



## Article

# Environment-Friendly Power Scheduling Based on Deep Contextual Reinforcement Learning

Awol Seid Ebrie <sup>1</sup>, Chunhyun Paik <sup>2</sup> , Yongjoo Chung <sup>3</sup> and Young Jin Kim <sup>4,\*</sup> 

<sup>1</sup> Major in Industrial Data Science and Engineering, Department of Industrial and Data Engineering, Pukyong National University, Busan 48513, Republic of Korea; es.awol@pukyong.ac.kr

<sup>2</sup> Department of Industrial Management and Big Data Engineering, Dongeui University, Busan 47340, Republic of Korea; chpaik@deu.ac.kr

<sup>3</sup> Department of Global Marketing, Busan University of Foreign Studies, Busan 46234, Republic of Korea; chungyj@bufs.ac.kr

<sup>4</sup> Department of Systems Management and Engineering, Pukyong National University, Busan 48513, Republic of Korea

\* Correspondence: youngk@pknu.ac.kr; Tel.: +82-51-629-6486

**Abstract:** A novel approach to power scheduling is introduced, focusing on minimizing both economic and environmental impacts. This method utilizes deep contextual reinforcement learning (RL) within an agent-based simulation environment. Each generating unit is treated as an independent, heterogeneous agent, and the scheduling dynamics are formulated as Markov decision processes (MDPs). The MDPs are then used to train a deep RL model to determine optimal power schedules. The performance of this approach is evaluated across various power systems, including both small-scale and large-scale systems with up to 100 units. The results demonstrate that the proposed method exhibits superior performance and scalability in handling power systems with a larger number of units.

**Keywords:** power scheduling; unit commitment; reinforcement learning; agent-based simulation



**Citation:** Ebrie, A.S.; Paik, C.; Chung, Y.; Kim, Y.J. Environment-Friendly Power Scheduling Based on Deep Contextual Reinforcement Learning. *Energies* **2023**, *16*, 5920. <https://doi.org/10.3390/en16165920>

Academic Editors: Marialaura Di Somma, Jianxiao Wang and Bing Yan

Received: 16 July 2023

Revised: 6 August 2023

Accepted: 8 August 2023

Published: 10 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Electrical energy generated from fossil fuels emits significant amounts of greenhouse gases (GHGs), including carbon dioxide (CO<sub>2</sub>), sulfur dioxide (SO<sub>2</sub>), and nitrous oxide (N<sub>2</sub>O). These emissions have detrimental effects on human health and contribute to climate change and global warming. However, the prevailing focus on economic concerns in power generation often results in higher emission levels since economic costs and environmental impacts tend to be inversely related. This effect has been amplified since the implementation of the global emissions trading scheme (ETS) in 2005, aimed at controlling GHGs [1,2]. As a result, power generation based solely on economic costs leads to increased financial penalties for emissions, along with adverse environmental impacts. Therefore, the traditional approach of scheduling power generation based solely on economic costs, which overlooks environmental ETS, is no longer acceptable [2]. Consequently, it becomes imperative to determine an efficient and environmentally friendly power generation schedule that might have lower emission costs, indicating that it generates fewer GHGs or other pollutants during its operation. The primary objective of environmentally friendly power scheduling is to achieve a sustainable cost and emission balance, where economic concerns are considered without compromising environmental sustainability. This means effectively meeting electricity demand while minimizing the environmental impact, especially in terms of greenhouse gas emissions (GHGs) and other pollutants. By adopting this approach, not only does the profitability of power generation increase, but it also leads to reduced emission levels through efficient management and scheduling of generating units [1,2].

Prior research has explored various model-based approaches, such as conventional and dynamic programming and stochastic optimizations, to address the challenges in power

scheduling [3]. However, these methods often face the curse of the dimensionality problem, which leads to the use of heuristic rules and simplifications that may not effectively handle real-sized problems [3–5]. As power systems continue to grow in size and complexity, even small improvements in efficiency achieved through enhanced power scheduling methods can yield significant economic and environmental benefits [4].

Recently, artificial intelligence (AI) has shown promise in learning optimal strategies without prior knowledge. Particularly, reinforcement learning (RL) can achieve this by employing self-play learning and adapting decision-making policies over time based on feedback from dynamic environments [4,6]. Furthermore, RL does not rely on precise mathematical models [4], making it more suitable for real-world scenarios where power generation dynamics may be uncertain or challenging to accurately model.

However, despite the potential of RL-based models to offer improved power scheduling solutions, only a few studies (such as [4–10]) are available in the literature. These RL-based models tend to prioritize economic costs and overlook environmental impacts, leading to excessive carbon emissions and neglecting long-term consequences [6]. Additionally, scalability remains a challenge for both the model- and RL-based approaches, as the dimensionality grows exponentially with an increasing number of units [11]. Consequently, simplified power scheduling definitions are often used, which may not adequately represent realistic power systems.

This study aims to address these limitations by proposing a novel deep RL-based method for power scheduling that minimizes both economic and environmental costs. The algorithm utilizes an agent-based contextual simulation environment, where generating units are represented as agents. The simulation environment automatically corrects illegitimate commitments and adjusts supply capacity to meet demand, enabling agents to learn optimal behaviors more efficiently. Furthermore, the proposed method mitigates the dimensionality problem associated with large-scale problems, distinguishing it from existing approaches in the literature. The power scheduling dynamics are simulated using a Markov decision process (MDP), and the results are fed into a deep Q-network (DQN) with separate output nodes (ON and OFF) for each agent, allowing for effective decision making. The remainder of this article is organized as follows: The description of the power scheduling problem is presented in Section 2. Then, Section 3 provides the technical details of the proposed methodology, which is demonstrated with a numerical example in Section 4. Concluding remarks follow in the Section 5.

## 2. Problem Description

Given a power system with  $n$  generating units and a power scheduling horizon of 24 h, let  $z_{it} \in \{0, 1\}$  denote the commitment (ON/OFF) status of unit  $i$  at period  $t$ , and  $p_{it} \in [0, \infty)$  be the optimal power output of unit  $i$  at period  $t$  (MW).

### 2.1. Objective Function

The operating cost of power production at each period  $t$  is often defined by the sum of the production costs ( $c_{it}^{prod}$ ), start-up costs ( $c_{it}^{ON}$ ), and shutdown costs ( $c_i^{OFF}$ ) of all units. This definition of total operation cost ignores environmental constraints, which depend on local regulation and emission allowance trading market schemes [12]. A properly represented power scheduling model should also include other costs that are not linked to fuel prices but related to fuel consumption and technological efficiency [3]. Hence, it is necessary to include emission costs ( $c_{it}^{emis}$ ) as part of the total operation cost [13]. In the existing models, the environmental impact of power generation is primarily addressed in the form of emissions constraints and penalties. Emission constraints involve setting limits on the maximum allowable GHGs and other pollutants, whereas the penalty method assigns penalty costs to units emitting beyond the allowed limits. However, a common issue with both the emission constraint and penalty methods is their lack of flexibility, as they do not account for dynamic adjustments based on real-time changes, such as demand fluctuations, outages of units, and fuel and other operational expenses. Both methods

tend to prioritize compliance with emission limits rather than actively pursuing emission reduction strategies. To address these limitations, this study proposes the adoption of emission cost parameters integrated into the main objective function. By representing the emissions produced per MW for different types of units, this approach offers a continuous and gradual representation of environmental impacts [13]. This not only allows for more nuanced and flexible decision making by treating emissions as a continuous variable rather than a binary constraint, but it also enables the assessment of the emission-reduction potential of different power plants.

Thus, the objective function representing the total operational economic and environmental costs of the entire planning horizon ( $\mathcal{C}$ ) can be expressed as Equation (1):

$$\mathcal{C} = \sum_{t=1}^{24} \sum_{i=1}^n \left\{ z_{it} (c_{it}^{prod} + c_{it}^{emis}) + z_{it}(1 - z_{i,t-1})c_{it}^{ON} + (1 - z_{it})z_{i,t-1}c_i^{OFF} \right\} \quad (1)$$

where

$$c_{it}^{prod} = \alpha_i p_{it}^2 + \beta_i p_{it} + \delta_i \quad (2)$$

$$c_{it}^{emis} = p_{it} \sum_{h=1}^m \phi_h \psi_{ih} \quad (3)$$

$$c_{it}^{ON} = \begin{cases} c_i^{hot,ON}, & t_{i*}^{down} \leq t_{i,t-1}^{OFF} \leq t_{i*}^{down} + t_{i*}^{cold} \\ c_i^{cold,ON}, & t_{i,t-1}^{OFF} > t_{i*}^{down} + t_{i*}^{cold} \end{cases} \quad (4)$$

Equation (2) is the production cost function of unit  $i$  at period  $t$  where  $\alpha_i$ ,  $\beta_i$ , and  $\delta_i$  are the corresponding quadratic, linear, and constant coefficients, respectively. Next, the emission cost of  $m$  types of pollutants released by unit  $i$  is represented in Equation (3). Since three emission types (namely, CO<sub>2</sub>, SO<sub>2</sub>, and NO<sub>x</sub>) have been considered,  $m = 3$  in this study. The emissions level is often directly related to the fuel consumption and technological efficiency [3]. As a result, the cost of emissions can be expressed as a linear function of the power outputs [13], where  $\phi_h$  is the external cost of emission type  $h$  (\$/g), and  $\psi_{ih}$  is the  $h^{\text{th}}$  emission factor of unit  $i$  (g/MW). Finally, the start-up cost given in Equation (2) is a function of the time duration for which unit  $i$  has been continuously offline (OFF) until the period  $t$ ,  $t_{i,t-1}^{OFF}$ . The shutdown costs are fixed but usually negligible [14] and mostly considered zero [3].

## 2.2. Constraints

The objective function in Equation (1) is required to be minimized subject to different unit-specific and system level constraints as presented in Equations (5)–(9):

$$\text{Capacities : } z_{it} p_{i*}^{min} \leq p_{it} \leq z_{it} p_{i*}^{max} \quad (5)$$

$$\text{Ramp rates : } z_{i,t-1} z_{it} (p_{i,t-1} - p_{i*}^{down}) \leq p_{it} \leq z_{i,t-1} z_{it} (p_{i,t-1} + p_{i*}^{up}) \quad (6)$$

$$\text{Operating times : } t_{it}^{ON} \geq t_{i*}^{up} \text{ and } t_{it}^{OFF} \geq t_{i*}^{down} \quad (7)$$

$$\text{Load Balance : } \sum_{i=1}^n z_{it} p_{it} = d_t \quad (8)$$

$$\text{Reserve : } \sum_{i=1}^n z_{it} p_{it}^{max} \geq (1 + r) d_t \quad (9)$$

The unit constraints in Equations (5)–(7) affect each unit taken separately, which collectively accounts for the technical specifications of generating units, and the system constraints in Equations (8) and (9) are used to balance the power supply and demand during each period. Due to the non-convexity of the objective function, the combinatorial nature of commitments, and the time-dependent technical characteristics of the power supply units, solving the objective function in Equation (1) subject to the constraints in

Equations (5)–(9) using classical methods is highly computationally demanding even for a moderate number of units and might also stack in some local optima. As a result, the power scheduling problem remains a strongly NP-hard problem, causing the curse of dimensionality, and the implicit burden of computation has limited the scope of numerical optimization [3]. A model-free RL approach may provide a promising methodological framework for solving the power scheduling problem [5].

### 3. Proposed Methodological Framework

A novel multi-agent deep contextual RL algorithm for power scheduling is proposed by constructing a specialized environment called an “agent-based contextual simulation environment”, whose main peculiarities are explained in Section 3.1. Within the environment, the generating units are represented as cooperative types of RL agents [15]. The agents are active in observing the contextual changes in the environment, and they can make independent decisions regarding their commitment status and optimal power outputs. On the other hand, the agents collaborate to satisfy demand, including the reserve availability at each hour (timestep) described in Equations (8) and (9), and the total operation cost of the entire planning horizon (episode) is to be minimized. The power scheduling dynamics from the agent-to-environment interactions follow an agent-based simulation strategy [16] because the agents are heterogeneous, active, and autonomous. These dynamics are simulated in the form of a Markov decision process (MDP), whose key elements are defined below, and then fed as inputs for the deep RL model.

#### 3.1. The Power Scheduling Dynamics as an MDP

Since the planning horizon is an hourly divided day, each hour is considered a timestep  $t, \forall t$ . For each timestep  $t$  of an episode, the system will be state  $s_t$  consisting of different components: current timestep  $t$ , minimum capacities ( $p_{it}^{min}; \forall i$ ), and maximum capacities ( $p_{it}^{max}; \forall i$ ) based on the maximum ramp-down ( $p_{i*}^{down}; \forall i$ ) and ramp-up ( $p_{i*}^{up}; \forall i$ ) rates of the units, current operating (online/offline) time durations ( $t_{it}; \forall i$ ) of the units based on the minimum up-time duration ( $t_{i*}^{up}; \forall i$ ) and down-time duration ( $t_{i*}^{down}; \forall i$ ) of the units, and the demand ( $d_t$ ) to be satisfied. The state  $s_t$  at timestep  $t$  is defined as  $s_t = (t, p_t^{min}, p_t^{max}, t_t, d_t)$  where  $t$  is the current timestep,  $p_t^{min}$  is a vector of minimum capacities,  $p_t^{max}$  is a vector of maximum capacities,  $t_t$  is a vector of current (online/offline) time duration, and  $d_t$  is the demand. Overall, the system's state space for the entire episode can be described as  $\mathcal{S} = (t, P^{min}, P^{max}, T, d)$ . There are two possible actions (switch-to/stay ON or switch-to/stay OFF) for each of the  $n$  agents. This implies, there are a total of  $2^n$  action combinations (i.e., unit commitments) in the action space  $\mathcal{A}$ . Thus, each agent  $i$  will decide their optimal action ( $a_{it} \in \{0, 1\}$ ) at timestep  $t$  (or in state  $s_t$ ). Then, the decisions of all the  $n$  agents together constitute an  $n$ -dimensional vector  $a_t = \{0, 1\}^n \in \mathcal{A}$ . This actions vector  $a_t$  is the change of the switch (ON/OFF) status  $z_t$  of units at period  $t$  to  $z_{t+1}$  at the next period  $t + 1$ . Once the agents take actions  $a_t \in \mathcal{A}$  in the current state  $s_t \in \mathcal{S}$ , there is a transition (or probability) function  $\mathcal{P}(s_{t+1}|s_t, a_t)$  that leads to the next state  $s_{t+1}$ . The transition function must satisfy all the constraints.

At each timestep  $t$  of the planning horizon, the agents first observe the state  $s_t = (t, p_t^{min}, p_t^{max}, t_t, z_t, d_t) \in \mathcal{S}$ . Then, each agent can decide to be either ON or OFF, which would result in  $2^n$  combinations of unit commitment found in an action space  $\mathcal{A}$ . The decisions of all agents constitute the action vector  $a_t = \{0, 1\}^n \in \mathcal{A}$ . Then, the agents get an aggregate reward  $r_t \in \mathbb{R}$  that can lead to the next state  $s_{t+1} \in \mathcal{S}$  through a transition (probability) function  $\mathcal{P}(s_{t+1} = s' | s_t = s, a_t)$ . Therefore, these power scheduling dynamics can be represented as a 4-tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$  Markovian decision process where  $\mathcal{S}$  is a state space,  $\mathcal{A}$  is an action space,  $\mathcal{P}$  is a transition (or probability) function, and  $r$  is a reward.

### 3.2. Agent-Based Contextual Simulation Environment Algorithm

The structure of agent-based simulation environment is roughly similar to an OpenAI Gym as described below.

**Step 1.** The parameters of supply units are initialized as  $\mathcal{S}_0 = (0, p_0^{min}, p_0^{max}, \tau_0, z_0, d_0)$ .

**Step 2.** The minimum and maximum marginal costs of all agents are determined based on the average production cost, Equation (10).

$$\lambda_i^{min} = \alpha_i p_{i*}^{max} + \beta_i + \frac{\delta_i}{p_{i*}^{max}} \text{ and } \lambda_i^{max} = \alpha_i p_{i*}^{min} + \beta_i + \frac{\delta_i}{p_{i*}^{min}}; \forall i \quad (10)$$

**Step 3.** The must-ON ( $u_{it}^1$ ) or must-OFF ( $u_{it}^0$ ) agents are identified based on the operating times, Equation (11).

$$u_{it}^1 = 1 \text{ if } 0 < \tau_{it} < \tau_{i*}^{up}; \text{ and } u_{it}^0 = 1 \text{ if } -\tau_{i*}^{down} < \tau_{it} < 0; \forall i. \quad (11)$$

**Step 4.** The agents execute their action  $a_t$  in state  $\mathcal{S}_t$ , and then pass to the next state  $\mathcal{S}_{t+1}$  through a transition (or probability) function,  $\mathcal{P}(\mathcal{S}_{t+1} | \mathcal{S}_t, a_t)$ , satisfying all constraints.

**Step 4.1.** The legality of action  $a_{it} \in a_t$  of each agent is confirmed and legalized if there are any violations of the constraints specified in Equation (11), as shown in Equation (12).

$$a_{it} = 1 \text{ if } a_{it} = 0 \mid u_{it}^1 = 1; \text{ and } a_{it} = 0 \text{ if } a_{it} = 1 \mid u_{it}^0 = 0, \forall i. \quad (12)$$

**Step 4.2.** The aggregate supply capacity of agents is checked for sufficiency in satisfying the demand and future demands when OFF units have not completed their downtime. Then contextual capacity adjustments are made if necessary, and if possible.

- If  $\sum_{i=1}^n a_{it} p_{it}^{max} < (1 + \epsilon) d_t$ , then set each  $a_{it} = 1 \mid u_{it}^1 = 0$  based on the increasing order of  $\lambda_i^{min}$ 's of Equation (10) until  $\sum_{i=1}^n a_{it} p_{it}^{max} \geq (1 + \epsilon) d_t$ . If the capacity shortage is not fully corrected due to unconstrained OFF units, then  $\mathcal{S}_t$  is labeled as a terminal state ( $\mathcal{S}_t^+$ ) that would result an incomplete episode ( $\mathbb{I}_{[\mathcal{S}_t^+]} = 1$ ).
- If  $\sum_{i=1}^n a_{it} p_{it}^{min} > (1 + \epsilon) d_t$ , then set each  $a_{it} = 0 \mid u_{it}^0 = 1$  as per the decreasing order of  $\lambda_i^{min}$ 's of Equation (10) until  $\sum_{i=1}^n a_{it} p_{it}^{min} \leq (1 + \epsilon) d_t$ . If the excess capacity is not yet fully adjusted due to an insufficient number of unconstrained ON units, it results in an incomplete episode ( $\mathbb{I}_{[\mathcal{S}_t^+]} = 1$ ) as the state  $\mathcal{S}_t$  is terminal ( $\mathcal{S}_t^+$ ).
- If the current capacity does not satisfy future demands, set each  $a_{it} = 1 \mid (\tau_{it} \geq \tau_{i*}^{up})$  as per the decreasing order of  $\lambda_i^{min}$ 's of Equation (10). The current state  $\mathcal{S}_t$  is also labeled as terminal ( $\mathcal{S}_t^+$ ) if the future demands cannot be meet while the offline units must still be OFF due to an insufficient number of unconstrained OFF units.

**Step 5.** The total operation cost at timestep  $t$  is determined. First, start-up and shutdown costs are obtained based on the action  $a_t$ . Second, a lambda iteration algorithm is used for solving the optimal power loads  $p_{it}$ , in Equation (2), which are then used to estimate the emission costs specified in Equation (3). Lastly, the total operation cost is obtained using Equation (13) where  $z_{i,t+1} = a_{it}, \forall i$ .

$$\mathcal{C}_t = \sum_{i=1}^n \left\{ z_{i,t+1} (c_{it}^{prod} + c_{it}^{emis}) + z_{i,t+1} (1 - z_{it}) c_{it}^{ON} + (1 - z_{i,t+1}) z_{it} c_{it}^{OFF} \right\} \quad (13)$$

**Step 6.** The agents get an aggregate reward according to the predefined function given in Equation (14), which is the negative of the normalized total operation cost scaled to 100.

$$r_t = \mathcal{R}(\mathcal{S}_t, a_t, \mathcal{S}_{t+1}) = \left( 1 - \frac{(1 - \mathbb{I}_{[\mathcal{S}_t^+]}) \mathcal{C}_t + \mathbb{I}_{[\mathcal{S}_t^+]} \mathcal{C}_t^+ - \mathcal{C}^{min}}{\mathcal{C}^{max} - \mathcal{C}^{min}} \right) \times 100 \quad (14)$$

In the episodic task of RL, incomplete episodes need to be avoided, and large penalties are recommended by [11]. For this purpose, while the cost function in Equation (13) is used for non-terminal states, the cost for terminal states is defined as  $\mathcal{C}_t^+ = \mathcal{C}^{max} - \frac{t}{23} (\mathcal{C}^{max} - \mathcal{C}^{p^{max}})$  where  $\mathcal{C}^{max} = \sum_{i=1}^n \lambda_i^{max} p_{i*}^{max}$  and  $\mathcal{C}^{p^{max}}$  is the sum of Equations (2) and (3),



assuming  $p_{it} = p_{i*}^{max}$ . This provides evenly distributed penalties, maintaining the desired proximity to the final timestep.

**Step 7.** If the current state is terminal (i.e.,  $\mathbb{I}_{[s_t^+]}$  = 1) or  $t \geq 24$ , then go to Step 1 to re-initialize the environment and restart a new episode to state  $s_0$ . But, if  $t < 24$  and  $\mathbb{I}_{[s_t^+]} = 0$ , the agents pass to the next state  $s_{t+1} = (t + 1, p_{t+1}^{min}, p_{t+1}^{max}, \ell_{t+1}, z_{t+1}, d_{t+1})$  where  $z_{t+1} = a_t$  is a vector commitment status;  $p_{t+1}^{min} = \max\{p_{*}^{min}, z_t' z_{t+1}(p_t - p_{*}^{down})\}$  is a vector of minimum capacities;  $p_{t+1}^{max} = \min\{p_{*}^{max}, z_t' z_{t+1}(p_t + p_{*}^{up}) + \mathbb{I}_{[z_t' z_{t+1}=0]} p_{*}^{max}\}$  is a vector of maximum capacities;  $\ell_{t+1} = (\ell_t + 1 \text{ if } z_{t+1} = 1 | \ell_t > 0, -1 \text{ else if } z_{t+1} = 0 | \ell_t > 0, 1 \text{ else if } z_{t+1} = 1 | \ell_t < 0, \ell_t - 1 \text{ if } z_{t+1} = 0 | \ell_t < 0)$  is a vector of operating time durations.

**Step 8.** Update the must-ON and must-OFF agents for the next timestep  $t + 1$  as defined in Equation (11):  $u_{i,t+1}^1 = 1$  if  $0 < \ell_{i,t+1} < \ell_{i*}^{up}$ , and  $u_{i,t+1}^0 = 1$  if  $-\ell_{i*}^{down} < \ell_{i,t+1} < 0; \forall i$ .

**Step 9.** The execution of  $a_t$  in the environment returns the next state ( $s_{t+1}$ ), the reward ( $r_t$ ), indicates whether the state is terminal ( $\mathbb{I}_{[s_t^+]}$ ) and loads dispatch information together with the legally confirmed action ( $a_t, p_t$ ).

It should be noted that action  $a_t$  returned in Step 9 is not necessarily the same as the action  $a_t$  executed in Step 4 since the environment makes contextual corrections in Step 4.1 and 4.2 using the idea of a contextual search [17]. This agent-based contextual simulation environment, one of the main contributions of this study, can be highly effective in reducing the computing and training time of the multi-agent deep contextual RL model described below.

### 3.3. Deep Contextual Reinforcement Learning

At each timestep  $t \in \mathcal{T}$ , the agents observe a state  $s_t$  from  $\mathcal{S}$  and select their respective actions  $a_t$  from the action space  $\mathcal{A}$  according to a policy  $\pi(a_t | s_t)$ , where  $\pi$  is a mapping from states  $s_t$  to actions  $a_t$ . The agents then receive a reward  $r_t$  and proceed to the next state  $s_{t+1}$ . This process continues until the agents finish the entire episode or reach a terminal state, both of which reinitialize the environment. The agents' goal is to learn a policy  $\pi(a_t | s_t)$  that maximizes the long-run cumulative sum of rewards called return, defined as  $G_t \doteq \sum_{k=0}^{\infty} \gamma^k r_{t+k}$  where  $\gamma$  is a discount rate  $\gamma \in [0, 1]$  of the MDP. The expected return of action,  $a_t$  in the state  $s_t$  can be expressed as an action-value function  $Q^{\pi}(s_t, a_t) = \mathbb{E}(G_t | s_t, a_t)$ . It can be approximated using a deep Q-network (DQN) which can be applied in a high-dimensional state and/or action space [18]. The action-value function can now be written as  $Q(s_t, a_t | \theta)$ , where  $\theta$  consists of the parameters of the DQN model whose inputs are the power scheduling dynamics simulated from the environment in the form of MDPs. The size of action space  $\mathcal{A}$  is  $2^n$ , which may render an exponential growth in computation. Thus, it is impractical to parameterize the model into  $2^n$  output nodes. Instead, it is parameterized into  $2n$  output nodes corresponding to the two possible actions (ON/OFF) of each agent. As a result, the model estimates action-values for  $2n$  output nodes, and then the decisions of agents made using an exponential decay epsilon-greedy exploration strategy collectively constitute the action vector  $a_t$ .

In a power scheduling problem, the action in a particular state affects the rewards and a set of future states, which yields serial correlations among the MDPs. In such cases, direct application of DQN may not be efficient, as it might result in unstable and slow learning processes [11]. Employing the notion of experience replay, the autocorrelation among the states may be properly addressed, and the training process can be expedited and stabilized [19]. After storing a transition tuple  $(s_t, a_t, r_t, s_{t+1})$  to a replay buffer  $\mathbb{B}$ , a batch  $b$  of experiences  $(s, a, r, s')$  is sampled to approximate the action-value function  $Q(s, a | \theta)$  and the target network  $Q(s', a' | \theta')$ , where  $s$  and  $s'$  are of size  $(b \times 2n)$ , and the sizes of  $a$  and  $r$  are  $(b \times n)$  and  $(b \times 1)$ , respectively. The DQN algorithm minimizes the mean-squared loss (i.e., temporal difference)  $L_e(\theta)$  defined by

$$L_e(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathbb{B}} \left[ r + \gamma \max_{a'} Q(s', a' | \theta') - Q(s, a | \theta) \right]^2 \quad (15)$$

where  $\theta'$  is the parameter of target network, which is periodically updated from the Q-network parameters  $\theta$ , and  $e$  denotes the iteration index.

#### 4. Demonstrative Example

The applicability of the proposed deep contextual RL method is demonstrated with the power system investigated in [20]. The test system consists of five units, for which supply and demand profiles and emission parameters can be found in [20]. The deep RL utilizes a feedforward neural network featuring a rectified linear unit (ReLU) activation function on both the hidden and output layers. The learning rate and discount factor were set to 0.0001 and 0.99, respectively, and the Adam's optimizer was used. Employing the popular genetic algorithm, the optimal cost in [20] was \$430,331, summing up the start-up, production, and emission costs of \$3140 (0.7%), \$289,178 (67.2%), and \$138,010 (32.1%), respectively. On the other hand, the proposed method in this study yields an improved optimal operation cost of \$413,122 as shown in Table 1 and Figures 1 and 2, corresponding to \$16,808 (4.0%) lower daily operation cost than the results reported in [20] as compared in Table 2 and Figure 3. The total cost is composed of \$2230 (0.5%) for start-up costs, \$275,962 (66.8%) for production costs, and \$134,931 (32.6%) for emission costs. It is also asserted that the proposed method may be computationally efficient by adopting the experience replay. The scalability of the proposed method may thus be tested with a large-scale power system comprising a large number of generating units. Duplicating the five-unit test system multiple times and scaling the demands proportionately, the proposed method has been applied to obtain the optimal power scheduling scheme of individual units. The optimal costs of duplicated large-scale power systems are summarized in Table 3. It is worth noting that the optimal operating cost of each test system is lower than the scaled optimal operating cost of the original five-unit system. It is implied that the proposed method may easily be extended to render an economically and environmentally better solution for larger-scale power systems.

**Table 1.** Optimal commitments, optimal loads, and available reserve of test power system I using the proposed method.

Hour ( $t$ )	Optimal Commitments					Optimal Loads (MW)					$r'$ (%)
	$z_{1t}$	$z_{2t}$	$z_{3t}$	$z_{4t}$	$z_{5t}$	$p_{1t}$	$p_{2t}$	$p_{3t}$	$p_{4t}$	$p_{5t}$	
1	1	0	0	0	0	400.0	0	0	0	0	13.8
2	1	0	1	0	0	426.5	0	23.5	0	0	30.0
3	1	0	1	0	0	450.9	0	29.1	0	0	21.9
4	1	0	1	0	0	455.0	0	45.0	0	0	17.0
5	1	0	1	0	0	455.0	0	75.0	0	0	10.4
6	1	1	1	0	0	455.0	36.6	58.4	0	0	30.0
7	1	1	1	0	0	455.0	52.0	73.0	0	0	23.3
8	1	1	1	0	0	455.0	62.3	82.7	0	0	19.2
9	1	1	1	0	0	455.0	72.6	92.4	0	0	15.3
10	1	1	1	1	0	455.0	77.7	97.3	20	0	22.3
11	1	1	1	1	0	455.0	93.1	111.9	20	0	16.9
12	1	1	1	1	0	455.0	103.3	121.7	20	0	13.6
13	1	1	1	1	0	455.0	77.7	97.3	20	0	22.3
14	1	1	1	0	0	455.0	72.5	92.5	0	0	15.3
15	1	1	1	0	0	455.0	62.3	82.7	0	0	19.2
16	1	1	1	0	0	455.0	36.6	58.4	0	0	30.0
17	1	0	1	0	0	455.0	0	45.0	0	0	17.0
18	1	0	1	1	0	455.0	0	75.0	20	0	20.9
19	1	0	1	1	0	455.0	0	125.0	20	0	10.8
20	1	0	1	1	1	455.0	0	130.0	55	10	10.8
21	1	0	1	1	0	455.0	0	125.0	20	0	10.8
22	1	0	1	1	0	455.0	0	75.0	20	0	20.9
23	1	0	1	0	0	455.0	0	45.0	0	0	17.0
24	1	0	1	0	0	426.4	0	23.6	0	0	30.0

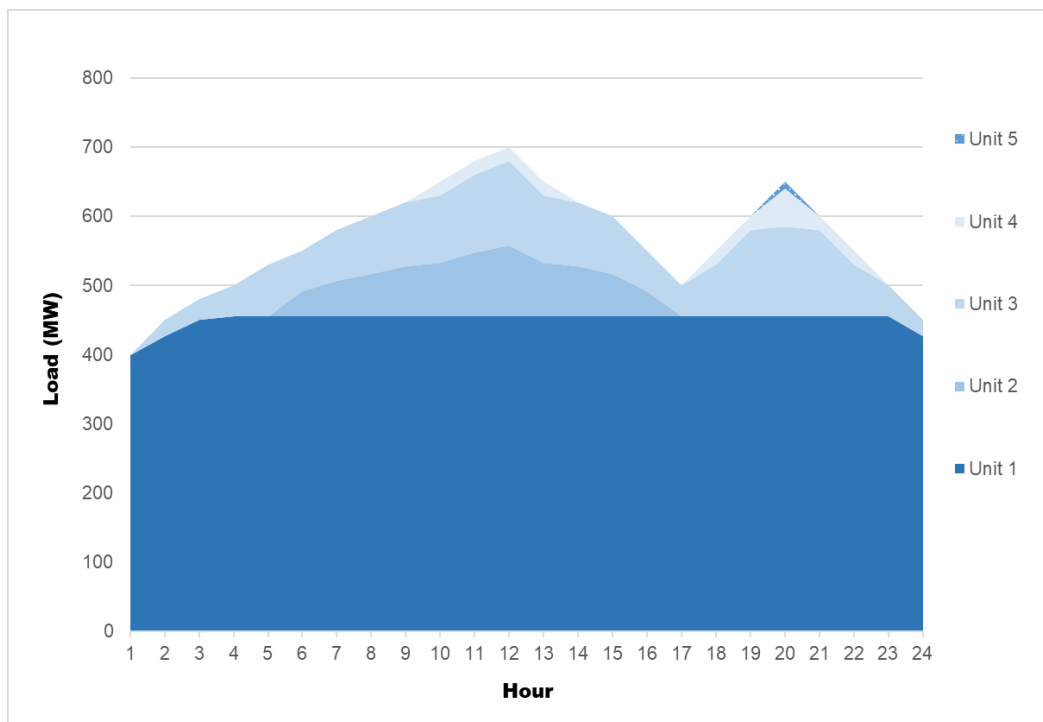


Figure 1. Optimal loads of test power system I using the proposed method.

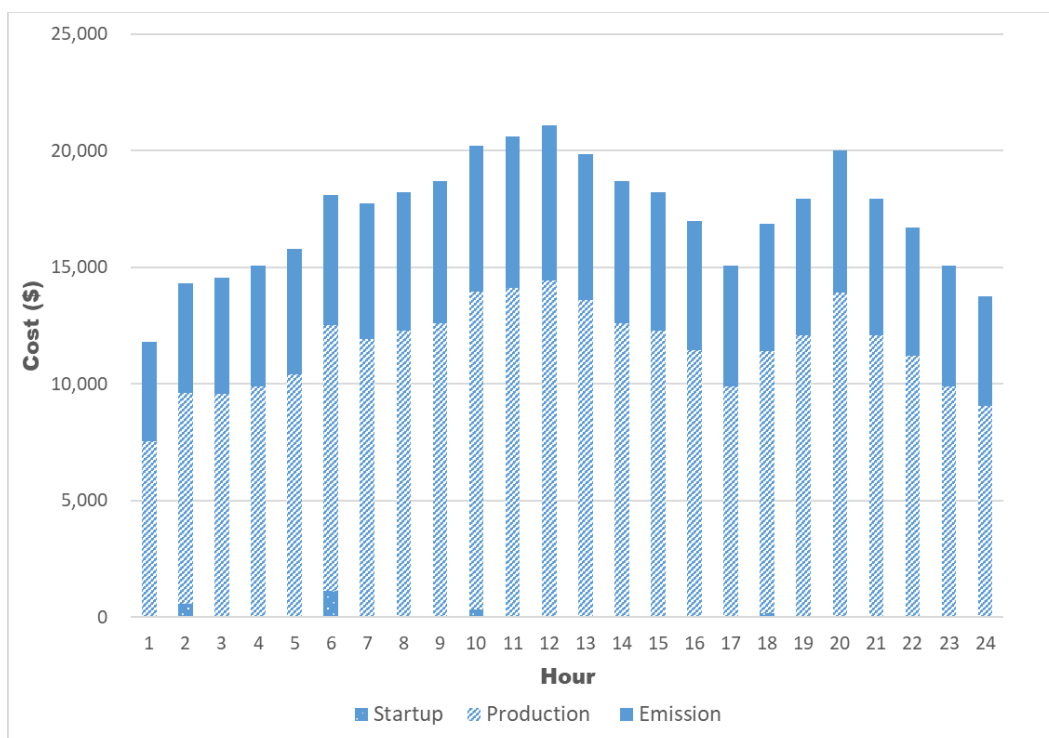
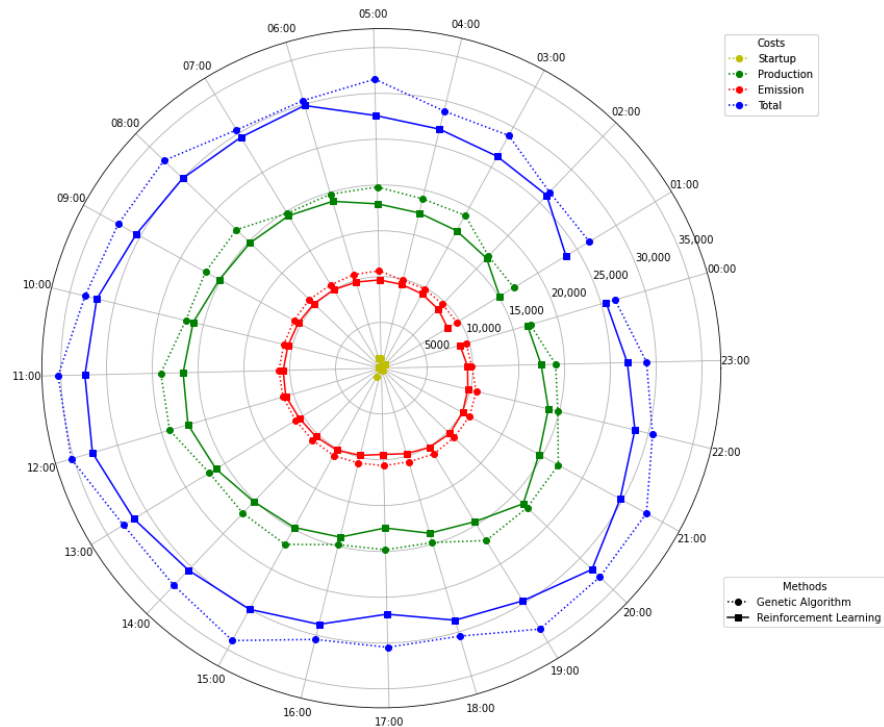


Figure 2. Costs of test power system I using the proposed method.



**Table 2.** Comparison of the optimal costs (start-up cost, production cost, emission cost, and total cost) of test power system I between genetic algorithm (GA) [20] and the proposed RL method.

Hour (t)	Genetic Algorithm [20]				Proposed RL			
	Start-Up	Production	Emission	Total	Start-Up	Production	Emission	Total
1	0	8466	4824	13,290	0	7553	4241	11,793
2	0	8466	4828	13,294	560	9061	4702	14,324
3	60	10,564	5044	15,668	0	9560	5004	14,564
4	0	10,564	5044	15,608	0	9893	5170	15,063
5	1120	11,327	5825	18,271	0	10,395	5401	15,797
6	0	11,327	5825	17,151	1100	11,427	5555	18,082
7	0	11,327	5825	17,151	0	11,930	5786	17,717
8	60	13,425	6045	19,529	0	12,267	5940	18,207
9	0	13,425	6045	19,469	0	12,604	6094	18,699
10	340	13,523	6145	20,008	340	13,591	6251	20,183
11	30	15,621	6365	22,016	0	14,099	6483	20,582
12	0	15,621	6365	21,986	0	14,439	6637	21,076
13	0	13,523	6145	19,668	0	13,591	6251	19,843
14	30	13,425	6045	19,499	0	12,604	6094	18,699
15	1100	13,456	6045	20,601	0	12,267	5940	18,207
16	0	11,358	5825	17,182	0	11,427	5555	16,982
17	0	11,358	5825	17,182	0	9893	5170	15,063
18	0	11,358	5825	17,182	170	11,213	5481	16,865
19	340	13,554	6145	20,039	0	12,059	5866	17,926
20	0	13,554	6145	19,699	60	13,862	6085	20,007
21	0	13,554	6145	19,699	0	12,059	5866	17,926
22	0	11,358	5825	17,182	0	11,213	5481	16,695
23	60	10,564	5044	15,668	0	9893	5170	15,063
24	0	8466	4824	13,290	0	9061	4702	13,764
Total	3140	289,178	138,013	430,331	2230	275,962	134,931	413,122



**Figure 3.** Radar plot of the hourly optimal costs using the genetic algorithm (GA) and the proposed reinforcement learning (RL).

**Table 3.** Optimal costs of power production for large-scale systems.

Number of Units	Cost (\$)			
	Start-Up	Production	Emission	Total
10	4840	545,837	270,724	821,401
20	9300	1,083,979	540,476	1,633,754
30	14,370	1,622,473	811,866	2,448,710
40	18,980	2,160,114	1,082,666	3,261,760
50	31,660	2,744,090	1,329,071	4,104,821
80	51,320	4,369,093	2,129,526	6,549,939
100	58,960	5,456,700	2,674,207	8,189,867

## 5. Concluding Remarks

This article presents a novel RL-based algorithm designed to optimize the economic and environmental cost of power scheduling. The algorithm utilizes a contextually corrective agent-based RL environment, which simulates power scheduling dynamics using the framework of MDP. To evaluate the applicability and performance of the proposed method, the algorithm is tested on different test systems comprising up to 100 generating units. It is demonstrated that the algorithm provides superior solutions and is scalable to handle larger power systems. The potential for incorporating renewable power sources and investigating their impacts further highlights the versatility and applicability of the proposed method in addressing real-world power scheduling challenges.

**Author Contributions:** Conceptualization, A.S.E. and Y.J.K.; methodology, A.S.E. and C.P.; software, A.S.E.; validation, C.P. and Y.C.; formal analysis, A.S.E.; investigation, C.P. and Y.C.; resources, Y.J.K.; data curation, Y.C.; writing—original draft preparation, A.S.E.; writing—review and editing, Y.J.K.; supervision, Y.J.K.; project administration, Y.J.K.; funding acquisition, Y.J.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2021R111A3047456).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Nomenclature

### Indices

$n$ :	Number of units.
$m$ :	Number of emission types.
$\mathcal{T} = \{1, 2, \dots, n\}$ :	Indices of all units, $i \in \mathcal{T}$ .
$\mathcal{M} = \{1, 2, \dots, m\}$ :	Indices of all types of emissions, $h \in \mathcal{M}$ .
$\mathcal{T} = \{1, 2, \dots, 24\}$ :	Indices of all periods, $t \in \mathcal{T}$ .

### Units and Demand Profiles

$p_{i*}^{max}, p_{i*}^{min}$ :	Max, min capacity of unit $i$ (MW).
$p_{it}^{max}, p_{it}^{min}$ :	Max, min capacity of unit $i$ at period $t$ (MW).
$p_{it}$ :	Power output of unit $i$ at period $t$ (MW).
$t_{i*}^{up}, t_{i*}^{down}$ :	Min online, offline duration of unit $i$ (hour).
$t_{it}$ :	Operating (online/offline) duration of unit $i$ at period $t$ (hour).
$t_{it}^{ON}, t_{it}^{OFF}$ :	Online (up), offline (down) duration of unit $i$ at period $t$ (hour).
$u_{it}^1, u_{it}^0$ :	Indicator if unit $i$ must – ON, must – OFF at time $t$ .
$d_t$ :	Demand at period $t$ (MW).
$r$ :	Percentage of demand for reserve capacity.

## Objective Function

$\mathcal{C}, \mathcal{C}_t$ :	Total generation cost function of a day and at period $t$ .
$\alpha_i, \beta_i, \delta_i$ :	Quadratic, linear, constant parameters of cost function of unit $i$ .
$\phi_h, \psi_{ih}$ :	Externality cost of emission type $h$ (\$/g), emission factor of unit $i$ for type $h$ (g/MW).
$c_{it}^{ON}, c_i^{OFF}$ :	Start – up cost at period $t$ , shutdown cost of unit $i$ .
Others	
$\mathbb{E}$ :	Expected value.
$\mathbb{I}$ :	Indicator function.

## References

- Asokan, K.; Ashokkumar, R. Emission controlled Profit based Unit commitment for GENCOs using MPPD Table with ABC algorithm under Competitive Environment. *WSEAS Trans. Syst.* **2014**, *13*, 523–542.
- Roque, L.; Fontes, D.; Fontes, F. A multi-objective unit commitment problem combining economic and environmental criteria in a metaheuristic approach. In Proceedings of the 4th International Conference on Energy and Environment Research, Porto, Portugal, 17–20 July 2017.
- Montero, L.; Bello, A.; Reneses, J. A review on the unit commitment problem: Approaches, techniques, and resolution methods. *Energies* **2022**, *15*, 1296. [\[CrossRef\]](#)
- De Mars, P.; O’Sullivan, A. Applying reinforcement learning and tree search to the unit commitment problem. *Appl. Energy* **2021**, *302*, 117519. [\[CrossRef\]](#)
- De Mars, P.; O’Sullivan, A. Reinforcement learning and A\* search for the unit commitment problem. *Energy AI* **2022**, *9*, 100179. [\[CrossRef\]](#)
- Jasmin, E.A.; Imthias Ahamed, T.P.; Jagathy Raj, V.P. Reinforcement learning solution for unit commitment problem through pursuit method. In Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Bangalore, India, 28–29 December 2009.
- Jasmin, E.A.T.; Remani, T. A function approximation approach to reinforcement learning for solving unit commitment problem with photo voltaic sources. In Proceedings of the 2016 IEEE International Conference on Power Electronics, Drives and Energy Systems, Trivandrum, India, 14–17 December 2016.
- Li, F.; Qin, J.; Zheng, W. Distributed Q-learning-based online optimization algorithm for unit commitment and dispatch in smart grid. *IEEE Trans. Cybern.* **2019**, *50*, 4146–4156. [\[CrossRef\]](#) [\[PubMed\]](#)
- Navin, N.; Sharma, R. A fuzzy reinforcement learning approach to thermal unit commitment problem. *Neural Comput. Appl.* **2019**, *31*, 737–750. [\[CrossRef\]](#)
- Dalal, G.; Mannor, S. Reinforcement learning for the unit commitment problem. In Proceedings of the 2015 IEEE Eindhoven PowerTech, Eindhoven, Netherlands, 29 June–2 July 2015.
- Qin, J.; Yu, N.; Gao, Y. Solving unit commitment problems with multi-step deep reinforcement learning. In Proceedings of the 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, Aachen, Germany, 25–28 October 2021.
- Ongsakul, W.; Petcharak, N. Unit commitment by enhanced adaptive Lagrangian relaxation. *IEEE Trans. Power Syst.* **2004**, *19*, 620–628. [\[CrossRef\]](#)
- Nemati, M.; Braun, M.; Tenbohlen, S. Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming. *Appl. Energy* **2018**, *2018*, 944–963. [\[CrossRef\]](#)
- Trüby, J. *Thermal Power Plant Economics and Variable Renewable Energies: A Model-Based Case Study for Germany*; International Energy Agency: Paris, France, 2014.
- Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Vamvoudakis, K.G., Wan, Y., Lewis, F.L., Cansever, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
- Wilensky, U.; Rand, W. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*; The MIT Press: London, UK, 2015.
- Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*; The MIT Press: London, UK, 2018.
- Matzliach, B.; Ben-Gal, I.; Kagan, E. Detection of static and mobile targets by an autonomous agent with deep Q-learning abilities. *Entropy* **2022**, *24*, 1168. [\[CrossRef\]](#) [\[PubMed\]](#)

19. Adam, S.; Busoniu, L.; Babuska, R. Experience replay for real-time reinforcement learning control. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 201–212. [[CrossRef](#)]
20. Yildirim, M.; Özcan, M. Unit commitment problem with emission cost constraints by using genetic algorithm. *Gazi Univ. J. Sci.* **2022**, *35*, 957–967.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.