Research Article

HVAC Optimal Control with the Multistep-Actor Critic Algorithm in Large Action Spaces

Zetian Huang, Jianping Chen, Qiming Fu, Hongjie Wu, You Lu and Zhen Gao

1Institute of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China
2Jiangsu Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou 215009, China
3Suzhou Key Laboratory of Mobile Network Technology and Application, Suzhou University of Science and Technology, Suzhou, Jiangsu 215009, China
4Faculty of Engineering, McMaster University, Hamilton L8S 0A3, Canada

Correspondence should be addressed to Jianping Chen; alanjpchen@yahoo.com and Qiming Fu; fqm_1@126.com

Received 17 August 2020; Revised 26 September 2020; Accepted 5 October 2020; Published 27 October 2020

Academic Editor: Neale R. Smith

Copyright © 2020 Zetian Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an optimization method, named as the Multistep-Actor Critic (MAC) algorithm, which uses the value-network and the action-network, where the action-network is based on the deep Q-network (DQN). The proposed method is intended to solve the problem of energy conservation optimization of heating, ventilating, and air-conditioning (HVAC) system in a large action space, principally for the cases with high computation and convergence time. The method employs the multistep action-network and search tree to generate the original state and then selects the optimal state based on the value-network for the original and the adjacent states. The results from the application of the MAC algorithm to a simulation problem on the TRNSYS system, where the simulation problem is referring to a real supertall building in Hong Kong, have shown that the proposed MAC algorithm balances control actions between different HVAC subsystems. Further, it substantially saves the computational time while maintaining a good energy conservation performance.

1. Introduction

The building sector, on a global scale, accounts for nearly 40% of the total energy consumption and contributes 30% of the total CO₂ emissions [1]. Buildings account for nearly 20.6% of the total energy consumption in China and contribute 19.4% of the total CO₂ emissions [2]. By 2018, the global energy consumption grew almost twice as fast as the average since 2010, with about 80% of that proceeding from fossil fuels [3]. HVAC is considered the largest energy consumption system in a building [4] since more than 50% of the building energy consumption is from it. Therefore, improving the efficiency of HVAC system control strategy can enhance performance [5]. The energy consumption of HVAC is directly related to the setting values of its systems and subsystems [6], besides the building type and climate that also affect the consumption.

Various optimization strategies have been used [7] to reduce the energy consumption of the HVAC system. It is far more sustainable and cost effective to improve the control algorithms with more efficient modern technologies to achieve higher efficiency than replacing the HVAC equipment [8]. Research has revealed that appropriate control strategies of the HVAC system can maximize the overall operating efficiency while maintaining satisfactory thermal comfort [9], whereas unreasonable control strategies can result in excessive energy consumption. However, most existing HVAC systems have been optimized locally leading to the moderate performance [10]. Concurrently, one disadvantage of these methods, such as dynamic programming,
game theory, and Markov Decision Processes, is that they must be recalculated when the system needs to be optimized or its underlying assumptions have changed. This can be time consuming in complex and changing environment [11]. Reinforcement learning and deep reinforcement learning have the characteristics of a free model, self-learning, and online learning that can provide faster solutions and better adaptation for the changing environments than the traditional method. Moreover, they are the most suitable for alleviating the cost minimization problem with the ability to learn the optimal behavior, despite the global optimum being unknown [12].

Reinforcement learning is a part of machine learning, which emphasizes how an agent act based on the environment to maximize the expected benefits [13]. The agent learns by trial and error and finally approaches the optimal decision [14]. Studies have shown that reinforcement learning is able to solve stochastic optimal control problems [15] and an energy consumption scheduling problem [16] with dynamic pricing [17]. The combination of decision-making ability of the reinforcement learning and the perception ability of deep learning engenders a novel method termed as deep reinforcement learning. Deep reinforcement learning has overcome the challenge of learning from high dimensional state inputs, by taking advantage of the end-to-end learning capability of deep neural networks [18]. Deep reinforcement learning has been successfully applied to games [19], robotics [20], and natural language processing (NLP) [21]. Concurrently, several papers have reported on the applications of reinforcement learning and deep reinforcement learning in the HVAC system.

We propose a novel optimization method for the energy conservation optimization of HVAC systems based on the DQN and value-network balancing performance and the computation time, called the MAC algorithm. The objective was to determine the ability of MAC to solve the problem of large state-action pair convergence caused by control actions, while maintaining better energy performance comparison. The HVAC system in a superhigh building in the subtropical region has been considered; besides, based on this system, various forms of the proposed method for optimizing HVAC systems have been tested and evaluated.

The rest of the paper is organized as follows. Section 2 introduces related optimization methods of HVAC. Section 3 introduces the accurate modeling in HVAC system and system formulation. The proposed optimization method has been expounded step-by-step in Section 4. Various forms of the proposed method for optimizing HVAC systems are tested and evaluated in Section 5. Lastly, the conclusions are given in Section 6.

2. Related Work

Manifold problems in the HVAC system can be transformed into decision-making tasks. Various optimization methods proposed for HVAC systems in recent decades are listed in Table 1.

The first category includes the traditional mathematical optimization methods, such as the Newton–Raphson method [22] and interior point method [23]. These methods generally have the advantage of rigorous mathematical model and tight logic. However, explicit objective function expressions in those methods are difficult to abstract in a myriad of actual optimization decision scenarios; hence, those methods are not viable.

The second one includes the heuristic methods, such as genetic algorithm (GA) [24], ant colony optimization [25], and particle swarm optimization [26]. These methods are suitable for almost all optimization problems and have good performance, especially for nonconvex optimization. However, the heuristic methods usually get the local optimal solution, and these methods are less robust and devoid of a rigorous mathematical proof [27].

Another category includes the machine learning methods, such as reinforcement learning [28–30] and deep reinforcement learning [31]. As the most popular technology in recent years, these methods are not restricted by the exact objective function like the traditional mathematical optimization methods, but they are more robust with more stable convergence results than the heuristic methods. However, an infinite number of state-action pairs can exist in buildings owing to the outdoor environment and thermal characteristics of spaces, as the HVAC systems are time varying. Most of the existing machine learning methods are limited to the local control of one subsystem or the low precision control of all subsystems, to avoid the extensive computation time which can be used for updating the policy [32].

Furthermore, there are several alternative optimization methods for HVAC systems. Sun et al. [33] have proposed a multiplexed optimization scheme for complex air-conditioning systems. He et al. [34] have proposed a computational intelligence algorithm based on the performance optimization for the HVAC. Wang et al. [35] have proposed a new method called event-driven optimal control for HVAC systems, in which optimization actions are triggered by events instead of a clock. Simone Baldi et al. [36] have proposed a holistic framework for HVAC systems with energy-aware and comfort-driven maintenance. Baldi et al. [37] have proposed a switched self-tuning approach to solve a multiple-mode feedback-based optimal control problem for HVAC systems.

However, all of these methods have the drawback of highly increasing computation time with respect to the increase of search space. For a vast search space, the increasing computation time makes most of these optimization algorithms unworkable. According to relevant studies [34], the average computational time for the Strength Pareto Evolutionary Algorithm is nearly 900 seconds, for an optimization time interval of 15 minutes. Hence, the computation time is unacceptably huge. In the online optimal control of HVAC systems, excessively long computational time leads to the degradation of optimal control performance owing to the ensuing delay in the control response.
3. System Description and Formulation

3.1. Complex HVAC System. A superhigh building in the subtropical region has been selected as the relevant case. The HVAC system of the building is all-electric cooling without thermal storage (with primary and secondary chilled water loops), which is a commonly used commercial system. Based on the reference building, a simulation model has been established, which is used by the TRNSYS software. The configuration of the HVAC system of the building includes a cooling water loop, two chilled water loops (primary loop before heat exchangers and secondary loop after heat exchanger), and an air distribution subsystem. In the simulation model, the dynamic performance of the heat exchangers and the AHUs are simulated by Type 699 and Type 508a. The time delay model of the chilled water and the cooling water is simulated by Type 661. The cooling load and the weather information are read into TRNSYS by Type 9e. MATLAB is used to simulate the control algorithm of subsystems in HVAC and optimization method by the component Type 155 of TRNSYS. The system simulation model, which is verified, is shown in Figure 1, and the accuracy of the model is found to be acceptable.

The structure of a complex AC system in this case study is shown in Figure 2, which consists of cooling towers, chillers, heat exchangers, air-handling units, and zones. Cooling towers deliver the condenser water to the chillers. The heat exchanger has been used for delivering the cold cooling water, chilled water, primary loop, secondary loop, and zones. The specific settings are shown in Table 2.

3.2. System Formulation. The optimization of the HVAC system is to seek optimal settings of all subsystems, viz., the cooling of water supply temperature, the chilled water supply temperature in the heat exchangers, and the supply air temperature. It is well-known that these set points have a significant impact on the energy consumption of the whole system, and their optimal values vary with the operational conditions by many studies.

\[ T_{cw,low} \leq T_{cw} \leq T_{cw,up}, \]
\[ T_{chw,prm,low} \leq T_{chw,prm} \leq T_{chw,prm,up}, \]
\[ T_{chw,sec,low} \leq T_{chw,sec} \leq T_{chw,sec,up}, \]
\[ T_{sa,low} \leq T_{sa} \leq T_{sa,up}, \]
\[ |T_{c(k+1)} - T_{ch}| \leq \Delta T_{Thres}, \]
\[ T_{chw,prm,e} + \Delta T_{min} \leq \Delta T_{chw,sec,e}. \]

The range of these set points is subjected to (2)–(5), which take the operations or other types of constraints into account. Concurrently, two additional constraints are taken in (6) and (7), which consider the account system stability and the minimal temperature difference between the primary and secondary sides of the chilled water loops. The specific settings are shown in Table 2.

4. Multistep-Actor Critic Algorithm

4.1. Multistep-Actor. We propose a basic action to solve the problem of high computation and convergence time effec-
tuated by large action space. By combining the transition model and search tree, the selection of multistep basic action is employed to replace the selection of the optimal action from all the actions. The specific contents are given as follows.

4.1.1. Constructing DQN Based on the Basic Action and Training Model. Map the states within a continuous space in HVAC output to the discrete state set (S), and then the discrete action (A) abstract from the discrete state set. Under the constraint of the HVAC system, the number of the controlled spaces in the HVAC system is N. The traditional DQN is constructed by all of the controlled spaces in the HVAC system. However, occasionally it does not converge in finite time when N is large owing to the extensive computation time for converging the policy.

To address these issues, the structure of DQN in our architecture has been built by basic actions (a), which

| Table 1: Methods classification. |
|-------------------------------|
| Traditional mathematical optimization methods | [22, 23] |
| Heuristic methods | [24–27] |
| Machine learning methods | [28–32] |
| Several alternative methods | [33–37] |
Figure 1: System simulation model (screenshot).

Figure 2: Schematic of the complex air-conditioning system.

Table 2: Factors.

| Factor name | Value (°C) |
|-------------|------------|
| $T_{cw,low}$, $T_{cw,up}$ | [20, 28] | [28, 35] | [24, 30] |
| $T_{chp,low}$, $T_{chp,up}$ | [5, 8] | [5, 8] | [5, 8] |
| $T_{chsec,low}$, $T_{chsec,up}$ | [6.5, 11.5] | [6.5, 11.5] | [6.5, 11.5] |
| $T_{sa,low}$, $T_{sa,up}$ | [12, 18] | [12, 18] | [12, 18] |
| $\Delta T_{\text{Thres}}$ | 1 | 1 | 1 |
contain \( \text{card}A \) actions. Concurrently, building the value-network and transition model is also based on the basic actions. The basic actions can be defined as follows: if action space can be represented by the combination of itself with a subset of it, then call this subset basic action. The choice of basic actions is based on the principles given below:

\[
A_i \in A, \quad \text{for } i = 1, \ldots, \text{card}A,
\]

\[
A_i = \sum_{j=1}^{\text{card}A} c_j \cdot a_j \cdot c_j \geq 0, a_j \in a.
\]  

DQN, value-network, and transition model have been built by neural network and replay memory and trained in the simulation model. At each optimization stage, we could get \((s, E, s', E')\) and place it to replay memory, where \((s, E, s', E')\) have been from the simulation model and “\(a\)” represents the action with abstraction from \(s\) to \(s\). Among them, \(s\) is the current state and \(E\) is the energy consumption of the current state. The factors in \(s\), are the next setting state, the factors in are the next actual state, and \(E'\) is the energy consumption of the next state.

Transition model is trained by \((s, a, s')\) from replay memory, where \(s\) and \(a\) are input and \(s'\) is output. Value-network is trained by \((s, E, s', E')\) from replay memory, where \(s\) is the input and \(E\) is the output. Concurrently, \(s'\) is the input and \(E'\) the output. DQN is trained by a set of \((s, a, r, s')\) from replay memory, where \(r = E' - E\). The training process of transition model and value-network is the same as that of traditional neural network. The training process of DQN-based HVAC control algorithm is shown in Algorithm 1 where \(t \mod k\) is a control time step.

4.1.2. Multistep-Actor Based on the Transition Model and Search Tree. Under the constraint of the HVAC system, we have constructed the search tree by a transition model and the current state to get leaf nodes, where action \(a_m\) is chosen by DQN based on Section 4.1.1. DQN is used to parameterize the Q function and provide action.

Search tree chooses the next state \((s_{im})\) by \(a_m\). If \(a_m\) is not \(a_j\) \((a_j = [0,0,0,0])\), then put \(s_{im}\) to actor (DQN) to get next action \(a_j\), and search tree chooses next state \((s_{ij})\) by \(a_j\) from actor. Until the action \(a\) from the actor makes state invariability, search tree chooses next state \((S)\) by action \(a_j\) from actor and stops passing. Finally, we get the original state \(\overline{s}\) by search tree, which is decided by the action, transition model, and current state. The process is shown in Figure 3.

4.2. Critic. Critic module is added on the basis of the multistep-actor for further improving the energy saving efficiency and finding the optimal state. The purpose of the critic module is to select the optimal state by comparing the energy consumption value of the original state and its nearby states. The specific contents are as follows.

4.2.1. States Chosen by KNN. This original state from the multistep-actor will likely not be an optimal state, owing to the error of the transition model and the error accumulation of multiple steps. Nevertheless, the optimal state must be close to this original state. We need to map from \(\overline{s}\) to an element in \(S\). We can do this as follows: \(S_k = \text{argmin}_{s \in S} |s - \overline{s}|_2\).

\(S_k\) is a \(k\)-nearest neighbor mapping of the discrete action set \((S)\), which has \(k\) states in \(S\) that are the closest to \(\overline{s}\) by Euclidean distance. In the exact case, this lookup has the same complexity as the argmax in the value-function derived policies described in Section 4.2.2, but each step of evaluation is Euclidean distance, not the full value-function evaluation. This task has been extensively studied in the literature for approximate nearest neighbor and can be searched in a similar manner in logarithmic time. This step is described by the bottom half of Figure 4, where we can see the actor network producing a proto-action, and the \(k\)-nearest neighbors being chosen from the action embedding.

Time complexity of the above algorithm scales linearly with the number of selected actions \((k)\). However, as we will see in practice, increasing \(k\) to a certain limit does not improve the performance. Our approach has a study on the quantity of \(k\), and the results show that it provides significant energy conservation performance gains for the initial increase in \(k\), but quickly makes the additional performance gains negligible.

4.2.2. Critic Selection Based on Value-Network. Depending on how well the state representation is valued, the states with a low value may occasionally sit closest to \(\overline{s}\) even in a part of the space where most of the states have a high state value. Furthermore, in the motion embedding space, certain actions may be adjacent to each other, though in certain states they must be distinguished owing to having similar value. In both of these cases, trivially selecting the closest element to \(\overline{s}\) from the set of states generated previously is not the ideal.

To avoid the selection of these exception states, general improvements have been made to the final emitted state. The second phase of the algorithm, which is described by the top part of Figure 4, refines the choice of the state by selecting the highest-scoring state according to.

As we have demonstrated in Section 4.2, this second part makes our algorithm significantly robust without imperfections in the choice of action representation, and it is essential for our system learning in certain domains. The size of the generated action set, \(k\), is task specific and allows for an explicit trade-off between the policy quality and speed.
Figure 3: Multistep process.

Figure 4: Multistep-Actor Critic (MAC).
4.3. Multistep-Actor Critic (MAC) Algorithm. We propose a novel optimization method, termed the Multistep-Actor Critic (MAC), to solve the optimization problem of the HVAC energy consumption. The subsystems of the HVAC system interact with each other. The decrease in the energy consumption in one subsystem may lead to an increase in the energy consumption in another subsystem. Hence, the controllable decision factors in the HVAC systems are considered as a whole. The details are as follows.

First, to find basic actions in the action set and construct DQN, the value-network and the transition model were based on basic actions. Second, to train DQN, the value-network and the transition model were based on basic actions by the simulation model in the TRNSYS online at the same time. Third, at each optimization stage, i.e., every 30 minutes, the search tree has been constructed by the transition model to get the leaf nodes based on the current state, and we placed the current state in DQN to get the best action that yields the lowest energy consumption. Thereafter, the choice of the next state is based on the search tree, until it is stabilized. Fourth, the original state has been constructed by step 3, and k has been chosen from the states close to the original state by KNN. Finally, the setting value of the optimal energy consumption has been obtained by finding the best state that yields the lowest energy consumption in those k states, based on the value-network.

The Multistep-Actor Critic algorithm is described fully in Algorithm 2 and illustrated in Figure 4.

5. Experimental Results and Analysis

5.1. MAC

5.1.1. Building DQN. We consider the reliability and robustness of the data in this work. Based on this, the cooling water supply temperature, chilled water temperature, chilled water supply temperature in the heat exchangers, supply air temperature, cooling load, dry bulb temperature, and wet bulb temperature have been selected as the state (s). Among these, the cooling water supply temperature (Tcw), chilled water temperature (Tchw), chilled water supply temperature in the heat exchangers (Tchw_HX), and supply air temperature (Tsa) are the controllable decision factors.

Under the constraint of (6) and (7), the controllable space of the four controllable decision factors has been divided by 0.1. Hence each of the actions contains four elements corresponding to the four controllable decision factors, and a total of 194,481 actions constitute the action set (A). If all of these actions have been reached by basic actions, basic actions could replace those actions. In this case, we have three plans. They are plan 1, where basic actions contained 81 actions by three actions {0.1, −0.1, 0} of the four decision factors; plan 2, where the basic actions contained 625 actions by five actions {0.2, −0.2, 0.1, −0.1, 0} of the four decision factors; and plan 3, where each of the basic actions contained 2401 actions by seven actions {0.3, −0.3, 0.2, −0.2, 0.1, −0.1, 0} of the four decision factors. Finally, the constructed DQN in this paper contains seven neurons in the input layer, which correspond to seven neurons from the state (s). The first, second, and third of the three hidden layers contain 600, 1000, and 600 neurons, respectively, and the number of neurons in the output layer has been based on basic actions.

5.1.2. Training DQN, Value-Network, and the Transition Model. Since the biggest value of the basic actions is 0.3 with the rest being small enough, s′ cannot get near the exact result owing to PID control, and hence s must be approximated by s∗. Hence, we default the next value s′ to a set value s0, which implies that the probability distribution in transition model is 1. The constructed value-network in this paper is radial basis function neural network which contains seven neurons corresponding to the state (s) in the input layer, 50 neurons in the hidden layer, and one neuron corresponding to the energy consumption (E) in the output layer.

Considering the cooling load, dry bulb temperature, wet bulb temperature variation, and time lag in the system, 30 minutes has been set as the time interval for constructing the energy consumption model in this paper. At each optimization stage (every 30 minutes), we obtained (s, E, a, s′, E′) and put it to replay the memory. The value-network has been trained by (s, E, a, s′, E′) from replay memory, and DQN was trained as given in Section 4.1.1, where an episode was 30 minutes with r = E′ − E and C = 2 (every 1 hour) for this case.

5.1.3. System Optimization. At each optimization stage (every 30 minutes), first, the search tree has been constructed by transition model to get leaf nodes based on the current state, and then the current state was put in DQN to get the best action that yields the lowest energy consumption. Thereafter, the next state has been based on the search tree, until the next state was invariable. Second, the original state has been obtained by step 1, and k states are selected close to the original state by KNN. The optimal energy consumption setting value has been obtained by finding the best state that yields the lowest energy consumption in those k states, based on the value-network; hence, V∗ = argmins∈Sk Vθ(s). Finally, we have determined the value of the four controllable decision factors.

5.2. Other Methods

5.2.1. Exhaustive Method (EXM). The exhaustive method (EXM) enumerates all the cases related to the problem. In this experiment, there are four controllable decision factors, viz., Tchw_HX, Tsa, Tcw, and Tchw, which take into account the stability of the HVAC systems (the maximum change value of each controllable decision factor should be...
controlled within 1 degree) and the limit by the response time of HVAC systems (when the range of each decision factor is large, the search time is long). The simulation system does not get the set value in time, resulting in the system crash. Therefore, the range of each decision factor has been defined as \([T - 0.5^\circ C, T + 0.5^\circ C]\), where \(T\) is the initial value of each controllable decision factor. When the range exceeds (6) and (7), the range should be truncated to satisfy the constraint. The four controllable decision factors have formed 14641 states; subsequently, a trained neural network has been employed to obtain the energy consumption prediction of the state. Finally, the point has been selected with the lowest energy consumption.

5.2.2. **DQN for Low Precision Control (DQN-L).** DQN-L is a method which is based on the traditional DQN of the low precision control. The control action of DQN-L is \([-1, -0.5, 0, 0.5, 1]\), which takes into account of the stability of HVAC systems (the maximum change value of each controllable decision factor should be controlled within 1 degree) and the limited response time of HVAC systems. In this experiment, there are four controllable decision factors, viz., \(T_{chw_HX}, T_{sa}, T_{cw},\) and \(T_{chw}\). When the range exceeds that of (6) and (7), the range should be truncated to satisfy the constraint. This method has selected the best control action which had the lowest energy consumption from DQN. Finally, we set the values of the four controllable decision factors to the HVAC system.

5.3. **Results and Analysis.** The experiments focus on the HVAC energy consumption in spring, summer, and autumn, respectively. At each optimization stage (every 30 minutes), all methods have been optimized according to the current environment including the cooling load, dry bulb temperature, wet bulb temperature, and setting values of subsystems. According to each method, the setting values of the cooling water supply temperature, the chilled water supply temperature, the chilled water supply temperature in the heat exchangers, and the supply air temperature have been optimized. With respect to the two indicators, the daily energy saving and the computational time, the performance of all methods was simulated and evaluated. The first indicator has been used to represent the energy performance whereas the second one to reflect the computational complexity of the method.

5.3.1. **Energy Performance Comparison of Different Basic Actions.** The energy performance of the algorithm has been compared with the benchmark case in which the decision factors, including the set points of the cooling water supply temperature, the chilled water supply temperature, the chilled water supply temperature in the heat exchangers, and the supply air temperature, have been set as constants. Benchmark settings are given as:

1. **The spring case:** \(T_{cw} = 26^\circ C,\) \(T_{chw} = 7.5^\circ C,\) \(T_{chw_HX} = 9^\circ C,\) \(T_{sa} = 14^\circ C.\)
2. **The summer case:** \(T_{cw} = 30^\circ C,\) \(T_{chw} = 6^\circ C,\) \(T_{chw_HX} = 7.5^\circ C,\) \(T_{sa} = 15^\circ C.\)

3. **The autumn case:** \(T_{cw} = 28^\circ C,\) \(T_{chw} = 7^\circ C,\) \(T_{chw_HX} = 8.5^\circ C,\) \(T_{sa} = 14.5^\circ C.\)
4. **The HVAC energy consumption in spring, summer, and autumn is shown in Figures 5–7.**

Figures 5–7 show the energy saving effect of EXM, DQN-L, and MAC with different basic actions compared with the benchmark. The \(m\) in m-MAC indicates the number of basic actions. Frequently, EXM, 625-MAC, and 2401-MAC have achieved similar lower energy consumption of the HVAC system, whereas the energy consumption of HVAC system of 81-MAC and DQN-L is higher. Furthermore, EXM is limited by the small search space and cannot always find the optimal setting. In some cases, 625-MAC and 2401-MAC can find a lower power set point than EXM. Table 3 shows the daily energy saving rates of these methods in the spring, summer, and autumn. From Table 3, the highest daily energy
saving rate was achieved by EXM in spring and autumn, but 2401-MAC in summer.

5.3.2. Energy Performance Comparison of Different Nearest States. Although 2401-MAC has higher energy saving effect than 625-MAC, the slight improvement of energy saving effect seems negligible after comprehensive consideration of the training convergence cost; hence, we choose 625-MAC as the next research object. Figures 8–10 show the energy saving effect of EXM, DQN-L, and 625-MAC with different nearest states compared with the benchmark. The k in MAC-k indicates the number of nearest states. More frequently, 625-MAC-1000 and 625-MAC-5000 have almost the same power consumption and could reduce the energy consumption more than 625-MAC. Furthermore, in certain times, 625-MAC-1000 and 625-MAC-5000 can find the lowest power set point. Table 4 lists the daily energy saving rates of these methods in spring, summer, and autumn. From Table 4, 625-MAC-1000 and 625-MAC-5000 have the highest daily energy saving rate for all three seasons.
5.3.3. Computational Time. Table 5 compares the computational time of EXM, DQN-L, and MAC. The time required for the optimization has been taken as the calculation amount for the method to search for the optimal setting based on the CPU i5-4460. The calculations of each algorithm simulate the running load of an optimization (time is the average of one day), and EXM has been used as a benchmark for calculating the computational time comparison. We note that MAC has significantly reduced the computational time more than EXM, thus saving 97.4%,

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{Season} & \textbf{EXM} & \textbf{DQN-L} & \textbf{625-MAC} & \textbf{2401-MAC} \\
\hline
\textbf{Spring} & 0.050 & 0.043 & 0.045 & 0.047 & 0.047 \\
\hline
\textbf{Summer} & 0.124 & 0.097 & 0.099 & 0.124 & 0.127 \\
\hline
\textbf{Autumn} & 0.105 & 0.063 & 0.033 & 0.092 & 0.100 \\
\hline
\end{tabular}
\caption{The daily energy saving rate.}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{Season} & \textbf{EXM} & \textbf{DQN-L} & \textbf{625-MAC-1000} & \textbf{625-MAC-5000} \\
\hline
\textbf{Spring} & 5.83 & 0.11 & 0.15 \\
\hline
\textbf{Summer} & 5.92 & 0.13 & 0.20 \\
\hline
\textbf{Autumn} & 5.75 & 0.10 & 0.15 \\
\hline
\end{tabular}
\caption{The computational time.}
\end{table}
6. Conclusion

We have reviewed the existing HVAC optimization algorithms. We have proposed a novel optimization method called MAC, by optimizing the DQN framework to solve the HVAC optimal control problem in large discrete action spaces, while maintaining better energy performance comparison and low computational time. The results have shown that, in the 2-month simulation experiment, basic actions have an important influence on MAC, and the number of KNN has positive effect on MAC for the choice of a good set point. However, the number of KNN is not linearly related to the energy saving performance. Hitherto, we have chosen 625-MAC-1000 as the best setting. The 625-MAC-1000 can be easily trained for obtaining lower energy consumption points in a short time interval. It can be seen that MAC has a more comprehensive performance in terms of both energy saving and computational time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that appear to influence the work reported in this paper.

Acknowledgments

This work was financially supported by National Natural Science Foundation of China (nos. 61876217, 61876121, 61772357, 61750110519, 61772355, 61702055, 61672371, and 62072324), Primary Research and Development Plan of Jiangsu Province (no. BE2017663), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (no. SJCX20_1100).

References

[1] A. Costa, M. M. Keane, J. I. Torrens, and E. Corry, “Building operation and energy performance: monitoring, analysis and optimisation toolkit,” Applied Energy, vol. 101, pp. 310–316, 2013.

[2] E. Hou, “Overview of China building energy consumption research report,” Building Energy Conservation, vol. 45, no. 12, pp. 131–131, 2017.

[3] G. Energy, “CO₂ status report—the latest trends in energy and emissions in 2018,” International Energy Agency, 2019.

[4] M. W. Ahmad, M. Moursheed, B. Yuce, and Y. Rezgui, “Computational intelligence techniques for HVAC systems: a review,” Building Simulation, vol. 9, no. 4, pp. 359–398, 2016.

[5] J. Zhao, N. Zhu, and Y. Wu, “The analysis of energy consumption of a commercial building in Tianjin, China,” Energy Policy, vol. 37, no. 6, pp. 2092–2097, 2009.

[6] H.-W. Lin and T. Hong, “On variations of space-heating energy use in office buildings,” Applied Energy, vol. 111, pp. 515–528, 2013.

[7] Z. Ma, S. Wang, X. Xu, and F. Xiao, “A supervisory control strategy for building cooling water systems for practical and real time applications,” Energy Conversion and Management, vol. 49, no. 8, pp. 2324–2336, 2008.

[8] M. M. Rahman, M. G. Rasul, and M. M. K. Khan, “Energy conservation measures in an institutional building in subtropical climate in Australia,” Applied Energy, vol. 87, no. 10, pp. 2994–3004, 2010.

[9] M. Fasiuddin and I. Budaiwi, “HVAC system strategies for energy conservation in commercial buildings in Saudi Arabia,” Energy and Buildings, vol. 43, no. 12, pp. 3457–3466, 2011.

[10] N. Nassif, S. Kajl, and R. Sabourin, “Optimization of HVAC control system strategy using two-objective genetic algorithm,” HVAC&R Research, vol. 11, no. 3, pp. 459–486, 2005.

[11] N. Tsang, C. Collin, S. Wu et al., “Autonomous household energy management using deep reinforcement learning,” in Proceedings of the 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Sophia Antipolis, Park, Fr, June 2019.

[12] E. Mocanu, D. C. Mocanu, P. H. Nguyen et al., “On-line building energy optimization using deep reinforcement learning,” IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 3698–3708, 2017.

[13] R. S. Sutton and A. G. Barto, Introduction to Reinforcement Learning, MIT press, Cambridge, MA, USA, 2nd edition, 1998.

[14] K. Mason and S. Grijalva, “A review of reinforcement learning for autonomous building energy management,” Computers & Electrical Engineering, vol. 78, pp. 300–312, 2019.

[15] D. Ernst, M. Glavic, F. Capitanescu et al., “Reinforcement learning versus model predictive control: a comparison on a power system problem,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 517–529, 2008.

[16] D. O’Neill, M. Levorato, A. Goldsmith et al., “Residential demand response using reinforcement learning,” in Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, pp. 409–414, IEEE, Gaithersburg, MD, USA, October 2010.

[17] B. G. Kim, Y. Zhang, M. Van Der Schaar et al., “Dynamic pricing and energy consumption scheduling with reinforcement learning,” IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2187–2198, 2015.

[18] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: a brief survey,” IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.

[19] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, p. 529, 2015.

[20] A. A. Rusu, M. Vecerik, T. Rothörl et al., “Sim-to-real robot learning from pixels with progressive nets,” 2016, https://arxiv.org/abs/1610.04286.

[21] W. Y. Wang, J. Li, and X. He, “Deep reinforcement learning for NLP,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pp. 19–21, MEL, Au, July 2018.
[22] F. Coffele, R. GarciaValle, and E. Acha, “The inclusion of HVDC control modes in a three-phase Newton-Raphson power flow algorithm,” in Proceedings of the 2007 IEEE Lausanne Power Tech, Lake Geneva, GE, CH, July 2007.

[23] A. Kusiak, G. Xu, and Z. Zhang, “Minimization of energy consumption in HVAC systems with data-driven models and an interior-point method,” Energy Conversion and Management, vol. 85, pp. 146–153, 2014.

[24] J. A. Wright, “HVAC optimisation studies: sizing by genetic algorithm,” Building Services Engineering Research and Technology, vol. 17, no. 1, pp. 7–14, 1996.

[25] R. Srikanthapu and U. Vinatha, “Optimized design of collector topology for offshore wind farm based on ant colony optimization with multiple travelling salesman problem,” Journal of Modern Power Systems and Clean Energy, vol. 6, no. 6, pp. 1181–1192, 2018.

[26] A. Kusiak, M. Li, and F. Tang, “Modeling and optimization of HVAC energy consumption,” Applied Energy, vol. 87, no. 10, pp. 3092–3102, 2010.

[27] Z. Zhang, D. Zhang, and R. C. Qiu, “Deep reinforcement learning for power system: an overview,” CSEE Journal of Power and Energy Systems, vol. 6, no. 1, 2019.

[28] E. Barrett and S. Linder, “Autonomous HVAC Control, a Reinforcement Learning approach,” in Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 3–19, Springer, Porto, Portugal, September 2015.

[29] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, “Optimal control of HVAC and window systems for natural ventilation through reinforcement learning,” Energy and Buildings, vol. 169, pp. 195–205, 2018.

[30] H. Berlink and A. H. R. Costa, “Batch Reinforcement Learning for Smart Home Energy Management,” in Proceedings of the International Conference on Artificial Intelligence, AAAI Press, San Diego, CALIF, USA, June 2015.

[31] T. Wei, Y. Wang, and Q. Zhu, “Deep reinforcement learning for building HVAC control,” in Proceedings of the 54th Annual Design Automation Conference, ACM, p. 22, Austin, TX, USA, June 2017.

[32] K. U. Ahn and C. S. Park, “Application of deep Q-networks for model-free optimal control balancing between different HVAC systems,” Science and Technology for the Built Environment, vol. 26, no. 1, pp. 61–74, 2019.

[33] Y. Sun, G. Huang, Z. Li, and S. Wang, “Multiplexed optimization for complex air conditioning systems,” Building and Environment, vol. 65, pp. 99–108, 2013.

[34] X. He, Z. Zhang, and A. Kusiak, “Performance optimization of HVAC systems with computational intelligence algorithms,” Energy and Buildings, vol. 81, pp. 371–380, 2014.

[35] J. Wang, Q.-S. Jia, G. Huang, and Y. Sun, “Event-driven optimal control of central air-conditioning systems: event-space establishment,” Science and Technology for the Built Environment, vol. 24, no. 8, pp. 839–849, 2018.

[36] S. Baldi, F. Zhang, T. Le Quang, P. Endel, and O. Holub, “Passive versus active learning in operation and adaptive maintenance of Heating, ventilation, and air conditioning,” Applied Energy, vol. 252, Article ID 113478, 2019.

[37] S. Baldi, C. D. Korkas, M. Lv, and E. B. Kosmatopoulos, “Automating occupant-building interaction via smart zoning of thermostatic loads: a switched self-tuning approach,” Applied Energy, vol. 231, pp. 1246–1258, 2018.