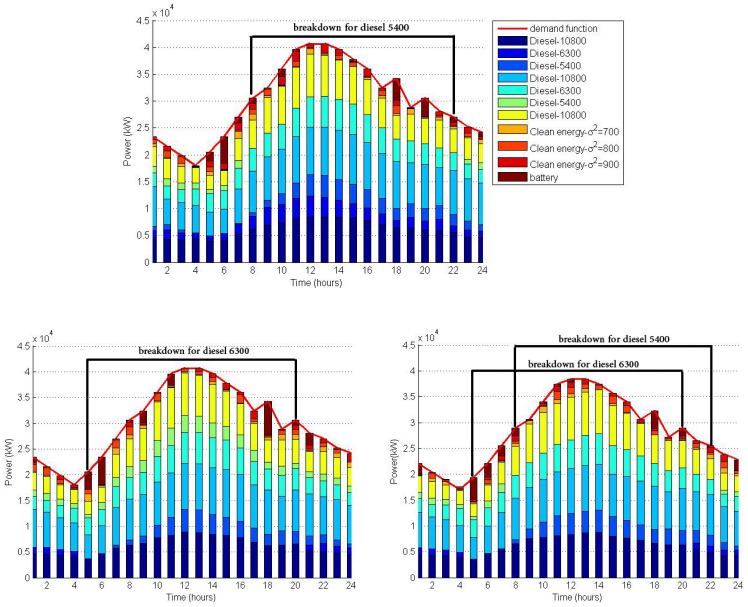


Figure 8: Demand Satisfaction with generators' failure.

8a, the sixth generator (i.e. maximum production power of 5400kW) has failed for 14 time steps (i.e. 14 hours) from time step 8 to 22. This is also true for Figure 8b with the failure of the second generator (i.e. maximum production power of 6300kW) for 15 time steps from time step 5 to 20. In Figure 8c, both failures occurred simultaneously.

The agents will satisfy the demand function based on the total amount of energy produced by the whole system, as long as all generators are working. Nevertheless, when some of the generators suddenly fail, the rest of them compensate for the energy shortfall, and as soon as they return, all agents return to the pre-failure status. Therefore, there is no shortage in demand satisfaction, and the failure of generators will not affect the total energy production and will not be evident for the consumers. This is an example of the self-healing feature of the smart grid.

The other feature of the proposed algorithm is the ability to demand satisfaction with the optimal total cost production. The total cost functions for Figure 8 are shown in Figure 9. To compensate the decrease in the generated energy by the failure of generators, other generators try to satisfy the demand with the lowest imposed cost. In this experiment, we have eliminated the impact of the clean energy production to accurately compare the imposed costs by controllable generators. It should be noted that, to minimize the imposed costs while trying to minimize the error of the demand satisfaction, two goals are defined for the learning process. Thus, this is a multi-objective optimization problem and the reward should be defined in such a way that both of goals can be achieved simultaneously. In this case, we have used summation of the normalized reward for the demand satisfaction and normalized reward associated with the total imposed cost.

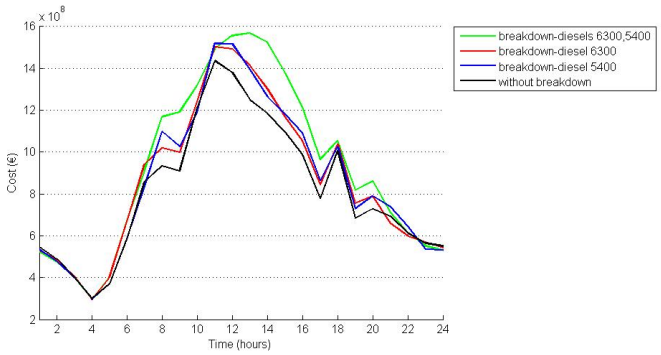


Figure 9: Cost comparison of demand satisfaction with generators' failure

It can be seen that in time steps 1 to 4 where all generators have the ability to generate energy, the total cost of the energy generation is the same in all three charts. In time steps 5 to 8 where the sixth generation fails, the second chart (i.e. failure of the second generator) continue to match the initial diagram, but the other two graphs incorporating the failure of the sixth

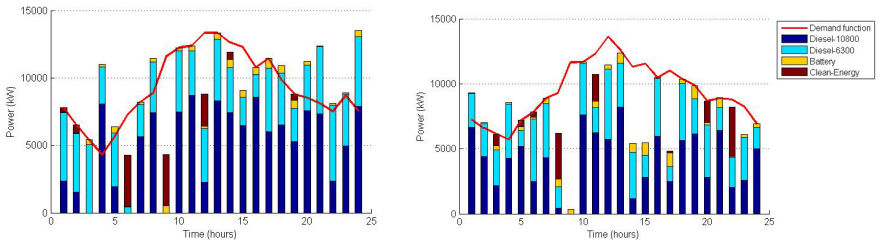


Figure 10: Learned policy by the proposed method in Reference [14] for 3 generators

generator will equally and slightly increase the cost. In time steps 8 to 20, while both the second and sixth generators fail, the increase in costs for the diagram of the simultaneous failure of two generators is clearly lower than the summation of the imposed costs by two single failures of the generators. Therefore, it is seen that the proposed method attempts to have the lowest cost for these failures. Even in some time steps such as 12 and 18, this increase will almost disappear and approaches to zero. Therefore, in such cases, despite the absence of two generators, no extra cost has been imposed. This is due to the presence of the battery in the environment. In other words, due to the fact that in time steps with one or two generator failure, the amount of consumption and storage differs from the presence of all generators. This is the reason for the mismatch in the demand satisfaction pattern in the return period of both generators at the end of a time period (i.e. a day).
We also compared our method to one of the latest proposed methods presented in [14]. Cardinality of the action space is defined equal to 101, imposing considerable complexity while creating a large gap between the actions selectable for generators with high generation. In addition, the maximum allowed episodes in each trial to be defined as 20,000 episodes, showing the low learning speed of this algorithm in comparison to our proposed method having a mean learning speed of less than 65 episodes. This is while, our method considers two types of uncertainty in the environment. In addition, our results are based on the different demand function types generated with a random term, and the result of the mentioned method is based on a fixed demand function. Thus, the proposed method in [14] training the agent for a fixed demand function, while our method simultaneously does this for a range of different demand functions.

With the same setting of framework 1, the method in [14] could not learn to satisfy the demand function in any trials. The demand satisfaction mean errors are high error and unacceptable for example Mean error 1 is equal to 2585kW. Figure 10b shows two samples of the learned policies.
The method in [14] could not learn to satisfy defined violated demand function of framework 2, at all (even using a high capacity battery). The mean errors in

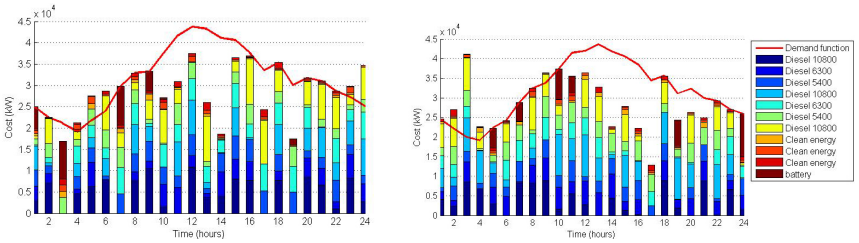


Figure 11: Learned policy by the proposed method in Reference [14] for 10 generators

the test phase are high. For instance mean error 1 is equal to 8373kW (for G(+) as the best result and with battery maximum capacity of 12000kW) which is large and inappropriate error for satisfying the demand function. One can see the samples of the extracted policy by this method in Figure 12.

On the other hand, corresponding production cost to the learned policy is not suitable as it is shown in Figure 12

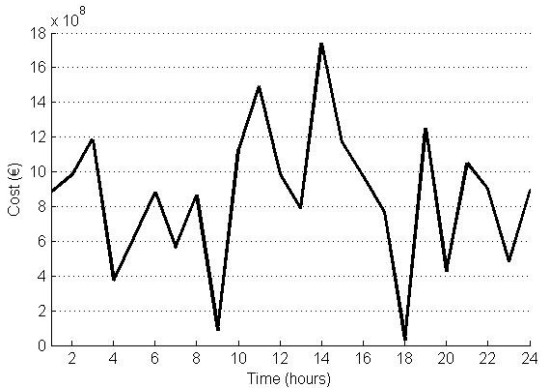


Figure 12: A sample cost function extracted from learned policy by proposed method in Reference [14]

5 CONCLUSION

In this paper, a multi-agent learning algorithm is proposed for the optimization problem of Distributed Stochastic Unit Commitment. The agents learn to satisfy the demand profile with minimum cost while considering the constraints. This algorithm uses reinforcement learning to learn a cooperative behavior in the continuous state-action spaces and do not share information. It is a reward-based multi-agent solution using special reward signal and state of the agent to approximate the system behavior implicitly, despite the presence of uncertainty in the environment. If the number of steps at time interval is

increased, the proposed algorithm could be used as a continuous time solution. The ability of learning a large number of demand functions in a desired range is another advantage of this method. In other words, with one trial of this algorithm, the agents could satisfy the different demand functions for a time interval (including one season or even one year) with the possibility of unpredicted fluctuations in the demand function, in a non-deterministic environment. The experiments in two different frameworks show the acceptable performance of this method in the DSUC problem.
We are going to develop the proposed solution for more complex stochastic unit commitment problems with more objective functions such as minimizing carbon emission. In addition, we will consider microgrids with more uncontrollable energy resources rather than controllable types and also plug-in electric vehicles as a type of energy storage.

References

[1] Agrawal P (2006) Overview of doe microgrid activities. In: Symposium on Microgrid, Montreal, June, vol 23

[2] Amin SM, Wollenberg BF (2005) Toward a smart grid: power delivery for the 21st century. IEEE power and energy magazine 3(5):34–41

[3] Ayodele TR (2015) Determination of probability distribution function for modelling global solar radiation: Case study of ibadan, nigeria. International Journal of Applied Science and Engineering 13(3):233–245

[4] Barker PP, De Mello RW (2000) Determining the impact of distributed generation on power systems. i. radial distribution systems. In: Power Engineering Society Summer Meeting, 2000. IEEE, IEEE, vol 3, pp 1645–1656

[5] Bellman R (2013) Dynamic programming. Courier Corporation

[6] Boutilier C (1999) Sequential optimality and coordination in multiagent systems. In: IJCAI, vol 99, pp 478–485

[7] Claus C, Boutilier C (1998) The dynamics of reinforcement learning in cooperative multiagent systems. AAAI/IAAI 1998:746–752

[8] Dibangoye J, Doniec A, Fakham H, Colas F, Guillaud X (2015) Distributed economic dispatch of embedded generation in smart grids. Engineering Applications of Artificial Intelligence 44:64–78

[9] Fang X, Misra S, Xue G, Yang D (2012) Smart gridthe new and improved power grid: A survey. IEEE communications surveys & tutorials 14(4):944–980

[10] Ghorbani F, Derhami V, Afsharchi M (2017) Fuzzy least square policy iteration and its mathematical analysis. International Journal of Fuzzy Systems 19(3):849–862

[11] Hawkes A, Leach M (2009) Modelling high level system design and unit commitment for a microgrid. Applied energy 86(7):1253–1265

[12] Logenthiran T, Srinivasan D, Khambadkone A, Aung H (2010) Multi-agent system (mas) for short-term generation scheduling of a microgrid. In: Sustainable Energy Technologies (ICSET), 2010 IEEE International Conference on, IEEE, pp 1–6

[13] Logenthiran T, Srinivasan D, Khambadkone A, Aung H (2010) Scalable multi-agent system (mas) for operation of a microgrid in islanded mode. In: Power Electronics, Drives and Energy Systems (PEDES) & 2010 Power India, 2010 Joint International Conference on, IEEE, pp 1–6

[14] Mannion P, Mason K, Devlin S, Duggan J, Howley E (2016) Dynamic economic emissions dispatch optimisation using multi-agent reinforcement learning. In: Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)

[15] Nagata T, Ohono M, Kubokawa J, Sasaki H, Fujita H (2002) A multi-agent approach to unit commitment problems. In: Power Engineering Society Winter Meeting, 2002. IEEE, IEEE, vol 1, pp 64–69

[16] Nikovski D, Zhang W (2010) Factored markov decision process models for stochastic unit commitment. In: Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on, IEEE, pp 28–35

[17] Nowak MP, Römisch W (2000) Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. Annals of Operations Research 100(1-4):251–272

[18] Ozturk UA, Mazumdar M, Norman BA (2004) A solution to the stochastic unit commitment problem using chance constrained programming. IEEE Transactions on Power Systems 19(3):1589–1598

[19] Papavasiliou A, Oren SS (2013) Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network. Operations Research 61(3):578–592

[20] Saravanan B, Das S, Sikri S, Kothari D (2013) A solution to the unit commitment problem–a review. Frontiers in Energy 7(2):223

[21] Soltani Z, Ghaljehei M, Gharehpetian G, Aalami H (2018) Integration of smart grid technologies in stochastic multi-objective unit commitment: An economic emission analysis. International Journal of Electrical Power & Energy Systems 100:565–590

[22] Stone P, Veloso M (2000) Multiagent systems: A survey from a machine learning perspective. Autonomous Robots 8(3):345–383

[23] Sutton RS, Barto AG (1999) Reinforcement learning. Journal of Cognitive Neuroscience 11(1):126–134

[24] Takriti S, Birge JR, Long E (1996) A stochastic model for the unit commitment problem. IEEE Transactions on Power Systems 11(3):1497–1508

[25] Wang Q, Wang J, Guan Y (2013) Stochastic unit commitment with uncertain demand response. IEEE Transactions on Power Systems 28(1):562–563

[26] Wang Y, Xia Q, Kang C (2011) A novel security stochastic unit commitment for wind-thermal system operation. In: Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011 4th International Conference on, IEEE, pp 386–393

[27] Xiao L, Xiao X, Dai C, Pengy M, Wang L, Poor HV (2018) Reinforcement learning-based energy trading for microgrids. arXiv preprint arXiv:180106285

[28] Yürüşen NY, Melero JJ (2016) Probability density function selection based on the characteristics of wind speed data. In: Journal of Physics: Conference Series, IOP Publishing, vol 753, p 032067