Online Scheduling Strategies for Renewable Energy Systems Based on Digital Twin Models

Yongqing Zhu*
Power Grid Planning and Research Center, Guizhou Power Grid Co., Ltd.
China
20285569@qq.com

Xuepeng Mou
Power Grid Planning and Research Center, Guizhou Power Grid Co., Ltd.
China
18275184868@163.com

Wei Wang
Power Grid Planning and Research Center, Guizhou Power Grid Co., Ltd.
China
772148970@qq.com

Xingyan Liu
Power Grid Planning and Research Center, Guizhou Power Grid Co., Ltd.
China
751960377@qq.com

Jiang Hu
Power Grid Planning and Research Center, Guizhou Power Grid Co., Ltd.
China
Hujiang9409@163.com

Abstract

To address the randomness and volatility of renewable energy generation technologies, we propose an online scheduling strategy for renewable energy power systems based on a digital twin (DT) model. By constructing a digital twin edge network (DTEN) model of the power system, the strategy enables real-time monitoring, efficient optimization scheduling, and decision support for renewable energy power systems. Firstly, under the constraints of computational resource limitations and latency requirements, the total energy consumption of the entire network is minimized through optimized offloading decisions and computational resource allocation. Secondly, this non-convex optimization problem is reformulated as a Markov Decision Process (MDP), and optimization algorithms within deep reinforcement learning (DRL) are employed to achieve optimal system decisions. Lastly, an online and real-time capable on-policy DRL algorithm is introduced to solve the offloading strategies and computational resource allocation strategies. Simulation results demonstrate that the proposed algorithm trains rapidly and effectively reduce system energy consumption while ensuring offloading and computational latency requirements are met.

CCS Concepts

• **Computing methodologies** → Modeling and simulation; Model development and analysis; Model verification and validation.

Keywords

Digital twin, Mobile edge computing, Deep reinforcement learning, Resource allocation and optimization

*Corresponding author.

AIAHPC 2024, July 19–21, 2024, Zhuhai, China

ACM ISBN 979-8-4007-1004-9/24/07
https://doi.org/10.1145/3690931.3690981

ACM Reference Format:
Yongqing Zhu, Xuepeng Mou, Wei Wang, Xingyan Liu, and Jiang Hu. 2024. Online Scheduling Strategies for Renewable Energy Systems Based on Digital Twin Models. In *International Conference on Artificial Intelligence, Automation and High Performance Computing (AIAHPC 2024), July 19–21, 2024, Zhuhai, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3690931.3690981

1 Introduction

With the demand for energy transformation in various countries around the world, the power system, as an important pillar of energy supply, urgently needs to develop "low-carbon" and "green" power generation energy, and achieve "intelligent" and "safe" operation modes. Mobile edge computing (MEC) [1-2], as one of the key technologies, is gradually becoming an important force to promote this change. Reference [3] constructed a joint computing offloading system, incorporating unmanned aerial vehicle (UAV) as an aid in processing sensor data for marine smart grids. Reference [4] delves into the optimization strategy for deploying smart grid containers supported by MEC, innovatively prioritizing applications based on their importance and constructing an optimization problem model that minimizes latency. Although new energy generation technology has been widely used in the low-carbon transformation process of power systems due to its environmental friendliness, its own randomness and volatility pose challenges to the scheduling of power systems.

In recent years, the Digital Twin (DT) [5-6] technology has attracted widespread attention from both academic and industrial circles at home and abroad. It is a digital replica of physical entities that maps real physical entities and environments in real time to virtual space. Reference [7] uses DT to create a virtual model corresponding to the physical power grid system, which reflects the real-time operation status and changes of the power grid, thereby achieving real-time monitoring and early warning of power grid safety. Reference [8] proposed a new method for operation monitoring and resource scheduling based on the DT model of the power grid. This method utilizes virtualization and visualization of power grid equipment status parameters, and utilizes high-performance data collection and transmission technology

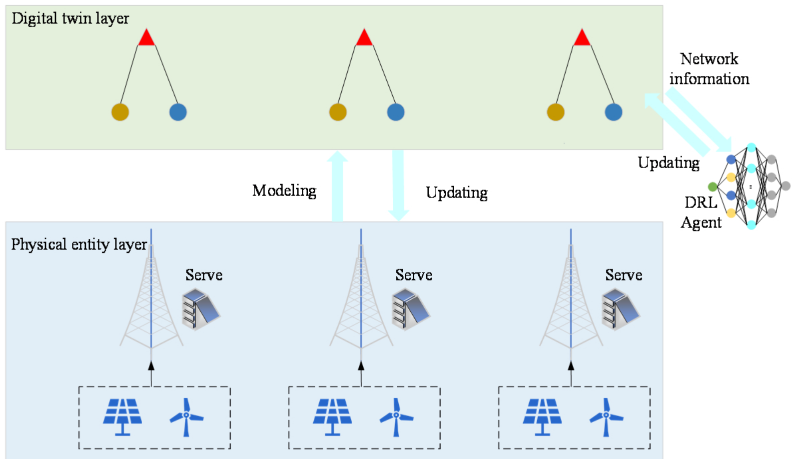

Figure 1: DT-Assisted Renewable Energy Power System.

to push real-time power grid status information to management departments.

Due to the volatility and randomness of new energy generation, we need to respond promptly to terminal equipment. Therefore, we utilize DT to assist the new energy power system, which can reflect the real-time operation status and changes of the power grid. However, existing scheduling algorithms cannot cooperate with DT for real-time scheduling. Therefore, we propose an online scheduling strategy for new energy power systems based on a digital twin model, which utilizes an on-policy strategy in conjunction with DT to achieve online real-time scheduling.

1. We propose a on online scheduling strategy for new energy power systems based on DT model, which minimizes the total energy consumption of the system by jointly optimizing the offloading strategy of smart terminals and the allocation of computational resources on edge servers.

2. To address the issues of volatility and randomness in new energy generation, we utilize DT to assist the new energy power system, which can reflect the real-time operational status and changes of the power grid.

3. Given the inability of existing resource scheduling algorithms to achieve online real-time scheduling, we propose leveraging an on-policy reinforcement learning algorithm in conjunction with DT for real-time offloading and resource scheduling.

2 System Model and Problem Formulation

2.1 System Model

Within the renewable energy power system, K smart terminals generate computational tasks, each of which is independent from the others. During a given cycle T, there are N time slots, each identified by a unique index n. Within each time slot, the smart terminals can choose to offload their computational tasks to an edge server or execute them locally. Each smart terminal is located at a specific coordinate $(x_k, y_k), \forall k \in \{1, 2, \ldots, K\}$. There are M edge servers available within the system, with the m-th edge server identified by the index m. Each edge server is equipped to provide offloading services and is positioned at a fixed coordinate $(x_m, y_m), \forall m \in \{1, 2, \ldots, M\}$. As shown in Figure 1, this paper considers a two-layer DTEN model, comprising the physical entity layer and the DT layer.

Within this framework, the physical entity layer is collectively composed of K smart terminals and M edge servers, while the DT layer encompasses the digital representations of all these physical entities as well as the communication environment that interconnects them. The smart terminals in the physical entity layer transmit their current operational states in real-time to the DT system. The DT system then constructs a virtual model of the renewable energy power system based on the data from the physical entity layer. Within this model, simulations are performed to optimize the offloading strategies for the smart terminals.

1) DT model. We consider two types of DTs: terminal devices and edge servers. As DTs reproduce the operation of physical entities, maintaining a massive number of devices will consume a lot of resources. In order to ensure the smooth operation of the DT network, all DTs are built on edge servers. In time slot n, the DT of the computing entity with respect to its computing frequency can be represented as:

$$\mathrm{DT}_i^n = \left\{ f_i^n, \tilde{f}_i^n \right\} \tag{1}$$

Among them, f_i^n is the estimated value of DT for the actual calculated frequency, $\tilde{f}_i^n$ is the error between the actual calculated frequency and the DT estimation value. Due to the time delay error in data interaction between DT and real devices, DT sometimes cannot accurately reflect the true state of the device. However, the error between the actual calculated time delay and DT estimated time delay can be obtained in advance.

Assume that the size of the task data in time slot n is D^n and the computation period required to calculate unit bit data is C^n, the error between the actual computation delay and DT estimation delay can be expressed as:

$$\Delta \mathrm{T} = \frac{D^n C^n}{\left(f_i^n + \tilde{f}_i^n\right)} - \frac{D^n C^n}{f_i^n} = \frac{-D^n C^n \tilde{f}_k^n}{\left(f_i^n + \tilde{f}_i^n\right) f_i^n} \tag{2}$$

2) Offload the computing model. In time n, each smart terminal k randomly generates an indivisible computation task denoted as:

$$M_k^n = \left\{D_k^n, C_k^n, T_k^n\right\} \tag{3}$$

where D_k^n is the size of the task data, C_k^n is the computation cycle required to compute unit bit data, and t is the task latency limit.

To avoid interference between different smart terminals during task offloading, we use time-division multiple access technology. Smart terminals offload tasks during time slots $\Delta\sigma = T/N$. Smart terminals determine the task offloading strategy based on the offloading factor of DT:

$$a_{k,m}^n = \{0, 1\} \tag{4}$$

where $a_{k,m}^n$ =0 indicates that the task generated by smart terminal is computed locally in time slot n, and $a_{k,m}^n$ =1 indicates that the task generated by smart terminal is computed on the edge server. And the offloading factor must meet the following requirements:

$$\sum_{m=1}^{M} a_{k,m}^n \leq 1 \tag{5}$$

When $a_{k,m}^n$ =0, the smart terminal decides to perform local computing in time slot n, and its computing latency and energy consumption are:

$$T_k^n = \frac{D_k^n C_k^n}{f_k^n} \tag{6}$$

$$E_k^n = k_1\left(f_k^n\right)^2 D_k^n C_k^n \tag{7}$$

where f_k^n is the local computing capability of smart terminal k in time slot n, and k_1 is the power consumption factor. Due to the limited computing power of each smart terminal, the computing power of smart terminal k in time slot n is:

$$0 \leq f_k^n \leq f_k^{\max} \tag{8}$$

where $f_k^{\max}$ is the maximum local computing power of each smart terminal.

According to formula (2), we can obtain the locally calculated real computation delay and the estimated delay error of the DT. Therefore, the total latency of smart terminal processing task on local device is:

$$T_k^{n,all} = T_k^n + \Delta\mathrm{T} \tag{9}$$

Considering the reality, the channel gain from the k-th smart terminal to the m-th edge server in time slot n is:

$$h_{k,m}^n = a_0\left(\mathrm{d}_{k,m}^n\right)^{-2} \tag{10}$$

where a_0 is the channel power gain at a reference distance of 1m and $\mathrm{d}_{k,m}^n$ is the distance between the k-th smart terminal and the m-th edge server.

If smart terminal k is served by edge server m in time slot n, i.e. $a_{k,m}^n$ =1, then the corresponding signal-to-noise ratio of edge server m in time slot n is represented as:

$$r_{k,m}^n = \frac{p_k h_{k,m}^n}{\sum_{j=1, j\neq k}^{K} p_j h_{j,m}^n + \sigma^2} \tag{11}$$

where p_k represents the transmission power of smart terminal k, and σ^2 is the power of the UAV receiver's additive Gaussian white noise. $\sum\limits_{j=1,j\neq k}^{K} p_j h_{j,m}^n$ shows the interference caused by the transmission of all other intelligent terminals in time slot n. In this case, the offloading rate of smart terminal k in time slot n is:

$$R_{k,m}^n = \sum_{m=1}^{M} a_{k,m}^n \log\left(1 + r_{k,m}^n\right) \tag{12}$$

If the smart terminal k offloads the task to the edge server m in time slot n, the offloading delay and energy consumption are:

$$T_k^{n,p} = \frac{D_k^n}{R_{k,m}^n} \tag{13}$$

$$E_k^{n,p} = p_k T_k^{n,p} \tag{14}$$

After the smart terminal k offloads the task to the edge server m, the edge server m needs to perform calculations for it, and its computing latency and energy consumption is:

$$T_m^n = \frac{D_k^n C_k^n}{f_m^n} \tag{15}$$

$$E_m^n = k_2\left(f_m^n\right)^2 D_k^n C_k^n \tag{16}$$

where f_m^n is the computing capability of edge server m, and k_2 is the power consumption factor. Due to the limited computing power of each edge server, the computing power of edge server m is:

$$0 \leq f_m^n \leq f_m^{\max} \tag{17}$$

where $f_m^{\max}$ is the maximum computing power of each edge server.

2.2 Problem Formulation

This paper considers the delay constraint and reasonably allocates offloading tasks to edge servers, while optimizing the allocation of computing resources to minimize the total energy consumption of the network. Defining offloading strategy $A = \{a_{k,m}^n, \forall k \in K, m \in M, n \in N\}$ and Computing resource $\{F = \{f_k^{\mathrm{n}}, f_m^{\mathrm{n}}, \forall k \in K, m \in M, n \in N\}$, the optimization problem is modeled as:

$$\mathbf{P}: \min_{A,F} \sum_{n=1}^{N} a_{k,m}^n E_k^n + a_{k,m}^n E_m^n + a_{k,m}^n E_k^{n,p} \tag{18}$$

$$\text{s.t.} \quad (4), (5), (8), (17), \tag{19}$$

$$\max\left(a_{k,m}^n T_k^{n,\text{all}}, a_{k,m}^n T_{k,m}^{n,\text{all}}\right) \leq T_k^n, \tag{20}$$

3 PROPOSED SOLUTIONS

Due to the non-convexity of the objective function, the dynamic nature of the scene, and the diversity of tasks, traditional offline optimization methods are difficult to solve it. In order to achieve online real-time decision-making, this paper uses the DRL method to solve the problem.

In this scenario, the edge servers don't require any prior information about the environment and can only obtain causal information from the environmental state. Therefore, the transition probability in this model is unknown and can be modeled as a MDP without a model or transition probability. In MDP, the agent continuously interacts with the dynamic environment to optimize its own strategy. For example, at time step n, when the environment is in a certain state s_n, the agent performs action a_n, and the environment transitions to any feasible successor state s_{n+1} with a certain probability, the agent receives a reward r_n, and then n=n+1. The intelligent agent adjusts its own strategy by observing the state s_{n+1} and the reward r_n, in order to maximize the accumulated reward. During this process, the state space, action space, and reward function are three key elements.

1) State space

$$s_n = \left\{D_k^n, C_k^n\right\} \tag{21}$$

2) Action space

$$a_n = \left\{a_{k,m}^n, f_k^n, f_m^n\right\} \tag{22}$$

3) Reward function

$$r_n = a_{k,m}^n E_k^n + a_{k,m}^n E_m^n + a_{k,m}^n E_k^{n,p} \tag{23}$$

Due to the continuity of the state space and action space mentioned above, we adopt the Proximal Policy Optimization (PPO) algorithm to minimize system energy consumption. The PPO algorithm adopts the Actor-Critic structure, and the Actor network is divided into two parts: old and new, corresponding to parameters θ and old θ_{old}, respectively. The Critic network parameter is ξ. The Actor network outputs action a_n based on state s_n and interacts with the environment; The Critic network calculates the state value $V^{\xi}(s_n)$ based on state information, which can be expressed as:

$$V^{\xi}\left(s_n\right) = \mathbb{E}_{s_n,a_n}\left[\sum_{l=0}^{\infty} \gamma^l \mathcal{R}\left(a_{n+1}|s_{n+1}\right)\right] \tag{24}$$

where γ is the discount factor, $\mathbb{E}[\cdot]$ represents the expected value, and $\mathcal{R}(\cdot)$ represents the reward function. To evaluate the performance of action a_n, the algorithm introduces an advantage function $\hat{A}(s_n)$, which is:

$$\hat{A}\left(s_n\right) = \sum_{l=0}^{\infty} (\gamma\eta)^l \left(r_n + \gamma V\left(s_{n+1}\right) - V\left(s_n\right)\right) \tag{25}$$

where $0 \leq \eta \leq 1$ is the GAE (general advantage estimation) factor. Subsequently, the objective functions of the Actor network θ and the Critic network ξ are calculated, which can be expressed as:

$$L^{actor}\left(\theta\right) = \mathbb{E}_{\pi_\theta}\{\min[\frac{\pi_\theta\left(a_n|s_n\right)}{\pi_{\theta_{\text{old}}}\left(a_n|s_n\right)}\hat{A}\left(s_n\right),$$

$$clip\left(\frac{\pi_\theta\left(a_n|s_n\right)}{\pi_{\theta_{\text{old}}}\left(a_n|s_n\right)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}\left(s_n\right)]\} \tag{26}$$

$$L^{critic}\left(\xi\right) = \left[V^{\xi}\left(s_{n+1}\right) - V^{\xi}\left(s_n\right)\right]^2 \tag{27}$$

where $\pi_\theta(\cdot)$ and $\pi_{\theta_{\text{old}}}(\cdot)$ represent new and old policy functions, with ϵ being the truncation parameter. NUMERICAL RESULTS

This section provides data simulation to verify the impact of the new energy power system online scheduling strategy based on the DT model proposed in this paper on the total energy consumption of the system. The TensorFlow framework is used to build a simulation environment and analyze the performance of the proposed scheme. Consider a ground square area with an area of 500 m and 500 m, where smart terminals are randomly distributed within the area.

The convergence of the scheme is shown in Figure 2. As the number of training steps increases, the reward of the proposed scheme gradually decreases. The reinforcement learning agent can significantly improve the reward value for each training step, which confirms the effectiveness of the scheme in computing offloading. From Figure 2, it can be observed that the algorithm exhibits significant volatility between 0-100 rounds, due to its exploratory nature in the initial stage. As the number of turns increases, the exploratory nature gradually decreases, leading to a decrease in the volatility of the algorithm. When the number of turns reaches 300, the algorithm begins to converge. This indicates that as the algorithm goes through a certain number of rounds, it gradually stabilizes and converges to a better solution.

The relationship between total network energy consumption and the number of smart terminals is shown in Figure 3. The simulation result shows that as the number of smart terminals increases, the energy consumption of the network shows an upward trend. This is because as the number of smart terminals increases, the time slots allocated to each terminal device become smaller, so the terminal devices need to transmit tasks with higher power, resulting in an increase in the total energy consumption of users. When the number of smart terminals is the same, the total energy consumption of the algorithm proposed in this article is always lower than the three benchmark schemes. This is because our algorithm adopts an on-policy strategy to optimize offloading strategy and computational resource allocation.

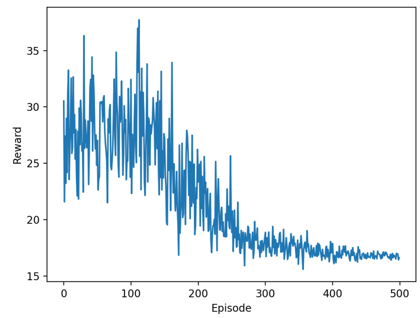

Figure 2: Reward convergence of algorithm.

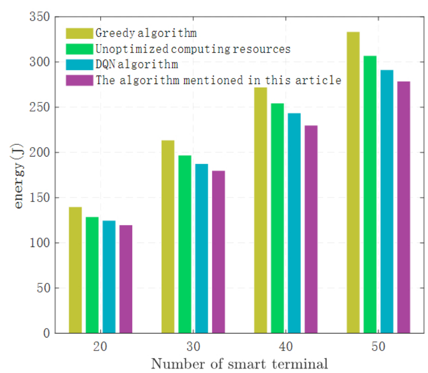

Figure 3: Comparison of energy consumption of smart terminals under different numbers of users.

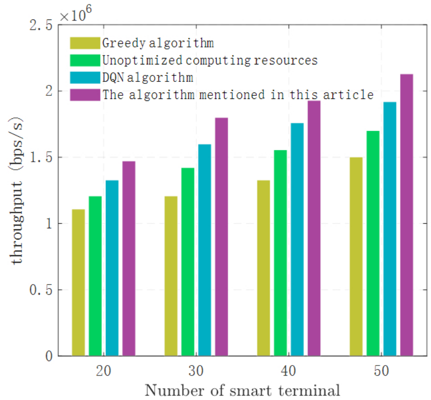

Figure 4: Comparison of throughput of smart terminals under different numbers of users.

According to Figure 4, as the number of users increases, the throughput of various algorithms gradually increases. It is particularly noteworthy that the proposed algorithm is the best in this trend. Due to the possibility of changing resource allocation in smart terminals, our algorithm adjusts in real-time through online strategies to quickly adapt to different user numbers and resource allocation situations.

4 Conclusion

This paper investigates a task offloading scheme assisted by the DTEN. With the objective of minimizing the total energy consumption of smart terminals and edge servers, it achieves intelligent task offloading from smart terminals by jointly optimizing user offloading strategies and computational resource allocation. Furthermore, a PPO-based task offloading algorithm is proposed. Simulation results demonstrate that the DRL-based optimization algorithm is capable of rapidly obtaining optimal solutions. In future work, we intend to detail how to balance the computing load of edge servers and how to ensure the fairness and efficiency of resource allocation.

References

[1] Adrah C M, Palma D, Kure Ø, *et al.* Deploying 5G architecture for protection systems in smart distribution grids[C]//2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, **2022**: 1-5.

[2] Shakhov V, Koo I. On Security Supporting in Edge Computing-Based Smart Grid[C]//2023 6th International Scientific and Technical Conference on Relay Protection and Automation (RPA). IEEE, **2023**, 1: 1-9.

[3] Xu R, Lin F, Shao W, *et al.* A Union MEC-UAV Computation Offloading System Applied to Wireless Sensor Network for Marine Renewable Energy Smart Grids[C]//2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT). IEEE, **2023**: 829-834.

[4] Jiang J, Xin P, Wang Y and Zhang .Y, A Priority-based Online Container Placement Strategy for 5G MEC Cloud enabled Smart Grid[C]//2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), **2022**: 59-64.

[5] Sun W, Zhang H, Wang R, *et al.* Reducing offloading latency for digital twin edge networks in 6G[J]. IEEE Transactions on Vehicular Technology, **2020**, 69(10): 12240-12251.

[6] Sun L, Shang Y, Ma Z, *et al.* Power Distribution Safety Assessment Based on the Digital Twin[C]//2023 4th International Conference on Smart Grid and Energy Engineering (SGEE). IEEE, **2023**: 71-74.

[7] Qiao J, Yu H, He Z, *et al.* Research and Implementation of Operation Monitoring and Resource Scheduling Methods Based on the Digital Twin Model of the Power Grid[C]//2023 5th International Conference on Applied Machine Learning (ICAML). IEEE, **2023**: 524-528.

[8] Soares A A Z, Lopes Y, Fernandes N C, *et al.* Enabling Digital Twins for Smart Substation Networks[C]//2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, **2023**: 1-7.