

Article

# A Deep Reinforcement Learning-Based Power Resource Management for Fuel Cell Powered Data Centers

Xiaoxuan Hu <sup>1</sup> and Yanfei Sun <sup>2,\*</sup>

<sup>1</sup> School of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; xiaoxhu@outlook.com

<sup>2</sup> School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

\* Correspondence: sunyanfei@njupt.edu.cn

Received: 31 October 2020; Accepted: 30 November 2020; Published: 3 December 2020



**Abstract:** With the increase of data storage demands, the energy consumption of data centers is also increasing. Energy saving and use of power resources are two key problems to be solved. In this paper, we introduce the fuel cells as the energy supply and study power resource use in data center power grids. By considering the limited load following of fuel cells and power budget fragmentation phenomenon, we transform the main two objectives into the optimization of workload distribution problem and use a deep reinforcement learning-based method to solve it. The evaluations with real-world traces demonstrate the better performance of this work over state-of-art approaches.

**Keywords:** green data center; deep reinforcement Learning; fuel cell; workload scheduling

---

## 1. Introduction

With the increasing number of cloud computing and Internet services, high energy consumption contributed by data center loads has become a crucial issue [1]. For example, Google and Microsoft pay tens of millions of dollars for electricity, and 50 tons of carbon dioxide a year are produced due to the high power consumption [2–4]. Besides the rising pressure from energy consumption and the deterioration of the climate, the power budget provided by the power transmission infrastructure of data centers usually limits the number of servers that can be added to address the growing load [5,6]. Usually, the problem is alleviated by developing new data center facilities and new power infrastructure, but this is expensive and time-consuming. Therefore, in this resource-constrained environment, maximizing the use of the existing power infrastructure and becoming environmentally friendly are two important goals that should be considered.

On the one hand, as one alternative green energy resource, fuel cells have emerged as a promising energy source for data centers due to their advantages of high energy efficiency, high reliability, and low carbon dioxide emission [7,8]. Although fuel cells show many advantages, they are slow in changing the output power [9], i.e., it may take a few minutes to reach the energy requirement. In order to address this challenge, some research was proposed introducing energy storage devices to reduce the effect of limited load following, which may add extra costs [10,11].

On the other hand, to maximize power resource use of the existing power infrastructure, a major challenge arises from power provisioning in the power delivery infrastructure of data centers. This challenge is called *power budget fragmentation* [12]. In the multi-level power delivery infrastructure, if some servers with synchronous power consumption mode are connected to the same power node, a high amplitude fast peak will be produced at this low-level node, which may quickly consumes the local power budget. In such a data center, although the high-level node still has a large power

budget, there is no space to add more servers to these low-level power nodes. As servers can only be powered by low-level power nodes, if the power budget is highly dispersed in the lower-level power supply infrastructure, the abundant power budget on the high-level node will never be used, resulting in low data center efficiency. Therefore, how to effectively reduce the rapid peaks at the low-level nodes and increase local power headrooms for supplying a greater number of servers is still a challenging problem. Recent work has paid much attention to power capping and load balancing for energy management in data centers. However, their potential is still largely limited by the power budget fragmentation. These efforts are mainly focused on the operations at high-level nodes while there are still amounts of power headroom which can be used at low-level nodes. Little work has been investigated in this field [5,13]. Hsu et al. [12] proposed a framework for modeling and rating the temporal heterogeneity between different services, achieving an efficient power infrastructure for data centers. If loads with the same energy consumption mode are placed at the same low-level node, high peaks will emerge. A clustering algorithm based on numerical analysis is used to distribute these loads among different low-level nodes. The proposed framework provides a promising option for exploring energy management at low-level nodes, but it still requires much more information about the performance metrics of the history-based loads.

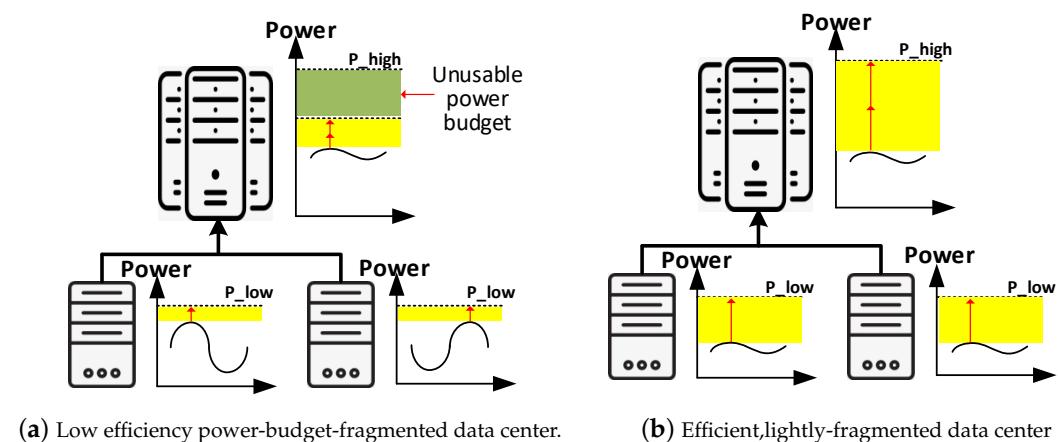
Considering the above two aspects, there are two key objectives to efficiently handling the energy management: (1) reducing the effect of limited load following; (2) mitigating the peak values at low level nodes. The above two objectives can be transferred as an optimization problem that makes the energy consumption curve more smooth at the low-level nodes in real time. However, there are still two challenges to solve this optimization problem. First, existing works need the knowledge of future data requests to control the energy supply of fuel cells. Unfortunately, it is difficult to accurately estimate future energy consumption of data centers, which was shown in [14,15]. Second, in order to ensure the energy demand gap between each time slots is as low as possible, the traditional methods are difficult to deal with high-dimensional calculation [16]. Due to the difficulty of directly solving this problem through conventional optimization methods, we employ the deep Q-learning Network (DQN)-based methods to conquer this challenge. Introducing the DQN-based methods to improve the energy efficiency of data centers is not new, some previous work has focused on this combination [17–20]. However, contrast to existing work, our objective is to propose a green workload approach that jointly realizes the effective use of fuel cells and the reduction of power budget fragmentation. More precisely, we design a fine-grained DQN-based method, aimed at optimizing the above-mentioned two objectives at the same time. To improve the efficiency of the proposed method, we also introduce an acceleration mechanism to deal with high dimensional computing. With the support of real-world traces, our approach can achieve stable performance and restrict training loss into a low bound. In addition, our approach can effectively reduce the power budget fragmentation and the variation of energy consumption. Consequently, the proposed approach requires less peak energy consumption and a greater proportion of energy available over state-of-the-art methods, including the Static, Random and k-means.

The contributions of our work are summarized as follows:

- By jointly considering the applying of fuel cells and maximization of power resource use for data centers, we formulate this objective as a workload optimization problem and identify the key to achieving this target by mitigating the variation of energy consumption.
- We propose an effective use of the power resources approach by employing improved deep Q-learning methods. A real state experience pool is introduced in the DQN agent, aimed at reducing the number of redundant state calculations.
- We evaluate the performance of our approach through a simulation with real-world data center traces. Simulation results show that the proposed approach has good effectiveness and feasibility compared with state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 presents the motivation. In Section 3, we introduce the system model, and the F-DQN algorithm design is discussed in Section 4. We evaluate the performance of our approach in Section 5. Finally, Section 6 presents the related work, and Section 7 concludes the paper.

At present, most conventional data center infrastructure is deployed by a multi-level transmission design which is a tree-like structure [21]. The power from the grid is not directly delivered to servers. In fact, each server is powered by a leaf power node which is powered by the higher-level power node. This method can improve the stability and reliability of data center infrastructure. However, a particular problem that this causes is the bad effects on the power budget use which is called *Power Budget Fragmentation*. More specifically, if the power demand in leaf power node changes with high amplitude, the local power budget will be consumed quickly. Although the power demand in higher-level power nodes has not changed much over time, there is some power headroom at the higher-level node that has not been exploited, leading to inefficient data centers. On the contrary, if there are not many rapid power peaks at leaf power nodes, much more available power headroom can be used at higher power nodes. For ease of understanding, we show an example in Figure 1.



**Figure 1.** Power budget fragmentation.

## 2. Motivation

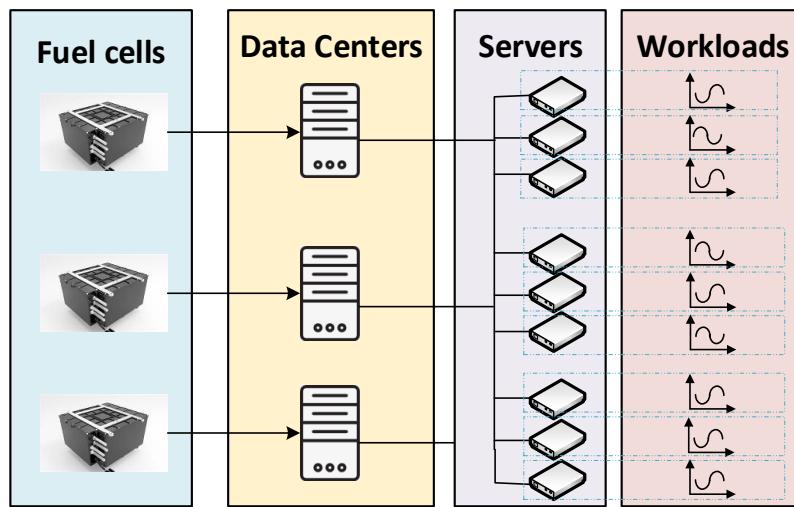
On the other hand, fuel cells are a promising energy resource for powering data centers due to the high energy efficiency and lower carbon emission. Nevertheless, fuel cells are subject to the weakness of limited load following. When the power demand changes with many rapid power peaks or valleys, fuel cells cannot provide sufficient energy supply in time [22]. Therefore, if we apply fuel cells as the energy supply for data centers, the amplitude of the power demand should not be too high. Traditional methods use energy storage devices to make up the insufficient energy supply in time caused by the limited load following. However, due to the limited capacity of energy storage devices, this solution cannot perform well when it meets large rapid local peaks. In addition, as can be seen from the above example, the stable power demand arising at higher nodes may not mean that the power demand at the leaf power nodes is also stable. The changes of power demand may be totally different at different power node levels. If we realize that the power demand in each level of power nodes is stable, the data centers can be powered by fuel cells with high efficiency and light fragmentation. Therefore, we take one step forward by studying how to manage the workloads from different servers to further optimize the power demand in each leaf power nodes.

## 3. System Model

Large-scale data centers usually apply tree-like and multi-level power infrastructure for better workload management, such as Google and Facebook. Each data center consists of several suits which

is equipped with several top-level power nodes. Each top-level power node is equipped with some secondary power nodes, which are further fed to a group of reactive power panels. Therefore, each rack consists of dozens of servers and the power budget at each node is the sum of its children's budget [23].

For better presentation, we consider a simplified model that includes several data centers, servers, fuel cells, and their workloads, as shown in Figure 2. Data centers distributed in different suits are denoted by a set  $D = \{d_0, d_1, \dots, d_n\}$ . Each data center is linked with several servers, which is denoted by a set  $S = \{s_{n0}, s_{n1}, \dots, s_{nm}\}$ . Each server receives a workload from the information network, which can be scheduled by the servers themselves. Therefore, we consider a set of users' workloads  $W = \{w_{n0}, w_{n1}, \dots, w_{nm}\}$ , each of which require more or less energy supply in different time slots. In addition, our model considers a discrete time series, which is denoted as  $T = \{0, 1, \dots, t\}$ .



**Figure 2.** System model.

If a large amount of workloads arrive, the improvement of computing resources use will consume more energy. We denote the energy consumed by workloads of server  $w_{nm}$  data center  $d_n$  at server  $s_{nm}$  in the time slot  $t$  as  $f_{nm}(t)$ . The relationship between  $f_{nm}(t)$  and  $w_{nm}(t)$  can be expressed by:

$$f_{nm}(t) = \mathbb{F}_{nm}(w_{nm}(t)), \forall n, m, t \quad (1)$$

where  $\mathbb{F}(\cdot)$  is a non-decreasing function. According to the existing works [14,15], a linear function is considered in this paper.

Based on the definition above, the energy demand of each server  $s_{nm}$  in time slot  $t$  is given as

$$u_{nm}(t) = \sum_{k=0}^K f_{nm}^k(t), \forall n, m, t \quad (2)$$

Let  $G_n(t)$  be the energy supply of fuel cell for data center  $n$  at time slot  $t$ . Because of the characteristic of slow load following of fuel cells,  $G_n(t)$  is given as

$$G_n(t) = G_n(t - 1) + \Delta G_n(t - 1), \forall n, t \quad (3)$$

Because of the limited capacity of fuel cells,  $G_n(t)$  is constrained by  $G_n^{max}(t)$ , as follows

$$0 \leq G_n(t) \leq G_n^{max}(t), \forall n, t \quad (4)$$

Our proposal aims to manage the coming workloads among servers to get with the limited variation of energy supply from fuel cells, which can be defined as

$$u_{nm}(t) \leq G_n(t), \forall n, t \quad (5)$$

Therefore, we are going to optimize the sum of energy variation of energy demand from each data center by planning the servers at each time slot. The future workload information should be obtained in advance. However, only the current workload information can be knew in practice. The solution is performed without the knowledge of future incoming workloads. Although some data profiles can be predicted in advance, the online energy management is a very popular topic in recent research [24,25]. It is difficult to predict energy demand profiles in all cases. In addition, the constraints (3) may cause the "time coupling" property. To be specific speaking, the current energy variation can have an effect on the future energy output of fuel cells. Dynamic programming is an alternative solution to deal with this issue. However, it will also bring the "curse of dimensionality" problem. Consequently, these challenges motivate us to propose a deep learning-based approach to solve the problem.

#### 4. Deep Reinforcement Learning Based Distribution Method

##### 4.1. Deep Reinforcement Learning Problem Formulation

In this section, we investigate Reinforcement learning-based energy management optimization.

Reinforcement learning is an effective method that can learn to realize the maximum profits in different situations [26]. The key elements are state, reward, action, and agent. Reinforcement learning is to use an agent to learn a series of actions and the corresponding rewards. Each state corresponds to the rewards produced by all actions according to the agent's reward function. Then, the agent will choose the appropriate operation according to a strategy and the state will be changed. Reinforcement learning is a promising method that does not need any prior knowledge, which is an ideal choice to optimize the energy demand in data centers. However, traditional reinforcement learning is limited to the action space and the sample space. The realistic tasks often have a large state space and continuous action space. If the input data is image or sound, it often has a very high dimension, which is difficult for traditional reinforcement learning to deal with. Deep reinforcement learning is proposed to solve this challenge, which is to combine the high dimension input of deep learning with reinforcement learning.

First, we need to define the elements of reinforcement learning in our model, including state, action, and reward.

- $s$  is the state space. The goal of our proposal is to decide which data center is assigned to each request.  $n$  denotes the number of data centers in the previous section. Hence, we denote the state space  $s_{w_{nm},t,n} = \{0, 1, \dots, nm - 1\}$
- $a$  is the action space defined as choosing the data center  $n$ . Therefore, we also have  $a_{w_{nm},t,n} = \{0, 1, \dots, nm - 1\}$
- In this problem, our goal is to mitigate the variation of power demand of all the data centers. The sum of all the data centers' power demands in each time slot is defined as:

$$P_D(t) = \sum_{n=0}^n p_D^n(t), \forall t \quad (6)$$

where  $p_D^n(t)$  is the total power demand of data center  $n$  in each time slot, which can be calculated by

$$p_D^n(t) = \sum_{m=0}^m f_{nm}(t), \forall n, t \quad (7)$$

where  $f_{nm}(t)$  is the energy demand of each workload  $w_{nm}$  in time slot  $t$ , which was defined before. Then, the variation of power demand for all the data centers between the adjacent time slots is

$$\Delta P_D^n(t) = p_D^n(t) - p_D^n(t-1), \forall n, t \quad (8)$$

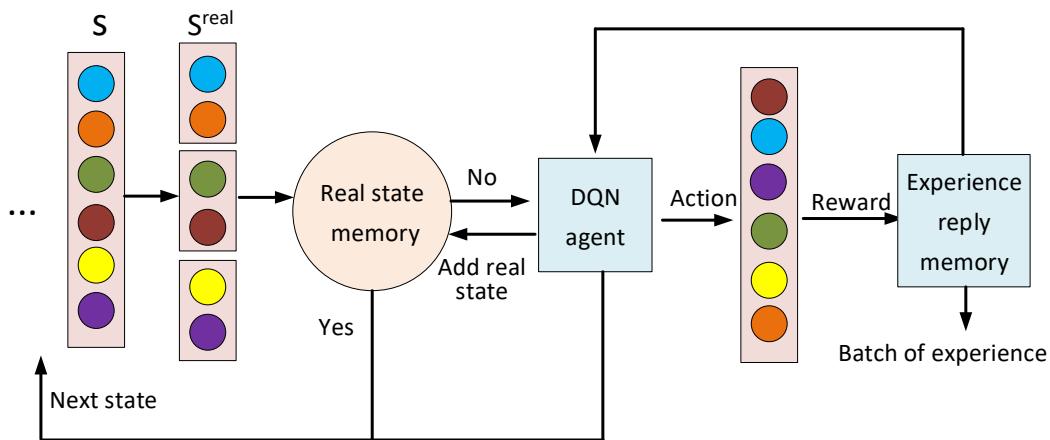
In addition, the variation in power demand of all the data centers cannot exceed the capacity of fuel cells. Therefore, the reward function can be defined as

$$\mathbb{R}(t) = \begin{cases} 0 & \Delta P_D^n(t) > \Delta G_n(t) \\ \sum_{n=0}^n \frac{p_D^n(t)}{|\Delta P_D^n(t)|} & \text{others} \end{cases} \quad (9)$$

Finally, the state transition samples of Reinforcement learning can be represented as  $(s_{w_{nm},t,n}, a_{w_{nm},t,n}, \mathbb{R}(t), s_{w_{nm},t+1,n})$

#### 4.2. F-DQN Algorithm Design

The dimensions of action space and state space can be very large in our system model. In each step of the learning process, the number of actions which are learnt by the agent can be reached up to  $n^m T$ . Therefore, with the increasing dimensions of action space and state space, the amount of decisions needed will increases exponentially, which is hard to implement by applying traditional DQN. In addition, there are many meaningless actions during the learning process because of data center architecture. For example, there are four racks in each data center. If we move a workload from one rack to another that belongs to the same data center, there is no effect on the result of the reward. It means that the efficiency of learning is greatly reduced. Facing the high dimensional numerical calculation, we proposed an acceleration method based on deep Q-learning called F-DQN to find optimal action which brings the maximum reward function in our proposed model and the workflow of F-DQN is shown in Figure 3.



**Figure 3.** The workflow of F-DQN.

To improve the efficiency of the deep Q-learning algorithm, an additional state space  $s_{w_{nm},t,n}^{real}$  is introduced, which is defined as:

$$s_{w_{nm},t,n}^{real} = \{0, 0, \dots, 1, 1, \dots, n, n\} \quad (11)$$

The relationship between  $s_{w_{nm},t,n}$  and  $s_{w_{nm},t,n}^{real}$  is:

$$s_{w_{nm},t,n}^{real} = \left[ \frac{s_{w_{nm},t,n}}{n} \right] \quad (12)$$

In each episode, before the current status  $s_{w_{nm},t,n}$  is sent to evaluate Q-networks, it will be transferred to  $s_{w_{nm},t,n}^{real}$  according to 11. Then, the new status will be put into an  $s^{real}$  experience memory. If the new status is the same as the status which is stored in the  $s^{real}$  experience memory, F-DQN will skip this episode and conduct the next state. If the new state is different from the state which is stored in the  $s^{real}$  experience memory, it will be sent to the DQN network and will conduct the learning process.

Then, we propose the use of F-DQN as an online method to perform optimal workload allocation at the lower level nodes. The general architecture of our proposed method is depicted in Figure 3. With the four fundamental properties given in Section 4.1, we can present the learning methodology. The key rationale of our methodology is the policy  $\pi$ . Then,  $\pi(a(t)|s_{w_{nm},t,n})$  is denoted as the probability of choosing action  $a(t)$  when the environment state is  $s_{w_{nm},t,n}$ . Given  $s_{w_{nm},t,n}$  and  $a_{w_{nm},t,n}$ , we define an action-value function  $Q^\pi(s_t, a_t)$  to evaluate the expected reward of policy  $\pi$  as follows.

$$Q^\pi(s_t, a_t) = E^\pi \left[ \sum_{k=0}^{\infty} \lambda^k \mathbb{R}(t+k+1) | s_t, a_t \right] \quad (13)$$

where  $\lambda$  is a discount factor. Let  $\lambda \in [0, 1]$  so that the rewards in the nearer future have larger weights. Then, the evaluation method of  $Q((s_t, a_t))$  is updated as follows:

$$Q(s_t, a_t) = Q((s_t, a_t)) + \alpha (\mathbb{R}(t) + \lambda \max_{a'_t} Q(s'_t, a'_t) - Q((s_t, a_t))) \quad (14)$$

where  $\alpha$  is a learning rate, which satisfies  $\alpha \in (0, 1]$ .  $Q(s'_t, a'_t)$  represents the optimal future value. Due to the influence of many elements on future rewards, the traditional Reinforcement learning cannot obtain  $Q^\pi(s_t, a_t)$  accurately. Hence, DQN (Deep Q-Network) is used to train a function  $Q(s_t, a_t, \theta_t)$  that approximates the action-value function with high accuracy. DQN can be considered to be a composite function, which takes state  $s_t$  as input and outputs an operation  $a_t$ . To minimize the loss after updating the weights, we define the loss function as the variance between the target value and the predicted value. The loss function is expressed as follows:

$$L(\theta_t) = E[(\mathbb{R}(t) + \lambda \max_{a'_t} Q(s'_t, a'_t, \theta'_t) - Q(s_t, a_t))^2] \quad (15)$$

In addition, another independent network with the same structure named target network  $Q_{target}((s_t, a_t, \theta_{target,t}))$  is introduced to make the method more efficient. Every few steps, the weights of the main network are copied to the weights of the target network. As the target network remains unchanged for a period of time, the correlation between the current Q value and the target Q value is reduced and the stability of the algorithm will be improved. In each step, the samples  $(s_t, a_t, \mathbb{R}_t, s_{t+1})$  obtained from the interaction between agent and environment are stored in experience replay memory. A batch of the samples will be randomly selected for training DNN, in order to make the agent learn from past experiences stored in the memory.

## 5. Experimental Results

### 5.1. Simulation Settings

We use several kinds of workload traces collected from the Wiki data center, which show different characteristics [27]. In this experiment, the length of each time slot is set to 1 hour. To facilitate calculation and comparison, all the energy consumption data are normalized. We use a CPU-based server, which has 16 GB DDR4 memory, 2.8 GHz Inter Core i7, and 512 GB drive. Python 3.6.8 with

Pytorch 1.6.0 is used to provide software environment. The other key experimental settings are given in Table 1.

**Table 1.** Experimental parameter settings

Parameter	Value	Parameter	Value
$T$ (hour)	600	$\alpha$	0.01
$n$	2–9	$\lambda$	0.1
$m$	3–4	memory capacity	2000
$\Delta G_{max}$ (kW/h)	10	target update frequency	100
$G_0$ (kW/h)	5	batch size	128

We compare our proposed F-DQN-based method with the following schemes.

- Static: assuming the coming workloads are not changed to other servers.
- Random: The coming workloads in each time slot are changed randomly to all the servers.
- K-means: The coming workloads are transferred through k-means to get with the optimal variation of energy consumption. For each workload, the asynchrony score is calculated and each server will be considered to be a data point. Then we apply k-means clustering to these data points and obtain a set of cluster [12].

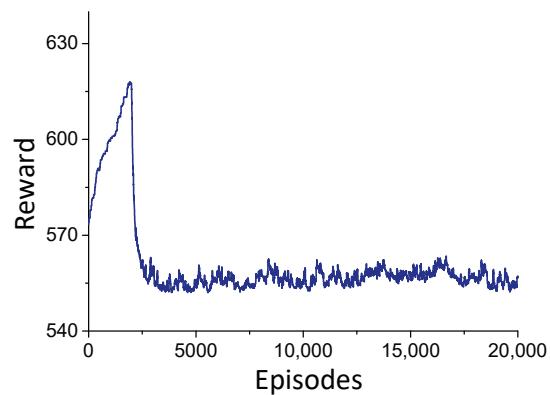
## 5.2. Simulation Results

As our objective is to minimize the variation of energy consumption and unusable power budget for fuel cell powered data centers, we focus on the metrics in four aspects: (1) the performance of F-DQN algorithm (in Section 5.2.1), (2) energy consumption traces before and after optimization (in Section 5.2.2), (3) the comparison of variation of energy consumption among different number of data centers (in Section 5.2.3) and (4) the comparison of proportion of power budget among different number of racks (in Section 5.2.4).

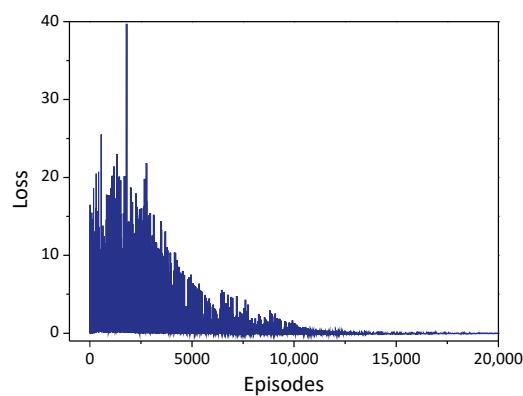
### 5.2.1. The Performance of F-DQN Algorithm

Figure 4 shows the reward value at each training episode. The convergence of the reward values achieves stale convergence, which indicates the stable convergence of proposed algorithm. In the beginning of the training episode, the reward value is around 570. This is because the weights in main networks are initialized randomly. With the increasing of training episodes, before about the 2500th episode, the reward value increases to about 620. This is due to the fact that the parameters in greedy rule do not decay to the minimal value, and the agent takes more exploration in the initial several training. Therefore, the main networks are not well trained in the beginning. At around the 2500th episode, the reward value dropped rapidly from 620 to about 560. After around 5000 training episodes, the smoothed reward value curve is convergent to the value about 555, which shows good convergence characteristic of the proposed algorithm.

Figure 5 presents the changing trend of learning loss by F-DQN in the training process. The convergence of the proposed algorithm is also illustrated. It can be seen that since the input data in F-DQN changes gradually, the curve does not decline smoothly. At the beginning of the training process, Figure 5 also reflects the same phenomenon as Figure 4. Initially, the agent always take exploratory moves (i.e., random action), which leads to a high immediate loss value. When training step reaches around 3000, the loss of F-DQN start to decrease gradually which means the algorithm eventually converges. Therefore, it can be found that the F-DQN has a better training performance.



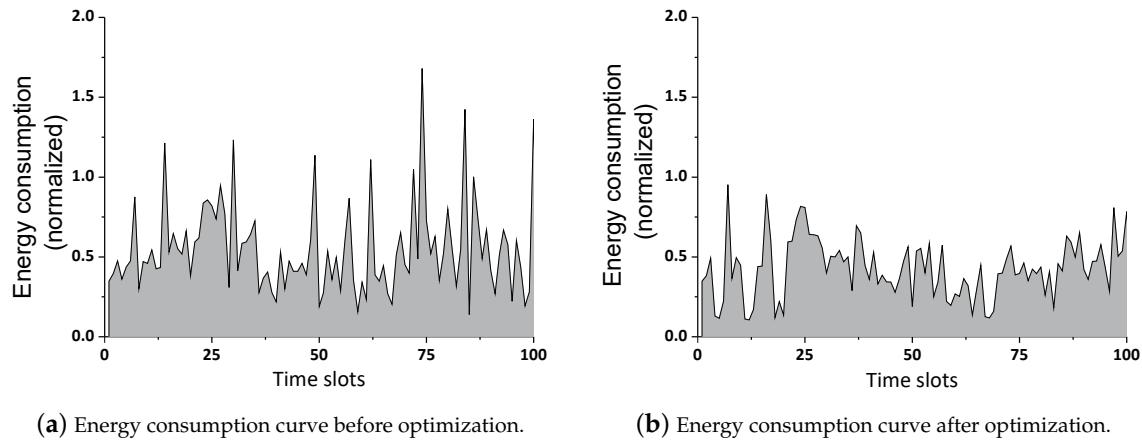
**Figure 4.** The reward curve of F-DQN.



**Figure 5.** Training loss of F-DQN.

### 5.2.2. Energy Consumption Traces before and after Optimization

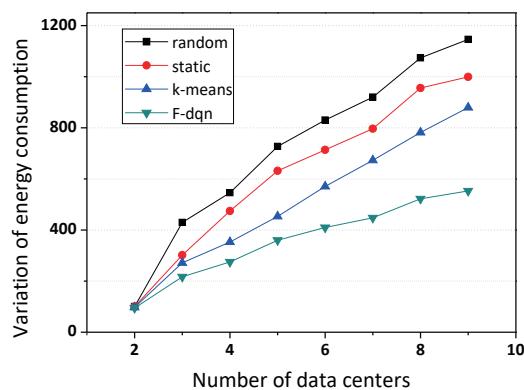
Figure 6 presents the comparison of energy consumption curve at low-level node before and after optimization. In the two figures, the Y-axis represents the normalized energy consumption at each time slot and the X-axis represents the time slots from 0th hour to 100th hour. As shown in Figure 6a, the maximum peak value can reach about 1.7 and the maximum energy gap is about 1.3 in 85th time slot. Therefore, only about 1/3 of power headroom is available for adding extra services without the optimization of power resource use. Figure 6b shows the energy consumption curve at the same node after applying the proposed method. Compared with Figure 6a, the peak value in Figure 6b is lower, which is only about 1 at 8th time slot. Besides, the maximum energy gap is no more than 1 because of the constraints on the characteristics of fuel cells. Obviously, the power headroom in Figure 6b is about 1/2, which is bigger than that in Figure 6a.



**Figure 6.** The comparison between before and after applying F-DQN.

#### 5.2.3. The Comparison of Variation of Energy Consumption among Different Number of Data Centers

We compare the variation of energy consumption of our proposed algorithm (marked as "F-DQN" in green) with other three baseline approaches in terms of different number of high-level nodes. The Y-axis is accumulated by the energy consumption gap between each adjacent time slots. In this simulation, the number of time slots is set to 600. The energy consumption in each time slot is also normalized. As shown in Figure 7, our approach yields less variation of energy consumption when the number of high-level nodes exceeds 2, while the other three baselines generate much more variation of energy consumption and the curves grow sharply as the number of high-level nodes grows. When the number of high-level nodes is set to 2, the results of four methods are very close because there is not enough space to exchange workloads. The K-means method approximately follows the linear trend with the increase of number of high-level nodes, while our proposed method works better as the number of high-level nodes increase. Therefore, our DQN-based method can effectively reduce the energy gap between adjacent time slots, especially in higher dimension.

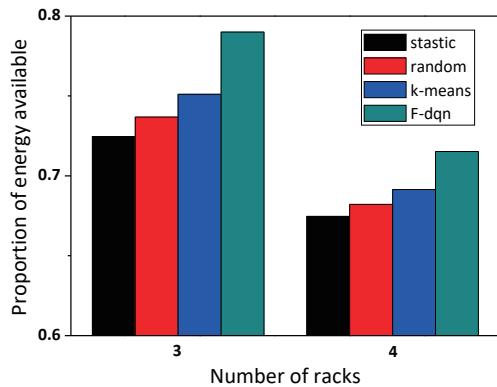


**Figure 7.** The Comparison of Variation of Energy Consumption.

#### 5.2.4. The Comparison of Proportion of Power Budget among Different Number of Racks

As to inspecting the energy efficiency use, we also compare the proportion of power budget with the three baselines in terms of different number of low-level nodes at each high-level node. The proportion of power budget is denoted as the ratio of unused power to total energy. We inspect how the number of low-level nodes impact this metric. As shown in Figure 8, all the four approaches consume more energy with the increase of the number of low-level nodes. However, in both 3 and 4 low-level nodes, our approach can achieve the highest proportion of power budget. Therefore,

our DQN-based method can achieve better energy saving efficiency and mitigate power budget fragmentation over three baselines. More precisely, our proposed approach can save energy by up to 7.5%, 5.2% and 4.3%, on average, more than the static, random and k-means, respectively.



**Figure 8.** The Comparison of Proportion of Power Budget.

## 6. Related Work

### 6.1. Fuel Cells for Data Centers

Fuel cells emerged as a promising energy source for data centers due to their advantages of high energy efficiency, high reliability, and low carbon dioxide emission [11]. Therefore, fuel cells are useful as a second redundant energy source for relatively longer peak intervals. If a malfunction or maintenance occurs, the redundant unit supplies the energy needed for assuring uninterrupted operation [28]. Therefore, the fuel cells has been applied in many areas. Rieksttin et al. [29] introduced the key research issues in the design of data center power distribution system powered by fuel cells. Zhou et al. [30] first tried to quantitatively analyze the benefits of fuel cell power generation, and explained how to realize intelligent coordination between power grid in data center networks and fuel cell power generation. Li et al. [9] first proposed an ESD classification framework for data centers powered by fuel cells. A variety of power capping strategies with different degrees of knowledge of fuel cells and workload behavior are introduced to evaluate the effect on workload performance and ESD size. Sevencan et al. [31] studied the economic feasibility of a combination of cooling, heating and power system based on fuel cells in an existing data center. The feasibility of this hybrid power system can be predicted in the future when the energy price changes.

### 6.2. Deep Reinforcement Learning for Data Centers

For the application of deep reinforcement learning methods in data centers, many studies were carried out in different areas. Chen et al. [17] developed a two level system based on DRL methods to simulate the peripheral and central nervous systems of animals for solving the scalability problem of data centers. Yang et al. [18] proposed a new green cloud data center architecture, aiming at the high energy consumption of data centers. A scheduling control engine and in intelligent refrigeration engine based on DRL are introduced. The experiment result showed the architecture can effectively reduce energy consumption and increase resource use rate of data centers. Ran et al. [19] proposed a DRL-based optimization framework, which considers both IT and cooling systems to improve the energy efficiency of data centers. Comparing with conventional approaches, the proposed algorithm can achieve a better compromise between energy saving and quality of service. Yi et al. [20,32] established an assignment algorithm by using DRL to deal with the increasing, persistent and computationally intensive tasks in recent computing requirement. The power and thermal dynamics of data centers are captured by training the deep Q-network, leading the reduction of the online convergence speed, low energy efficiency and potential server overheating in the process of DRL

exploration. Gao et al. [33] used DRL to predict the production of each renewable energy source and the energy demand of each predefined region. To minimize the number of SLO violations, total energy cost and total carbon emissions, an optimization problem was proposed to match different renewable energy resources with different regions. Li et al. [34] proposed a novel DRL architecture to optimize data center cooling control. The proposed method provided an end-to-end cooling control algorithm combined with deep deterministic strategy gradient algorithm, which is helpful to improve the cooling efficiency.

## 7. Conclusions

This paper focuses on the power budget fragmentation problem in data center architecture powered by fuel cells. Observing the limitations of existing approaches that aim at minimizing energy cost while neglecting the resource use at high-level nodes, this paper jointly considers objectives of both the energy supply by fuel cell and resource use. Due to online environment of data center architecture, the main target is formulated as an optimization problem with minimization the variation of energy consumption at low-level nodes. A fine-grained workload distribution approach is designed via the deep reinforcement learning method and a real state pool is introduced in traditional DRL to deal with high computational dimension. The evaluation based on real-world traces demonstrates better performance of the proposed approach over state-of-the-art methods. The simulation results show that our proposed method can maintain a better training performance and save about 16% power headroom. Our results on the real trace show that we can reduce the energy gap and save more energy at around 5%.

At the end of this paper, we list a few issues if the proposed method is applied to practical data centers. At first, the limitation output of fuel cells will have a huge effect on the performance of the proposed method. Using heterogeneous energy resources to meet different kinds of energy demand may be an effective way to solve the problem. The second issue is the parameter settings. Our experiments show that the tuning process is almost inevitable. How to design a stable DRL-based method to deal with data diversity is our future work.

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

**Funding:** This paper was supported in part by the National Natural Science Foundation of China under Grant 61772286 and Grant 61802208.

**Acknowledgments:** This paper was supported in part by the National Natural Science Foundation of China under Grant 61772286 and Grant 61802208, in part by Project funded by China Postdoctoral Science Foundation Grant 2019M651923, in part by Natural Science Foundation of Jiangsu Province of China under Grant BK20191381, in part by Primary Research and Development Plan of Jiangsu Province Grant BE2019742, in part by the Natural Science Fund for Colleges and Universities in Jiangsu Province under Grant 18KJB520036.

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

## References

- Wei, Z.; Yonggang, W.; Lei, L.L.; Fang, L.; Rui, F. Cost Optimal Data Center Servers: A Voltage Scaling Approach. *IEEE Trans. Cloud Comput.* **2018**, *1*. [[CrossRef](#)]
- Qureshi, A. Power-Demand Routing in Massive Geo-Distributed Systems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
- Ren, C.; Wang, D.; Urgaonkar, B.; Sivasubramaniam, A. Carbon-Aware Energy Capacity Planning for Datacenters. In Proceedings of the 2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Washington, DC, USA, 7–9 August 2012; pp. 391–400. [[CrossRef](#)]
- Seh, Z.W.; Kibsgaard, J.; Dickens, C.F.; Chorkendorff, I.; Norskov, J.K.; Jaramillo, T.F. Combining theory and experiment in electrocatalysis: Insights into materials design. *Science* **2017**, *355*, eaad4998. [[CrossRef](#)]

5. Pelley, S.; Meisner, D.; Zandevakili, P.; Wenisch, T.; Underwood, J. Power Routing: Dynamic Power Provisioning in the Data Center. *ACM SIGPLAN Not.* **2010**, *45*, 231–242. [[CrossRef](#)]
6. Qiang, W.; Deng, Q.; Ganesh, L.; Hsu, C.H.; Song, Y.J. *Dynamo: Facebook’s Data Center-Wide Power Management System*; IEEE: New York, NY, USA, 2012.
7. Zhao, L.; Brouwer, J.; James, S.; Siegler, J.; Peterson, E.; Kansal, A.; Liu, J. Servers Powered by a 10kW In-Rack Proton Exchange Membrane Fuel Cell System. In Proceedings of the International Conference on Fuel Cell Science, Engineering and Technology, Boston, MA, USA, 30 June 2014. [[CrossRef](#)]
8. Ng, M.F.; Zhao, J.; Yan, Q.; Conduit, G.J.; Seh, Z.W. Predicting the state of charge and health of batteries using data-driven machine learning. *Nat. Mach. Intell.* **2020**, *2*, 161–170. [[CrossRef](#)]
9. Li, Y.; Wang, D.; Ghose, S.; Liu, J.; Govindan, S.; James, S.; Peterson, E.; Siegler, J.; Ausavarungnirun, R.; Mutlu, O. SizeCap: Efficiently Handling Power Surges in Fuel Cell Powered Data Centers. In Proceedings of the 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), Barcelona, Spain, 12–16 March 2016. [[CrossRef](#)]
10. Miyazaki, M.R.; Sorensen, A.J.; Vartdal, B.J. Reduction of Fuel Consumption on Hybrid Marine Power Plants by Strategic Loading With Energy Storage Devices. *IEEE Power Energy Technol. Syst. J.* **2016**, *3*, 207–217. [[CrossRef](#)]
11. Hu, X.; Li, P.; Wang, K.; Sun, Y.; Zeng, D.; Guo, S. Energy Management of Data Centers Powered by Fuel Cells and Heterogeneous Energy Storage. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018. [[CrossRef](#)]
12. Hsu, C.H.; Deng, Q.; Mars, J.; Tang, L. SmoothOperator: Reducing Power Fragmentation and Improving Power Utilization in Large-scale Datacenters. *ACM SIGPLAN Not.* **2018**, *53*, 535–548. [[CrossRef](#)]
13. Kontorinis, V.; Zhang, L.; Aksanli, B.; Sampson, J.; Homayoun, H.; Pettis, E.; Tullsen, D.; Rosing, T. Managing Distributed UPS Energy for Effective Power Capping in Data Centers. *ACM SIGPLAN Not.* **2012**, *40*, 488–499. [[CrossRef](#)]
14. Yu, L.; Jiang, T.; Cao, Y.; Qi, Q. Carbon-Aware Energy Cost Minimization for Distributed Internet Data Centers in Smart Microgrids. *Internet Things J. IEEE* **2014**, *1*, 255–264. [[CrossRef](#)]
15. Guo, Y.; Gong, Y.; Fang, Y.; Khargonekar, P.P.; Geng, X. Energy and Network Aware Workload Management for Sustainable Data Centers with Thermal Storage. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2030–2042. [[CrossRef](#)]
16. Sun, P.; Wen, Y.; Han, R.; Feng, W.; Yan, S. GradientFlow: Optimizing Network Performance for Large-Scale Distributed DNN Training. *IEEE Trans. Big Data* **2019**, *1*. [[CrossRef](#)]
17. Chen, L.; Lingys, J.; Chen, K.; Liu, F. AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In *SIGCOMM ’18: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*; Association for Computing Machinery: New York, NY, USA, 2018. [[CrossRef](#)]
18. Yang, J.; Xiao, W.; Chun, J.; Hossain, M.S.; Muhammad, G.; Amin, S. AI Powered Green Cloud and Data Center. *IEEE Access* **2018**, *1*. [[CrossRef](#)]
19. Ran, Y.; Hu, H.; Zhou, X.; Wen, Y. DeepEE: Joint Optimization of Job Scheduling and Cooling Control for Data Center Energy Efficiency Using Deep Reinforcement Learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 645–655. [[CrossRef](#)]
20. Yi, D.; Zhou, X.; Wen, Y.; Tan, R. Toward Efficient Compute-Intensive Job Allocation for Green Data Centers: A Deep Reinforcement Learning Approach. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019. [[CrossRef](#)]
21. OpenComputeProject. Available online: <https://www.opencompute.org/> (accessed on 21 November 2019).
22. Hu, X.; Li, P.; Wang, K.; Sun, Y.; Zeng, D.; Wang, X.; Guo, S. Joint Workload Scheduling and Energy Management for Green Data Centers Powered by Fuel Cells. *IEEE Trans. Green Commun. Netw.* **2019**, *3*, 397–406. [[CrossRef](#)]
23. Zhang, Y.; Prekas, G.; Fumarola, G.M.; Fontoura, M.; Goiri, I.; Bianchini, R. History-Based Harvesting of Spare Cycles and Storage in Large-Scale Datacenters. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 755–770.
24. Yu, L.; Jiang, T.; Cao, Y. Energy Cost Minimization for Distributed Internet Data Centers in Smart Microgrids Considering Power Outages. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 120–130. [[CrossRef](#)]

25. Shi, W.; Li, N.; Chu, C.C.; Gadh, R. Real-Time Energy Management in Microgrids. *IEEE Trans. Smart Grid* **2015**, *8*, 228–238. [[CrossRef](#)]
26. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [[CrossRef](#)]
27. Wiki dump data. Available online <http://dumps.wikimedia.org/other/pagecounts-raw/> (accessed on 24 December 2018).
28. Kalhoff, N. Integration of fuel cell applications into the power supply for information and telecommunications technology. In Proceedings of the INTELEC 07 - 29th International Telecommunications Energy Conference, Rome, Italy, 30 September–4 October 2007; pp. 444–448. [[CrossRef](#)]
29. Riekstin, A.; James, S.; Kansal, A.; Liu, J.; Peterson, E. No More Electrical Infrastructure: Towards Fuel Cell Powered Data Centers. In Proceedings of the Workshop on Power-Aware Computing and Systems, HotPower 2013, Farmington, PA, USA, 3–6 November 2013. [[CrossRef](#)]
30. Zhou, Z.; Liu, F.; Li, B.; Li, B.; Jin, H.; Zou, R.; Liu, Z. Fuel Cell Generation in Geo-Distributed Cloud Services: A Quantitative Study. In Proceedings of the International Conference on Distributed Computing Systems, Madrid, Spain, 30 June–3 July 2014; pp. 52–61. [[CrossRef](#)]
31. Sevencan, S.; Lindbergh, G.; Lagergren, C.; Alvfors, P. Economic feasibility study of a fuel cell-based combined cooling, heating and power system for a data centre. *Energy Build.* **2016**, *111*, 218–223. [[CrossRef](#)]
32. Yi, D.; Zhou, X.; Wen, Y.; Tan, R. Efficient Compute-Intensive Job Allocation in Data Centers via Deep Reinforcement Learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1474–1485. [[CrossRef](#)]
33. Gao, J.; Wang, H.; Shen, H. Smartly Handling Renewable Energy Instability in Supporting A Cloud Datacenter. In Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 18–22 May 2020; pp. 769–778. [[CrossRef](#)]
34. Li, Y.; Wen, Y.; Guan, K.; Tao, D. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning. *IEEE Trans. Cybern.* **2017**. [[CrossRef](#)]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).