Alexandria Engineering Journal 82 (2023) 145–153

Contents lists available at ScienceDirect

Alexandria Engineering Journal

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

Original Article

Novel deep deterministic policy gradient technique for automated micro-grid energy management in rural and islanded areas

Lilia Tightiz [a], L. Minh Dang [b,c], Joon Yoo [a,*]

[a] School of Computing, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam-si, 13120, Gyeonggi-do, Korea
[b] The Institute of Research and Development, Duy Tan University, Da Nang, 550000, Viet Nam
[c] Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Viet Nam

ARTICLE INFO

Keywords:
Deep deterministic policy gradient
Energy management system
Microgrid
Responsive loads
Transfer learning

ABSTRACT

The microgrid enhances power grid reliability, resiliency, and sustainability, particularly in rural and islanded areas with limited manual network management. However, microgrid energy management systems (EMS), especially in islanded mode, require precise and reliable techniques to prevent severe blackouts/brownouts. This paper presents a novel deep deterministic policy gradient (DDPG) algorithm to schedule EMS for the autonomous microgrid in real-time. Our solution utilizes deep reinforcement learning (DRL) to converge model-free, sequential, random, and continuous characteristics of the microgrid. Additionally, we use reward shaping and transfer learning attachment to DDPG to support microgrid performance restrictions and minimize load shedding during peak hours. This solution offers an efficient training process comparable to other DRL techniques in simplicity, less computation, and supporting future system extension. Residential Gasa Island microgrid profile characteristics have been selected and tested to examine the proposed approach. Results demonstrate the high efficiency and accuracy of the proposed technique compared to existing methods.

1. Introduction

As a result of their numerous economic and environmental advantages, renewable energy resources (RES) are predicted to surpass other energy sources within a few years. In addition to being CO_2-neutral, nature-based, and eternally available, RESs also provide an excellent labor market for their maintenance and installation [1]. Microgrid is a popular form of deploying RESs, which enables their integration into the grid. IEEE Standard 2030.7–2018 defined the microgrid as a "group of interconnected loads and distributed energy resources with clearly defined electrical boundaries that act as a single controllable entity concerning the grid and can connect and disconnect from the grid to enable it to operate in both grid-connected or island modes" [2]. Microgrids became popular for supplying electricity to rural and islanded areas through the application of RESs, particularly in the case of releasing power systems from the cost of transmission line development. The main drawback of RESs is their stochastic characteristics. With the advent of energy storage systems (ESS), randomness in RES generation can heal. It is possible to compensate for the absence of RESs by scheduling ESSs charging and discharging. In addition to ESSs, loads can contribute to this compensation via demand response (DR). This approach requires a robust management system to consider all elements of the microgrid constraints and behaviors [3]. Hence, in this paper, we consider scheduling the microgrid EMS. Researchers have applied different optimization methods to improve the performance of the microgrid EMS up to now. Global optimization and traditional machine learning techniques are commonly used methods [4], [5], [6], [7], [8]. Solving EMS is a non-deterministic and continuous problem due to the stochasticity of RES output power and loads in the microgrid. Therefore, traditional machine learning and global optimization methods cannot defeat the uncertainty in the EMS problem. In addition, these methods do not support real-time issues, and each adjustment in the microgrid elements requires those solutions to be run and updated fully or partially.

The curse of dimensionality and difficulty in deploying traditional machine learning and meta-heuristic algorithms can be defeated with the contribution of supervised learning and heuristic algorithms when the different characteristics of the environment are known [9,10]. Reinforcement learning (RL) is highly effective in handling model-free and stochastic aspects of microgrids in situations where the environment is not entirely observable [11]. In this study, we mainly concentrate on the microgrid as a host of autonomous consumers in rural and remote places where the effectiveness of the EMS is crucial due to the absence

* Corresponding author.
E-mail address: joon.yoo@gachon.ac.kr (J. Yoo).

https://doi.org/10.1016/j.aej.2023.09.066
Received 13 May 2023; Received in revised form 27 August 2023; Accepted 25 September 2023
Available online 6 October 2023

of a backup grid. Therefore, we set out this study to resolve challenging aspects of EMS arrangements for microgrids with RL. These challenges include:

1. EMS provision in microgrid is a high-dimension and continuous problem.
2. The behavior of elements such as RESs and loads is stochastic.
3. EMS provision requires a safe online solution respecting microgrid elements' critical restrictions.
4. High-dimensional characteristics of the microgrid environment require a long process of agent interaction with the environment and result in high computational costs.
5. The EMS solution ought to facilitate any future extensions without requiring significant effort.

Using RL to tackle EMS problems has drawn considerable interest. To regulate the unpredictable behavior of RES, Q-learning, an RL technique, was implemented in [12] to manage the performance of photovoltaic (PV), ESS, and load in a standalone micro-gird. However, Q-learning suffers from the curse of dimensionality because of the extensive work required to establish the Q-table and calculate the value of each state and action. Therefore, it cannot support a precise model of microgrids that considers the different probability behaviors of RESs and loads. A different approach in RL involves using neural networks to aid in Q-value determination for high-dimensional state and action pairs in an environment. This method is known as deep reinforcement learning (DRL) [13]. In their study, Coa et al. [14] utilized deep Q-network (DQN) as a DRL technique to facilitate microgrid participation in the electricity market. This approach supported numerous state and action pairs within the microgrid environment.

Harrold et al. [15] tested the performance of the rainbow DQN with a range of deep Q-learning-based approaches. In this research, the authors demonstrated that rainbow DQN is superior in scheduling ESS to manage the uncertainty of wind turbines (WT), PV, and loads. In another study by Ji et al. [16], the uncertainty in load profile, RES, and the market price were considered to offer cost-effective EMS by DQN. EMS was a member of the EMS community in [17] and optimized with a double deep Q-network (DDQN). The main idea behind this study was ESS performance improvement through minimizing generation costs in grid-connected mode and load curtailment minimization in islanded mode. Electricity market prices, loads, and RES were uncertain in the environment's elements states. While DQN techniques are effective for discrete action spaces, they do not fully account for the continuous behavior of microgrid elements. To address this issue, we implemented DDPG, which supports the model-free, stochastic, and continuous nature of the action space of the microgrid. Gao et al. [18] explored how DRL could be used to optimize the EMS of a microgrid. Specifically, the authors used DDPG to regulate the building heating and cooling energy usage. DDPG was a solution to schedule the island microgrid in references [19,20]. Despite using standard DRL-based solutions, the problem of lengthy learning processes due to element restrictions and their effect on the stability of the learning process still exists. Three approaches can be taken to tackle these issues: reward shaping, constraint policy optimization, and Lagrangian-based DRL.

The simplest way in reward shaping approach includes motivating agents to respect microgrid restrictions by assigning penalties in reward functions. Penalty assignment to extreme situations, such as the state of charge (SoC) constraints and balance of energy consumption and generation, in the island microgrid environment will result in sparse rewards. In these severe circumstances, the agent should receive very negative punishment for not respecting the defined boundaries and concluding the learning process episode as soon as possible to speed up the learning process convergence. The sparse reward drives the agent training to poor solutions and instability in the learning process. Another method involves using Lagrangian-based DRL techniques to create a primal-dual problem and incorporate environmental limitations while training the agent. Yan et al. [21] respect the operating constraints of the power system as a penalty in the learning process of DDPG agents following Lagrangian relaxation to optimize load flow. Zhang et al. [22] applied the Lagrangian penalty in the soft actor-critic (SAC) method and rule-based filter to respect electric vehicles (EV) charging/discharging limitations in the EMS scheduling of a residential microgrid. However, these remedies will deteriorate the system's complexity due to adding Lagrangian coefficients. Proximal policy optimization (PPO) and trust region policy optimization (TRPO) methods are examples of constrained policies that fall short of the Lagrangian approach in terms of effectiveness [22]. Additionally, EMS should be the feasible method that supports the system's future development. The aforementioned approach imposes more coefficients and computations on the solution, making modifying the solution more challenging in any environment extension.

To find a tradeoff between accuracy and simplicity and facilitate extending solutions for future development of the system, in [23], the long learning process of DDPG for EMS provision in the microgrid was dominated by instance-based transfer learning by evaluating the similarity between target and source domains. However, this method still suffers from consuming large amounts of memory and computational resources to calculate similarity. Furthermore, this approach's accuracy depends on precise predictions of future RES power production and loads, and the uncertainty in their behavior has declined. To defeat these issues, we combined a parameter-based transfer learning method with the simplicity of penalty assignment to extreme behavior to enhance DDPG performance and minimize effort for system future extension in our study. As a result, the agents will gain meaningful information from simple tasks and produce better results in a more sophisticated environment. Therefore, we divide the task of the system into different subtasks. To this end, our method separates extreme conditions from normal ones, and the weights of subtask 1's trained network transfer to subtask 2 to foster the learning rate and stability of the system. Table 1 outlines the comparison between our approach and related work. Additionally, to provide our proposed method as an online solution, we deploy historical data of the Gasa Island microgrid in Korea to consider uncertainty in loads and RES generation.

Therefore, the contribution of this paper is as follows:

- Improve consumer comfort and reduce peak hour DR requirements by defining an appropriate reward function.
- Provide a safe solution for the EMS problem based on DDPG that respects environment restrictions by reward shaping.
- Justify the reward shaping approach to support an extendable solution for the EMS problem of the island microgrid.
- Enhance DDPG with transfer learning to tackle the long learning process and instability issues due to respecting the environment restrictions.

The outline of the paper is as follows. We clarify the technical characteristics and constraints of the microgrid elements and the solution algorithm based on DDPG attaching transfer learning in Section 2. After proposing and analyzing the results in Section 3, we draw our conclusions and future works in Section 4.

2. Applied method

2.1. Microgrid structure and constraints

The microgrid is a collection of loads and generators. Power generators of the microgrid include RESs and conventional generators, which the latter ones applied to dominate the intermittency and stochastic characteristics of the former ones. ESS is the other element of the microgrid, vastly used to make RES dispatchable. In other words, ESS supports the microgrid in the unavailability situation of RES. EV can be another form of ESS in the microgrid. We considered a nature-based microgrid for rural areas, including RES, responsive load, emergency

Table 1
Related work comparison.

EMS Optimization Solutions Methods	Quality Performance in Microgrids EMS						
	1	2	3	4	5	6	7
Global optimization and traditional machine learning techniques [4–8]	Poor	Poor	Good	Good	Poor	Poor	Poor
RL [12]	Poor	Good	Poor	Moderate	Moderate	Poor	Poor
Standard DRL [14–20]	Good	Good	Good	Moderate	Moderate	Poor	Poor
Lagrangian DRL [22]	Good	Good	Good	Moderate	Poor	Poor	Poor
Instance-based Transfer Learning+ DRL [23]	Good	Good	Good	Moderate	Good	Good	Poor
Our solution Reward Shaping+ Parameter-based Transfer Learning+DRL	Good	Good	Good	Good	Good	Good	Good

1: Support high-dimensional problem.
2: Support stochastic behavior of RESs and loads.
3: Support Continuous action space.
4: Respecting element limitations.
5: Low computation cost and complexity.
6: Online solution.
7: Sustainable for future extension of the system.

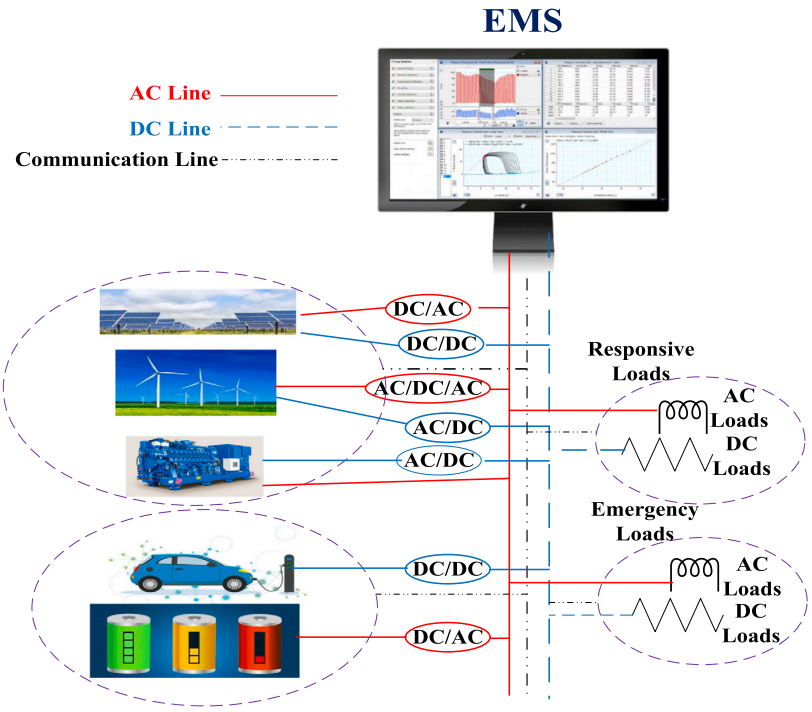

Fig. 1. Microgrid structure.

load, diesel generator (DG), and ESS, as shown in Fig. 1. Responsive loads participate in the DR program through load shedding. ESS is controlled by the level of available SoC and charged by RES. Therefore, ESS has a stochastic characteristic. EVs are categorized in the ESS group during idle time and during travel time, therefore, we postpone their role. We considered the battery energy storage system (BESS) to represent the role of ESS in our proposal.

$$-P^{ch/dis}_{bat,min} \le P^{ch/dis}_{bat}(t) \le P^{ch/dis}_{bat,max}, \tag{1}$$

$$SoC_{bat,min} \le SoC_{bat}(t) \le SoC_{bat,max}, \tag{2}$$

$$SoC(t) = SoC(t-1) + \tau(\eta^{ch}_{bat} P^{ch}_{bat}(t) - P^{dis}_{bat}(t)/\eta^{dis}_{bat}), \tag{3}$$

$$Cost_{bat}(t) = (P^{ch/dis}_{bat}(t) + E_{bat}(t)\omega_l)\rho\tau, \tag{4}$$

$$\rho = invest_{initial}/(E_{bat,rated} NUM^{ch,dis}_{cycle}). \tag{5}$$

Equation (1) reveals the battery power charging and discharging constraints. SoC boundaries are according to (2). At each time step, SoC calculated w.r.t. its previous time step amount as shown in (3) where P^{ch}_{bat} and P^{dis}_{bat} are battery charging and discharging power. τ is the time slot and η^{ch}_{bat} and η^{dis}_{bat} are battery charging and discharging efficiency. However, in our study, BESS is charged by RES. Consequently, in each time step, the SoC amount depends on the action of the system. The cost of power delivered by BESS is calculated by (4) where ρ is the battery lifetime degradation coefficient, E_{bat} is the amount of stored energy in the battery, ω_l is the battery leakage loss, $invest_{initial}$ is the initial investment for battery provision, and $NUM^{ch,dis}_{cycle}$ is the number of full charge and discharge cycle of the battery.

The constraint associated with RES is shown in (6). In this paper, RESs are PV and WT, which supply loads with higher priorities.

$$P^i_{RES,min} \le P^i_{RES}(t) \le P^i_{RES,max}, 1 \le i \le N, \tag{6}$$

where, N is the number of RESs and P_{RES} denotes their output power. DG is the other source of power generation in the microgrid serving demand in RES unavailability. The quadratic equation provides the cost of DG's power generation according to the following equations [27].

$$Cost_{DG}(t) = a_1 + a_2 P_{DG}(t) + a_3 (P_{DG}(t))^2 \tag{7}$$

$$P_{DG,min} \le P_{DG}(t) \le P_{DG,max} \tag{8}$$

where a_1, a_2, and a_3 are factors for the fuel cost of DG. In the proposed microgrid, there are two categories of loads. Emergency loads, which should be responded to in all situations, and responsive loads, take part in load shedding during contingencies. The power generation in the microgrid should respond to the loads in all situations with the contribution of BESS, DG, and load shedding. This fact as the microgrid constraints is shown in (9), (10).

$$\sum_{i=1}^{n} P^i_{RL} + \sum_{j=1}^{m} P^j_{EL} = P_{Demand}, \tag{9}$$

$$(\sum_{i=1}^{N} P^i_{RES}) - P_{Demand} = P_{net}, \tag{10}$$

where, n and m are the number of responsive loads and emergency loads, respectively. P_{RES} is power generated by RES, P_{bat} is power absorbed by or injected from BESS, P_{RL} is the power consumed by responsive loads, P_{EL} is the amount of emergency loads, respectively. Additionally, P_{net} is the amount of power shortage and surplus of the microgrid without DG and BESS contributions. To respect power balance in the microgrid, the sum of all power generation and consumption should be equal to zero as follows.

$$P_{net}(t) + P_{DG}(t) \pm P_{bat}(t) = \zeta, \forall t \quad \zeta = 0, \tag{11}$$

2.2. Microgrid's Markov decision process

Markov decision process (MDP) arrangement for the microgrid is the first step of implementing the RL process. MDP is defined by the state, action, transition function, and reward in the form of 4-tuple {S, A, T, R}. In each state, the amount of power consumption, production, or curtailment of the elements of the microgrid and the time step should be determined.

$$S = [P^i, SoC_{bat}, \tau], \tag{12}$$

$$P^i \in \{P^i_{RES}, P^i_{bat}, P^i_{DG}, P^i_L\}, \tag{13}$$

$$P_L \in \{P^i_{RL}, P^i_{EL}\}. \tag{14}$$

The action of the agent in each state is according to the (15).

$$A = [A_{bat}, A_{DG}, A_{Load}], \tag{15}$$

where, A_{bat} is the charging, discharging, or idle operation of BESS, A_{DG} is the DG status, A_{load} is responsive loads curtailment, and A_{RES} is the accessibility of RES. The transition function determines the next state (S') of the agent after action selection and recognized by P_a (S, S'). Since in this paper, we assume RES and load behaviors are stochastic, the transition function is unknown. The main objective of the understudy autonomous microgrid, as shown in (16), is minimizing the cost of using BESS while applying penalties for the contribution of loads in energy provision during peak hours.

$$min \sum^{\tau} \Big(P_{bat,dis}(t) Cost_{bat}(t)\alpha + (P_{net}(t) - P_{bat,dis}(t) - P_{DG}(t))\beta + P_{DG}(t) Cost_{DG}(t)\gamma \Big), \tag{16}$$

where, $\alpha < \beta \ll \gamma$, and α, β, and γ are coefficients that determine the environment penalties for the contribution of BESS, load shedding, and DG, respectively. Fig. 2 represents the policy for determining α and β according to minimum load curtailment. Since the minimum effect on consumers has been considered during contingencies, we apply a coefficient for training our model to reduce the integration of loads in energy shortage compensation, particularly during peak hours. The other way round, in non-peak hours, we try to preserve the stored energy in BESS storage and let more loads contribute to energy provision. We also define penalty terms to respect SoC and power balance limitation in the microgrid according to the following.

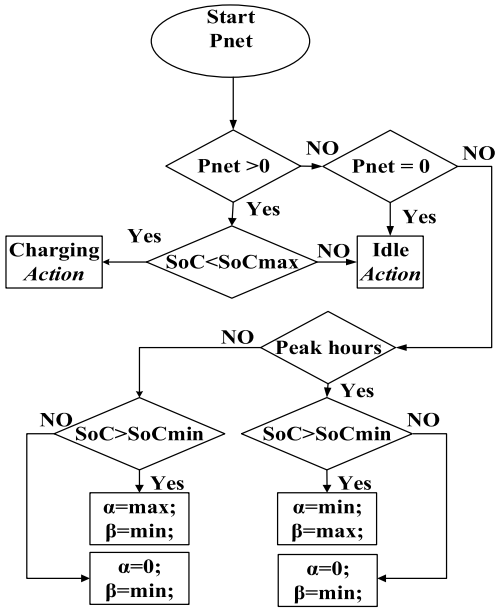

Fig. 2. Reward function coefficients definition.

$$Penalty = \varepsilon, \text{ if } \neg(2) \;||\; \zeta \neq 0, \tag{17}$$

2.3. Algorithm solution

2.3.1. DDPG

DDPG is an actor-critic and deep-learning-based algorithm. DDPG takes advantage of the solutions, namely replay memory and deploying four separated deep neural networks, including Q-network (θ^Q), deterministic policy function (θ^μ), target Q-network ($\theta^{Q'}$), and target policy network ($\theta^{\mu'}$) to overcome instability of Q-learning based training process [24]. The value in Q-network as in other value-based techniques updates by the Bellman equation as follows.

$$y_i = r_i + \gamma Q'(S_{i+1}, \mu'(S_{i+1}|\theta^{\mu'})|\theta^{Q'}), \tag{18}$$

where γ is a discount factor of future rewards. Since DDPG follows the actor-critic method, Q-values of the next state target value networks, i.e., actor and critic, are calculated according to (19) and (20).

$$\theta^{Q\prime} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \quad \tau << 1, \tag{19}$$

$$\theta^{\mu\prime} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}, \quad \tau << 1. \tag{20}$$

where τ is a coefficient to provide a soft alter in Q-network and policy network weights. In DDPG, the actor policy is updated by minimizing the loss function of the original Q-value and updated according to (21).

$$\frac{1}{N}\sum_i (y_i - Q(S_i, a_i|\theta^Q))^2, \tag{21}$$

2.3.2. Transfer learning

In transfer learning, there is a source agent A_s with prior knowledge D_s. The target agent A_t applies the D_s to fasten the learning rate. From

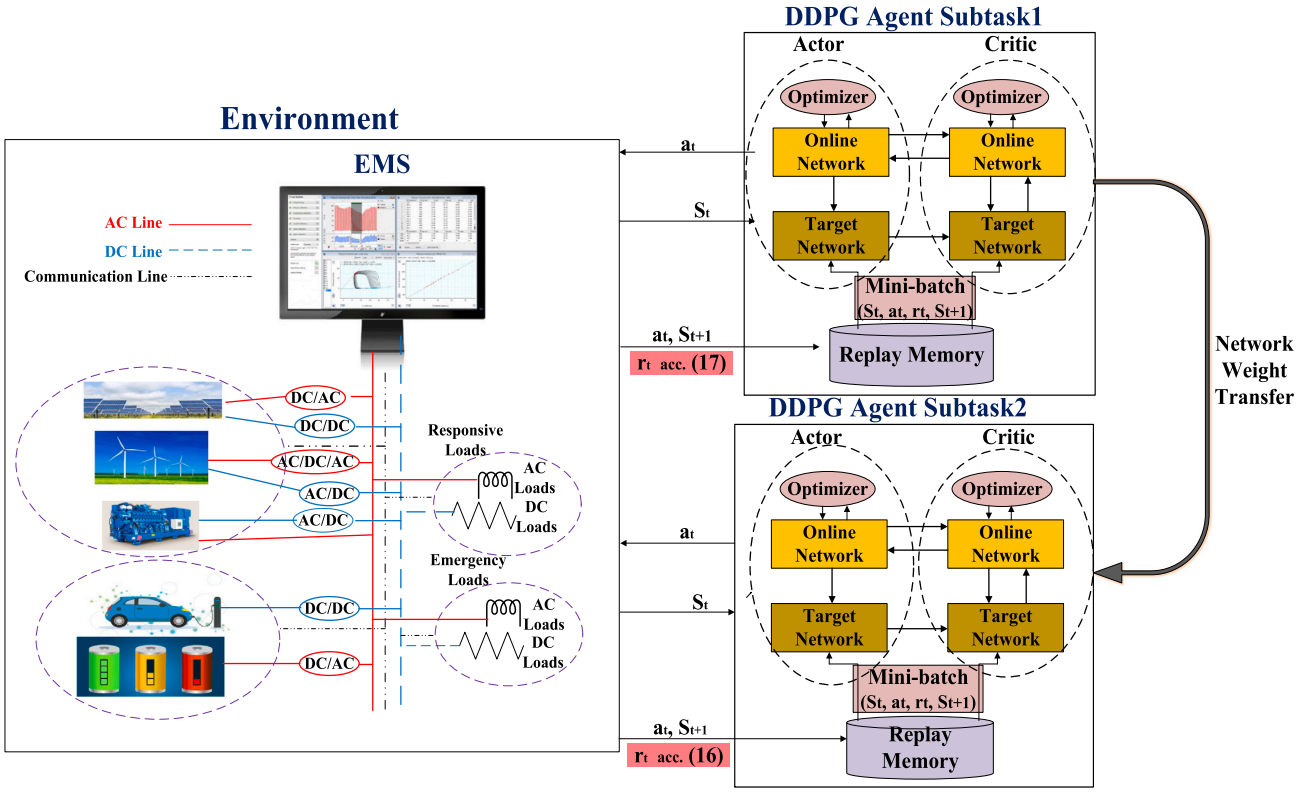

Fig. 3. Microgrid EMS training based on transfer learning-DDPG.

the RL point of view, the optimal policy of target agent π^* facilitates the contribution of exterior knowledge D_s, and interior knowledge D_t as follows.

$$\pi^* = arg_\pi max \mathbb{E}_{s\sim\mu_o^t, a\sim\pi}[Q_A^\pi(s,a)], \tag{22}$$

where,

$$\pi = \phi(D_s \sim A_s, D_t \sim A_t) : S^t \rightarrow A^t, \tag{23}$$

(23) delineates a function that based on learning from D_t and D_s maps the states to actions for the target agent A_t.

2.3.3. Transfer learning integration to DDPG

In this paper, our model uses DDPG, a subset of DRL, to predict state-value functions using deep neural networks. However, considering reward shaping technique tasks of learning SoC-level restrictions and power balance will result in penalties that terminate the agent from the learning process. Therefore, the agent will not have enough chance to explore the environment accurately and find the best actions meet minimize DG costs and DR implementation in peak hours. We divided the EMS optimization learning process into two subtasks. As shown in Fig. 3, two agents in the same environment try to find the best actions. In subtask 1, the agent will receive rewards based on respecting restrictions related to the power balance and SoC range. In this case, we will train the deep neural networks by scheduling power resources and assigning rewards by (17). In subtask 2, according to the hard weight-sharing approach of parameter-based transfer learning, the exact weight of the deep neural networks of subtask 1 will transfer to sub-task 2. Following that, we will train the second agent to respect giving priority to the customer supplying satisfaction during peak hours and minimize the cost of DG and battery degradation when it receives the reward by (16). The pseudo-code of our solution method Reward shaping+Transfer learning+DDPG (RS+TL+DDPG) is according to Algorithm 1.

Algorithm 1 RS+TL+DDPG.

Arrange i-number of subtasks
for i = 1 to 2 **do**
  Initialize Q_θ, $Q'_{\theta\leftarrow\theta}$, μ_θ, $\mu'_{\theta\leftarrow\theta}$, $\mathcal{D}$
  for episode = 1 to E **do** Initialize random process $\mathcal{N}$ Generate S_t based on (12)
    for $t = 1$ *to* T **do** Generate a_t based on (15) using current policy and exploration noise ($a_t = \mu(S|\theta^\mu) + \mathcal{N}_t$)
      if Subtask 1 **then**
        Calculate r_t by (17)
      else if Subtask 2 **then**
        Calculate r_t by (16)
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
        Sample minibatch of transitions
      if *episode terminates at step* $t+1$ **then**
        $y_j \leftarrow r_t$
      else if then $y_j \leftarrow (r_t + \gamma max Q(s_{t+1}, a_t|\theta'_t)$
      Update critic network by (21)
      Update policy network using sampled policy gradient by $\nabla_{\theta_\mu}\mathcal{J}(\theta) \approx \frac{1}{N}\sum_i[\nabla_a Q(s,a|\theta^Q, s=s_i, a=\mu(S_i))\nabla_{\theta_\mu}\mu(S|\theta^\mu, s=s_i)]$
      Update critic and policy target network by (19), (20)
    Store network parameters and load sub-task i+1
    Transfer network and hyper-parameters of subtask i to subtask $i+1$

3. Results discussion

In this section, we describe the results of utilizing our solution algorithm in solving the EMS problem of the Gasa Island microgrid. It is noted that we carried out our environment on MATLAB Simulink (R2020b).

3.1. Case study specifications

As a case study, we considered Gasa Island, which is a standalone microgrid in Korea. The load profile in our under-study islanded microgrid is according to Fig. 4. According to Table 2, Gasa Island is equipped with 314 kW PV, 400 kW WT, 3 MWh BESS, and 300 KW DG to supply mainly residential loads with 173 kW peak load [3]. There is an EMS unit to control and monitor the islanded microgrid elements.

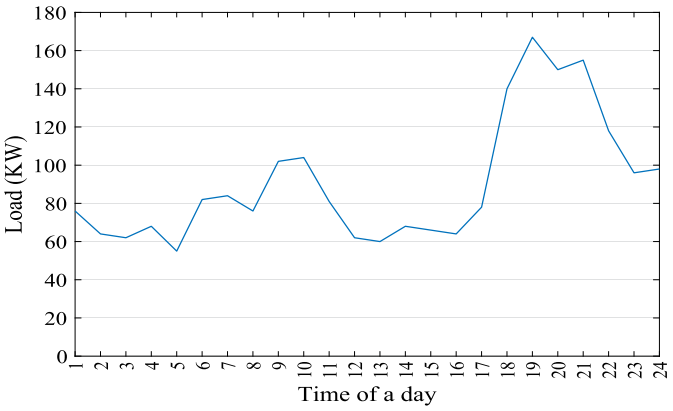

Fig. 4. Typical Korean islanded microgrid average daily load profile.

Table 2
Case study microgrid specifications.

DG	P_{min} (kW)		70
	P_{max}(kW)		300
	Quadratic coefficient	a_1	1.3
		a_2	0.0304
		a_3	0.00104
BESS	SoC_{min} (%)		20
	SoC_{max}(%)		90
	$P_{ch,max}$(kW)		200
	$P_{dis,min}$(kW)		-200
	efficiency		0.9
	ρ(KRW/MW)		100%
Hyper parameters	Learning rate		0.0001
	γ		0.9
	Size of (D)		50,000
	Number of training episode		5,000
	mini-batch size		128
	Number of hidden layers		2
	Number of neurons in hidden layers		128
	Activation function		ReLU
	Optimizer		SGD

As we discussed before, our microgrid environment is model-free concerning stochastic characteristics of loads and RESs output power. We deployed WT and PV output power during 2020 obtained using the Ninja app based on installed RESs capacity and Korea weather conditions [25]. To implement DR, we applied 15% shedding to the average daily load profile in our islanded microgrid model [26]. For the sake of simplicity, we split the continuous process of charging and discharging BESS into 401 steps in the span of [-200 KW, 200 KW]. All the hired neural networks are arranged by two fully connected hidden layers with ReLU as an activation function and stochastic gradient descent (SGD) as an optimizer. The replay buffer size is 50,000, where the number of training episodes and minibatch size are 10,000 and 128, respectively.

Action selection in DDPG follows the epsilon greedy strategy. We allocated 1 to epsilon in 100 initial episodes to select action without noise and decayed that gradually in further episodes to select noised actions.

3.2. Hyperparameters justification

Table 2 delineates the hyper-parameters of the proposed algorithm. We followed an empirical approach to tune the neural network and the DDPG algorithm parameters. Fig. 5 and Fig. 6 present the outcomes of utilizing this technique to determine the neural network size. We began evaluating the training process by starting with fewer hidden layers and neurons for the neural networks we hired. We then assessed the learning process's performance based on the rewards it acquired. According to the data presented in Figs. 5 and 6, we conclude that the neural network with two hidden layers and 128 neurons is optimal considering the balance between network size and performance which offers DDPG target achievement in 1,000 episodes. It is also observed more hidden layers will result in latency in the learning process. The main issue in DRL is stability and convergence. As the neural network increases in layers from two to three, the DDPG learning process stability is negatively affected. The learning process is disrupted due to the increasing variations in each bootstrap caused by the higher number of weights in a three-layer network compared to a two-layer network. These figures also reveal as we set the epsilon greedy for the agent, it explores to learn the environment in initial episodes, therefore obtaining fewer rewards. The agent will follow a greedy strategy as the number of episodes increases. Since there is noise in the gradient estimator, fluctuation is observed in DDPG execution.

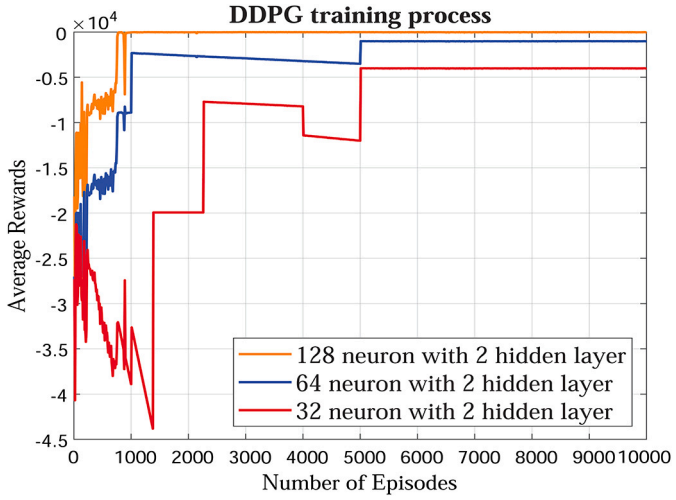

Fig. 5. DDPG training process average rewards with 2 hidden layers and different neuron numbers.

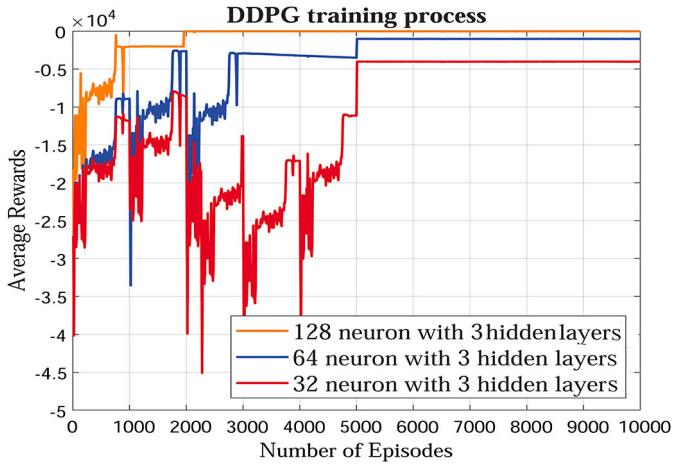

Fig. 6. DDPG training process average rewards with 3 hidden layers and different neuron numbers.

3.3. Proposed algorithm training process results in comparison with benchmark algorithms

Fig. 7 compares our solution with various DQN-based methods, including DQN, DDQN, and standard DDPG. In general, RS+TL+DDPG has a faster convergence rate and higher rewards than the other methods. DQN has the most prolonged training process and oscillation, as expected. The DQN learning process takes around 2500 episodes. While its structure improved in DDQN by adding replay memory to keep the record of previous experience in the environment and separating the Q-

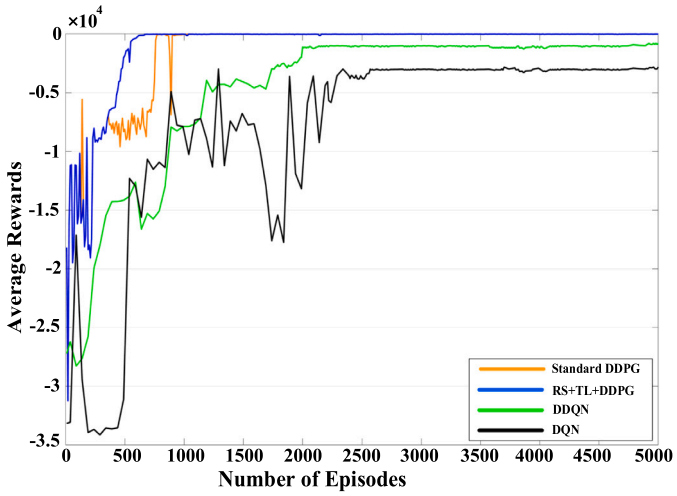

Fig. 7. RS+TL+DDPG and benchmark algorithms training process performance comparison.

network and target network to prevent correlation between them learning process dropped to 2000 episodes with less fluctuation. By contrast, DDPG could gain higher rewards due to its better exploration process by adding noise to action selection compared to DQN and DDQN. However, DDPG's learning process fluctuates because it is sensitive to hyperparameters. The learning process of DDPG experienced a 50% reduction from 1000 to 500 episodes in RS+TL+DDPG with an improvement in stability and decreasing fluctuations in the learning process in light of using transfer learning to facilitate respecting system limitations. Our proposed approach has a faster learning speed due to its reduced fluctuation during the training process. This is evident in Fig. 7, where we compared the DDPG training period from episode 200 to 500. Using reward shaping and implementing it through transfer learning effectively prevented the agent from terminating the learning process early and experiencing fluctuations. As a result, the RS+TL+DDPG approach offers greater efficiency in exploring the environment, as evidenced by the higher rewards in the compared episodes.

3.4. Proposed algorithm online performance in comparison with benchmark algorithms

We selected three sequential days of our test data set and applied RESs output, SoC of BESS, power demand, and output power of DG as a set of states of the environment to check our trained agent online performance. Fig. 8 compares the EMS scheduling results of DDPG and RS+TL+DDPG agents for the three sequential days. These figures include the surplus power of RESs output calculated by (10), SoC of BESS, P_{DG}, and power balance of resources in the microgrid. Although Fig. 8 reveals the microgrid's elements respect the power balance in energy provision in both algorithms, it can be observed from Fig. 8a that DDPG could not meet the SoC limitation. The transfer learning process empowered DDPG by sequential training to respect SoC limitation according to Fig. 8b.

If we look in-depth in Fig. 8, it reveals both algorithms give priority to BESS charging and discharging in case of power surplus and shortage, respectively. DG is the second source of energy that compensates power shortage of RESs when the SoC of BESS is not enough. Also, we noted that none of the scenarios involved EMS requesting load shedding, which was the main objective of the microgrid environment under study.

Table 3 shows the RS+TL+DDPG algorithm computation time compared to the benchmark algorithms in the training process and online application. This table delineates our approach will converge to the optimum solution in the training process with the lowest time consumption compared to other algorithms. However, all algorithms offer solutions to the system in a short fraction of a minute, which is due to the robustness of RL methods in online applications.

Table 3
Comparison of training time and online application of different methods.

Techniques	Training time (hrs)	Inference Time (sec)
DQN	32.32	2
DDQN	31.73	1.8
DDPG	25.06	1.6
RS+TL+DDPG	12.82	1.2

We considered the mixed-integer programming formulation of our island microgrid as a base solution and solved it with the help of CPLEX. Fig. 9 represents the operational cost of a microgrid for three understudy days with the proposed algorithm and a wide range of DRL-based methods. The results show that our proposed algorithm schedules the microgrid with the nearest results to the mixed integer programming-based method. Despite these promising results, there is still room for further progress in handling the instability issue and its effect on the efficiency of exploration in the learning process of DDPG, such as applying stochastic weight averaging and adding dropout to the neural networks as techniques for parameter tuning, which we will undertake in our future approach. While averaging the weights of both actor and critic networks will be done in the successor approach to prevent forgetting the policy with greater rewards, the predecessor solution can empower exploration by applying it to the actor network because of its duty in action selection. However, both approaches are prone to rising system computing costs, which should be noticed while hiring them.

Another open issue in this area is the need to generalize our solution for various environments as a prominent approach in transfer learning. Such generalization is particularly important given that there are 62 primary islands in Korea, with 12 of them currently featuring microgrids. Furthermore, this trend is expected to expand by 2030 due to the government's policy of increasing RES penetration [3]. Hence, it is beneficial to develop an adaptable EMS system that can be tailored to various settings. To accomplish this, we utilized a straightforward approach called reward shaping, which entailed imposing a penalty for extreme actions. We arranged our transfer learning subtasks based on the extreme behaviors related to two indispensable boundaries of each microgrid, including BESS boundaries for SoC and equivalency of power generation and consumption. Thus, we designed our approach to be easily scalable and adaptable to any future extension of the understudy microgrid itself, and it is expected to be adaptable to any other type of microgrid. We will consider our method examination in other Korean islanded microgrids, after method improvement according to the predecessor future work.

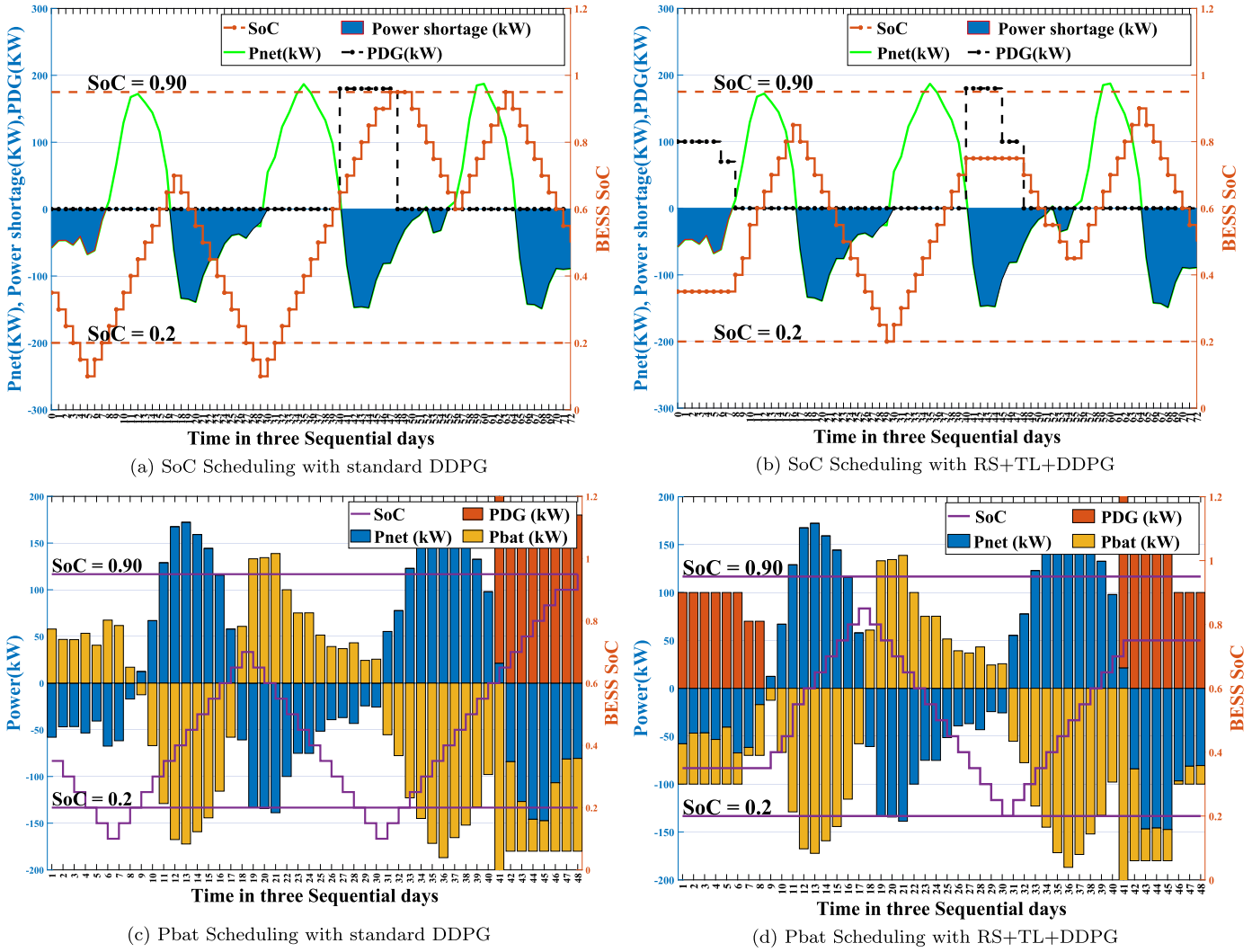

Fig. 8. SoC and Power balance restriction performance of standard DDPG and RS+TL+DDPG comparison.

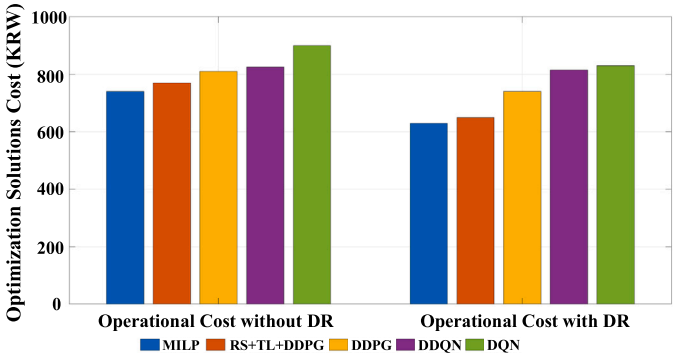

Fig. 9. Operational cost comparison of the proposed method with benchmark algorithms for three sequential under-study days.

4. Conclusions

Microgrids were developed as a solution to the issue of randomness in renewable energy sources. However, this solution requires a microgrid equipped with a well-trained EMS unit. Model-free DRL methods are well suited to continuous, stochastic, and multidimensional microgrid environments. Hence, in this paper, we deployed DDPG as one of the DRL methods demonstrating efficiency in continuous action space and high-dimension environment to solve the EMS problem of the islanded microgrid. By deploying reward shaping and attaching transfer learning to DDPG, we could tackle the long-learning process for DDPG training and achieve accurate scheduling concerning EMS requirements. The results showed that EMS with RS+TL+DDPG trained to supply loads with BESS and DG in RES absence without bothering power consumers with load shedding.

The findings of this research provide insights for generalizing the EMS solution for island microgrids, especially in Korea, where there is a growing demand for such deployments. In our future work, we will investigate the possibility of generalizing our model by evaluating its performance in other island microgrids in Korea. The remaining drawback of the training process of DDPG was slight instability in the learning process, which we consider tackling this issue in future work by applying solutions such as stochastic weight averaging to provide a highly efficient exploration of the environment during the learning process and dropout in neural network parameter selections.

Declaration of competing interest

The authors declare that there is no conflict of interest in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MSIT) (NRF-2021R1F1A1063640).

References

[1] L. Tightiz, H. Yang, A comprehensive review on IoT protocols' features in smart grid communication, Energies 13 (11) (2020) 2762.

[2] IEEE Standard for the Specification of Micro-grid Controllers; IEEE Std 2030.7, IEEE Standards Association, Piscataway, NJ, USA, 2018.

[3] L. Tightiz, J. Yoo, A robust energy management system for Korean green islands project, Sci. Rep. 12 (1) (2022) 22005.

[4] M. Haseeb, S.A.A. Kazmi, M.M. Malik, S. Ali, S.B.A. Bukhari, D.R. Shin, Multi objective based framework for energy management of smart micro-grid, IEEE Access 8 (2020) 220302–220319.

[5] M. Latifi, A. Rastegarnia, A. Khalili, W.M. Bazzi, S. Sanei, A self-governed online energy management and trading for smart micro/nano-grids, IEEE Trans. Ind. Electron. 67 (9) (2020) 7484–7498.

[6] C. Huang, H. Zhang, Y. Song, L. Wang, T. Ahmad, X. Luo, Demand response for industrial micro-grid considering photovoltaic power uncertainty and battery operational cost, IEEE Trans. Smart Grid 12 (4) (2021) 3043–3055.

[7] J. Kim, H. Oh, Jun K. Choi, Learning based cost optimal energy management model for campus microgrid systems, Appl. Energy 311 (2022) 118630.

[8] J.A. Silva, J.C. López, N.B. Arias, M.J. Rider, L.C.P. da Silva, An optimal stochastic energy management system for resilient microgrids, Appl. Energy 300 (2021) 117435.

[9] İ. Yağ, A. Altan, Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments, Biology 11 (12) (2022) 1732.

[10] Y. Ozcelik, A. Altan, Classification of diabetic retinopathy by machine learning algorithm using entropy-based features, in: Cankaya International Congress on Scientific, 2023, pp. 1–11.

[11] E.O. Arwa, K.A. Folly, Reinforcement learning techniques for optimal power control in grid-connected microgrids: a comprehensive review, IEEE Access 8 (2020) 208992–209007.

[12] Y. Khawaja, I. Qiqieh, J. Alzubi, O. Alzubi, A. Allahham, D. Giaouris, Design of cost-based sizing and energy management framework for standalone microgrid using reinforcement learning, Sol. Energy 251 (2023) 249–260.

[13] R.S. Sutton, G.B. Andrew, Reinforcement Learning: An Introduction, MIT Press, 2018.

[14] M. Cao, Z. Yin, Y. Wang, L. Yu, P. Shi, Z. Cai, A reliable energy trading strategy in intelligent microgrids using deep reinforcement learning, Comput. Electr. Eng. 110 (2023) 108796.

[15] Daniel J.B. Harrold, Jun Cao, Zhong Fan, Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning, Energy 238 (2022) 121958.

[16] Y. Ji, J. Wang, et al., Real-time energy management of a microgrid using deep reinforcement learning, Energies 12 (12) (2019) 2291.

[17] V. Bui, A. Hussain, H. Kim, Double deep q -learning-based distributed operation of battery energy storage system considering uncertainties, IEEE Trans. Smart Grid 11 (1) (2020) 457–469.

[18] G. Gao, J. Li, Y. Wen, DeepComfort: energy-efficient thermal comfort control in buildings via reinforcement learning, IEEE Int. Things J. 7 (9) (2020) 8472–8484.

[19] L. Tightiz, H. Yang, Resilience microgrid as power system integrity protection scheme element with reinforcement learning based management, IEEE Access 9 (2021) 83963–83975.

[20] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, K. Zheng, Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids, IEEE Int. Things J. 8 (10) (2021) 7938–7953.

[21] Z. Yan, Y. Xu, Real-time optimal power flow: a Lagrangian based deep reinforcement learning approach, IEEE Trans. Power Syst. 35 (4) (2020) 3270–3273.

[22] S. Zhang, R. Jia, H. Pan, Y. Cao, A safe reinforcement learning-based charging strategy for electric vehicles in residential microgrid, Appl. Energy 348 (2023) 121490.

[23] L. Fan, J. Zhang, Y. Liu, T. Hu, H. Zhang, Optimal scheduling of microgrid based on deep deterministic policy gradient and transfer learning, Energies 14 (3) (2021) 584.

[24] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint, arXiv:1509.0297, 2015.

[25] Open energy system databases § renewables.ninja, [Online]. Available: http://www.renewables.ninja.

[26] Y.K. Seo, W.H. Hong, Constructing electricity load profile and formulating load pattern for urban apartment in Korea, Energy Build. 78 (2014) 222–230.

[27] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, K. Zheng, Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids, IEEE Int. Things J. 8 (10) (2021) 7938–7953.