

Research Article

An Adaptive Deployment Algorithm for IaaS Cloud Virtual Machines Based on Q Learning Mechanism

Shuguang Chen 

Department of Information Engineering, Liuzhou City Vocational College, Liuzhou, 545002 Guangxi, China

Correspondence should be addressed to Shuguang Chen; cjxub5030062@163.com

Received 5 August 2021; Revised 6 September 2021; Accepted 23 September 2021; Published 31 October 2021

Academic Editor: Mu Zhou

Copyright © 2021 Shuguang Chen. 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.

When deploying infrastructure as a service (IaaS) cloud virtual machines using the existing algorithms, the deployment process cannot be simplified, and the algorithm is difficult to be applied. This leads to the problems of high energy consumption, high number of migrations, and high average service-level agreement (SLA) violation rate. In order to solve the above problems, an adaptive deployment algorithm for IaaS cloud virtual machines based on Q learning mechanism is proposed in this research. Based on the deployment principle, the deployment characteristics of the IaaS cloud virtual machines are analyzed. The virtual machine scheduling problem is replaced with the Markov process. The multistep Q learning algorithm is used to schedule the virtual machines based on the Q learning mechanism to complete the adaptive deployment of the IaaS cloud virtual machines. Experimental results show that the proposed algorithm has low energy consumption, small number of migrations, and low average SLA violation rate.

1. Introduction

In recent years, the development scale of service computing model has gradually expanded and has been widely used in various fields. The rapid progress of distributed computing, virtualization, and parallel computing technology has promoted the development of cloud technology [1, 2]. At the service model level, cloud computing is usually divided into software as a service (SaaS), platform as a service (PaaS), and IaaS. IaaS belongs to the basic level in cloud computing services and can provide resources and storage resources in the network. Therefore, it is of great significance to study the algorithm of deploying IaaS cloud virtual machines [3].

Lei and Wang [4] combined the best adaptation algorithm and hierarchical clustering algorithm to optimize the efficiency. On this basis, the local search algorithm was used to optimize the deployment of the virtual machines' locations. The deployment process of the algorithm was relatively cumbersome, leading to the high energy consumption of the algorithm. Liao et al. [5] combined dynamic programming method and Markov models to

predict the price of virtual machines and deployed virtual machines in a cloud environment according to the execution time limit of the workflow. However, this algorithm could not simplify the virtual machine deployment process. The number of migrations of virtual machines was relatively high. Jiliang et al. [6] established a multiservice-desk queuing model. The system energy saving rate and the average stay time of cloud user requests were obtained through the matrix geometry method and virtual birth and death process. The virtual machines were scheduled to realize deployment. The average SLA violation rate of this method was high, and the effectiveness of the algorithm was poor.

In order to solve the problems in the above algorithms, an adaptive IaaS cloud virtual machine deployment algorithm based on the Q learning mechanism is proposed.

The rest of this paper is organized as follows. Section 2 gives the basic issues of virtual machine deployment and an adaptive deployment algorithm for IaaS cloud virtual machines. The experiment is provided in Section 3 to illustrate the effectiveness of the proposed method. Finally, Section 4 concludes this paper.

2. Related Works

2.1. Basic Issues of Virtual Machine Deployment. The problem of the self-adaptive deployment of IaaS cloud virtual machines can be described as follows. n virtual machines constitute a virtual machine set [7, 8], which is denoted as $V = \{v_1, v_2, \dots, v_n\}$, where $v_i = \{p_i, Q_i, r_i\}$ is a virtual machine. p_i is a virtual machine type; $Q_i = \{c_i, m_i, s_i, b_i\}$ is a virtual machine configuration. c_i is the number of CPUs in the virtual machine, m_i is the memory of the virtual machine, b_i represents the expected bandwidth, and s_i represents the hard disk space; r_i represents the amount of data required to build the virtual machines [9]. A host machine set is composed of m host machines, denoted as $H = \{h_1, h_2, \dots, h_m\}$, where $h_j = \{p_j, Q_j, d_j\}$ represents a host machine. p_j is a virtualization platform existing in the host machine, $Q_j = \{C_j, M_j, S_j\}$ means the hardware configuration of the host machine, C_j means the number of CPUs in the host machine, S_j means the hard disk space of the host machine, and M_j means the memory of the host machine. A mapping $f : V \rightarrow H$ is constructed, where $f : v_i \rightarrow h_j$ is the deployment of the virtual machine v_i in the host machine h_j .

2.1.1. Deployment Principles. The virtual machines run on the basis of the host's hardware resources. The main hardware resources include hard disk, memory, and CPU.

After fully testing the applications existing in the virtual machines, the resource usage can be set. However, it is usually difficult to predict the behaviors of IaaS users. In order to improve the quality of service, the deployment of IaaS cloud virtual machines needs to be implemented on the basis of the following principles [10, 11]:

- (a) The number of CPU cores in the host machine for all virtual machines should be lower than the number of cores in the actual host machine for CPU
- (b) The memory of the host machines should be lower than the virtual memory of the virtual machines in the host machines

2.1.2. Deployment Characteristics. Regarding the deployment of virtual machines as an independent issue, its deployment has the following characteristics:

- (a) Resources are limited during task execution. When the traditional method is to schedule independent tasks, any number of tasks can be allocated in a certain resource, but when deploying virtual machines, it is necessary to provide sufficient resources for the host machines [12, 13]
- (b) When the task is scheduled to the resource, the task can be expanded and processed

2.1.3. Problem Description. In the virtual machine deployment algorithm, suitable host machines h_j should be found for all virtual machines v_i to optimize the target. Task sched-

uling goals can generally be divided into the following two categories:

- (a) Optimize the period of tasks by considering the interests of resource users
- (b) Optimize the specific conditions of the resources in the use process by considering the interests of resource providers

Introducing virtualization technology into the server can improve resource utilization. The main task of the IaaS cloud virtual machine deployment algorithm is to use the least host machines to complete the deployment of the virtual machines, regardless of deployment time, bandwidth, and hard disk space. There is a clear corresponding relationship between the type of virtual machines and the manager of the virtual machine of the hosts [14, 15], so the IaaS cloud virtual machine deployment problem can be simplified as follows.

For the virtual machine set $V = \{v_1, v_2, \dots, v_n\}$, $v_i = \{c_i, m_i\}$, and the host machine set $H = \{h_1, h_2, \dots, h_m\}$, $h_j = \{C_j, M_j\}$, we obtain a subset $H' = \{h'_1, h'_2, \dots, h'_{|H'|}\} \subset H$, $h'_j = \{C'_j, M'_j\}$, of the host machine set H and establish a mapping $f : V \rightarrow H'$ that satisfies

$$\left\{ \begin{array}{l} f : V \rightarrow H' \\ \text{s.t.} \quad \sum_{v_j \rightarrow h'_j} c_i \leq C'_j \\ \sum_{v_j \rightarrow h'_j} m_i \leq M'_j \end{array} \right. \quad (1)$$

where C'_j represents the number of CPUs in the host machine after the mapping process and M'_j represents the memory of the host machine after the mapping process.

2.2. An Adaptive Deployment Algorithm for IaaS Cloud Virtual Machines. According to the above analysis of the deployment problem of IaaS cloud virtual machines, the adaptive deployment algorithm for IaaS cloud virtual machines based on the Q learning mechanism uses the Markov decision process to describe the deployment problem of IaaS cloud virtual machines. The essence of IaaS cloud virtual machine deployment is to schedule virtual machines in the cloud environment to achieve adaptive deployment.

2.2.1. Q Learning Method. The Q learning algorithm can solve the Markov decision process, which belongs to the reinforcement learning algorithm [16, 17]. The Markov model can be described with five tuples $\langle S, A, R, P, V \rangle$, where S is the state set, A represents the action set, R is return function, P represents the state transition probability matrix, and V is the model function value.

Markov's strategy can be described by the mapping of the probability distribution in the action space and the state space [18, 19], expressed as $\pi : S \rightarrow \prod(A)$, where $\prod(A)$ is the probability distribution corresponding to the action set A .

In the state-action space, the value function in the strategy can be calculated by

$$Q^\pi(x, a) = R_x(a) + \sum_{y \in S} p_{xy}^a \times V^\pi(y), \quad (2)$$

where $R_x(a)$ represents the return obtained when the action a is transited under the state x , p_{xy}^a represents the corresponding probability when the action a transits from state x to state y , and $V^\pi(y)$ represents the expected total discounted return corresponding to state y under the strategy π .

Using the optimal Q value to obtain the optimal strategy π^* is the principle of the Q learning mechanism, and $Q^*(x, a)$ is used to describe the optimal Q value. If $a^* = \arg \max_a Q^*(x, a)$, the optimal strategy at this time can be described by $\pi^*(x) = a^*$.

The Q learning process is usually composed of two parts, namely, plot and step. The plot is a sequence of steps between the target state and the starting state; the step is the return acquisition and action execution in the determined state.

It is supposed that x_t represents the state corresponding to the agent at the time t , and a_t represents the execution of the action. We observe the return r_t and the next state x_{t+1} during the learning process and update the value function:

$$Q_{t+1}(x, a) = \begin{cases} (1 - \alpha_t)Q_t(x, a) + \alpha_t[r_t + V_t(x_{t+1})], & x = x_t, \\ & a = a_t, \\ Q_t(x, a), & \text{other,} \end{cases} \quad (3)$$

where the parameter $V_t(x_{t+1}) = \max_b \{Q_t(x_{t+1}, b)\}$ and α_t represents the learning rate.

2.2.2. Markov Description of Task Scheduling. When using reinforcement learning to solve the virtual machine scheduling problem, it is necessary to use the Markov model to replace the virtual machine scheduling problem. Markov elements can be selected according to the following principles:

- The state at the next moment is not affected by the historical state and is only related to the current state. The selected state needs to reflect Markov properties [20, 21]
- When choosing an action, it is necessary to ensure that there is a law of state transition, all states can be reached, and the goal of the agent is completed by performing a series of actions
- The selected return function requires the least decision-making steps to make the agent reach the target state

The Markov process is described on the basis of the above principles:

(a) The unit-task matching matrix is used to describe the Markov state. In the virtual machine scheduling process, the state of the matrix is the scheduling's feasible solution, and there is a state transition of reinforcement learning in the feasible solution space [22, 23]

(b) In all states, there are n tasks and n actions. The j th ($1 < j \leq n$) task can be transited to the next design unit through the j th action. At this time, the result obtained by executing the j th action can be expressed by $d_{kj} = 1$, and the parameter k can be calculated by

$$k = \begin{cases} i + 1, & i < m, \\ 1, & i = m. \end{cases} \quad (4)$$

(c) Determine the return value through $r(x, a, x') = 1/t$ (x') in the reinforcement learning

2.2.3. Multistep Q Learning Algorithm. In the Q learning process, the value function $Q(x_t, a_t)$ is updated through $r_t + \gamma V_t(x_{t+1})$, and the return r_t is determined by

$$r_t^{(n)} = r_t + \gamma r_{t+1} + \dots + \gamma^{n-1} r_{t+n-1} + \gamma^n V_{t+n-1}(x_{t+n}), \quad (5)$$

where n represents the number of iterations, γ represents the learning coefficient, and $r_t^{(n)}$ represents the truncated return corresponding to the correction step n at a time t .

The weighted average of the truncated return $r_t^{(n)}$ is used to update, and the learning steps are simplified. The weighted average $r_r^{\lambda, k}$ of truncated return $r_t^{(n)}$ can be calculated by k step time difference $TD(\lambda)$.

$$r_r^{\lambda, k} = (1 - \lambda) \sum_{i=1}^{k-1} \lambda^{i-1} r_t^{(i)} + \lambda^{k-1} r_t^{(k)}, \quad (6)$$

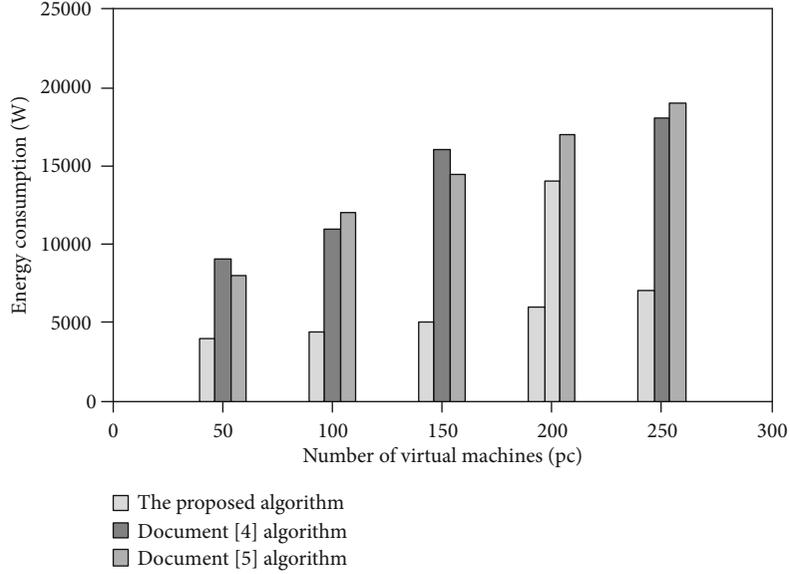
where λ represents the time difference.

$r_t^{(n)}$ is substituted in the weighted average $r_r^{\lambda, k}$ to simplify the processing:

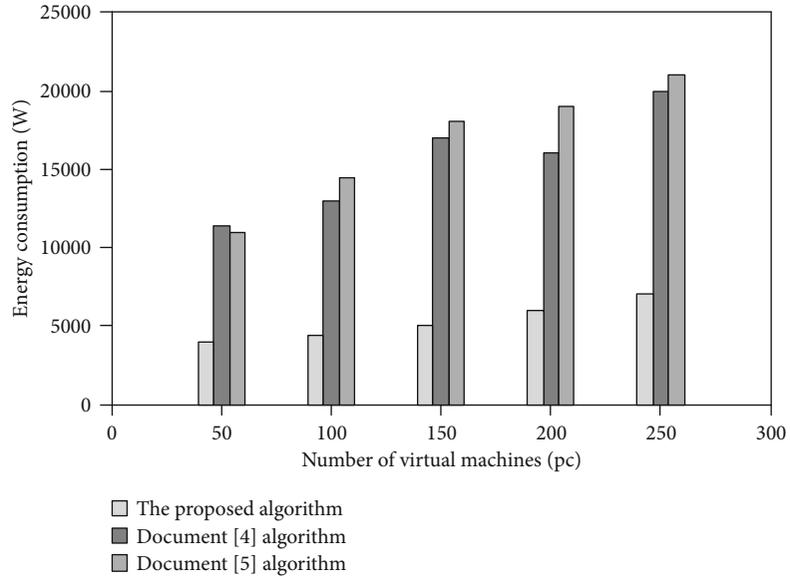
$$r_r^{\lambda, k} = r_t + \gamma V_t(x_{t+1}) + \sum_{i=1}^{k-1} (\lambda \gamma)^i [r_{t+1} + \gamma V_{t+i}(x_{t+i+1}) - V_{t+i-1}(x_{t+i})]. \quad (7)$$

Equation (7) is converted to

$$r_r^{\lambda, k} = r_t + \gamma V_t(x_{t+1}) + \sum_{i=1}^{k-1} (\lambda \gamma)^i [r_{t+1} + \gamma V_{t+i}(x_{t+i+1}) - V_{t+i-1}(x_{t+i})] + \sum_{i=1}^{k-1} (\lambda \gamma)^i [V_{t+i}(x_{t+i}) - V_{t+i-1}(x_{t+i})]. \quad (8)$$



(a)



(b)

FIGURE 1: Comparison of energy consumption by different algorithms. (a) Comparison of energy consumption without network congestion. (b) Comparison of energy consumption with network congestion.

When the learning rate α is low, the adjustment speed of the Q value is slow. At this time, the speed of the V value is also reduced. Let $e_t = r_t + \gamma V_t(x_{t+1}) - V_t(x_t)$, $e'_t = r_t + \gamma V_t(x_{t+1}) - Q_t(x_t, a_t)$, and we have

$$r_t^{\lambda,k} = e'_t + \sum_{i=1}^{k-1} (\lambda\gamma)^i e_{t+i} + Q_t(x_t, a_t). \quad (9)$$

Equation (9) is converted to

$$r_t^{\lambda,k} - Q_t(x_t, a_t) = e'_t + \sum_{i=1}^{k-1} (\lambda\gamma)^i e_{t+i}. \quad (10)$$

Through the above process, the update rules of the Q learning mechanism are obtained:

$$Q_{t+1}(x, a) = \begin{cases} Q_t(x, a) + \alpha_t e'_t + \sum_{i=1}^{k-1} (\lambda\gamma)^i e_{t+i}, & x = x_t, \\ Q_t(x, a), & a = a_t, \\ Q_t(x, a), & \text{other.} \end{cases} \quad (11)$$

The virtual machines are scheduled according to the above rules to realize the adaptive deployment of IaaS cloud virtual machines.

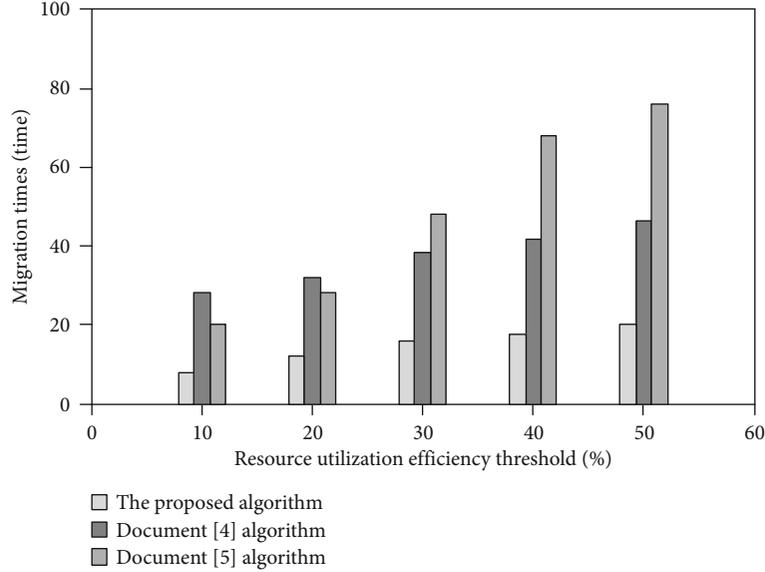


FIGURE 2: Comparison of the number of migrations by different algorithms.

3. Experiments and Results

In order to verify the overall effectiveness of the adaptive deployment algorithm for IaaS cloud virtual machines based on the Q learning mechanism, it is necessary to conduct related tests on the proposed algorithm. The switch in this test was a three-layer switch, with a power consumption of 80 W, physical machine power consumption of 750 W, and a network bandwidth capacity of 100 M.

The proposed algorithm, Reference [4] algorithm, and Reference [5] algorithm were used for comparison in the following tests.

3.1. Energy Consumption Test. The energy consumption of different algorithms in network congestion and noncongestion conditions was tested. The test results are shown in Figure 1.

From Figure 1, the energy consumption of the proposed algorithm did not change in the states with and without network congestion and was lower than that of Reference [4] algorithm and Reference [5] algorithm. The energy consumption of Reference [4] algorithm and Reference [5] algorithm in the state with network congestion was higher than that in the state without network congestion. Through the above analysis, it could be seen that the energy consumption of the proposed algorithm was not affected by network congestion. The reasons why the proposed algorithm simplified the learning process through a multistep Q learning algorithm based on the Q learning mechanism, and scheduling virtual machines improved the rationality of virtual machine deployment and reduced the energy consumption.

3.2. Analysis of the Number of Migrations. The number of migrations of the virtual machines of the proposed algorithm, Reference [4] algorithm, and Reference [5] algo-

rithm was compared under different resource utilization efficiency thresholds. The analysis results are shown in Figure 2.

From Figure 2, it shows that when increasing the resource utilization efficiency threshold, the number of migrations of the three algorithms gradually increased. Through comparison, it could be seen that the number of migrations of the proposed algorithm was the smallest, indicating that the proposed algorithm completed the deployment of the IaaS cloud virtual machines through the least migrations.

3.3. Average SLA Violation Rate. The average SLA violation rate was used as a test indicator to compare the deployment effectiveness of the proposed algorithm, Reference [4] algorithm, and Reference [5] algorithm. The lower the average SLA violation rate, the better the deployment effect of the algorithm. The average SLA violation rate can be calculated by

$$SLA_{\text{violation}} = \frac{N_q - N_f}{N_q}, \quad (12)$$

where N_q represents the processing capacity of all requests and N_f represents the processing capacity corresponding to the allocated requests.

The average SLA violation rates of the proposed algorithm, Reference [4] algorithm, and Reference [5] algorithm are shown in Table 1.

From Table 1, it indicates that the average SLA violation rate for the proposed algorithm did not change after the resource utilization efficiency threshold reached 60%. For Reference [4] algorithm and Reference [5] algorithm, the obtained average SLA violation rate increased with the resource utilization efficiency threshold increasing. Through analysis, it could be seen that the average SLA

TABLE 1: The average SLA violation rates by different algorithms.

Resource utilization efficiency threshold (%)	Average SLA violation rate (%)		
	Proposed algorithm	Reference [4] algorithm	Reference [3] algorithm
10	0	4.2	3.9
20	0	5.6	4.8
30	0	5.9	5.7
40	0	6.5	6.7
50	0	6.8	7.5
60	5.0	7.5	8.8
70	5.0	8.0	9.4
80	5.0	8.4	9.8
90	5.0	8.9	10.4
100	5.0	9.5	11.7

violation rate of the proposed algorithm was the lowest, indicating the best deployment effect by the proposed algorithm.

4. Conclusion

Various cloud service providers have begun to establish big data centers in the context of the continuous reduction of communication traffic costs and the gradual increase of network bandwidth. The energy consumption of big data centers is high. Therefore, enterprises and governments need to construct environmentally friendly and energy-saving cloud data centers. Deploying virtual machines can improve service quality and reduce energy consumption.

At present, the virtual machine deployment algorithm has the problems of high energy consumption, high number of migrations, and high average SLA violation rate. An adaptive deployment algorithm for IaaS cloud virtual machines based on the Q learning mechanism is proposed. First, the deployment problem of the IaaS cloud virtual machines is analyzed. Then, the Q learning mechanism is used to realize the adaptive deployment of IaaS cloud virtual machines, reducing the energy consumption, number of migrations, and average SLA violation rate. The future work is to improve the running speed of the algorithm while ensuring the accuracy.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares no competing interests.

Acknowledgments

The study is supported by 2019 Guangxi University's Young and Middle-aged Teachers' Basic Research Ability Improvement Project (2019KY1604).

References

- [1] S. Qiao, Q. Zhang, and Q. Zhang, "Hybrid seismic-electrical data acquisition station based on cloud technology and green IoT," *IEEE Access*, vol. 8, pp. 31026–31033, 2020.
- [2] M. K. Bouza, "Cloud technology is the foundation for designing efficient application," *Artificial Intelligence*, vol. 24, no. 1-2, pp. 14–23, 2019.
- [3] M. Mohammadhosseini, A. T. Haghighat, and E. Mahdipour, "An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm," *The Journal of Supercomputing*, vol. 75, no. 10, pp. 6904–6933, 2019.
- [4] Z. Lei and L. Wang, "Virtual machine deployment and migration optimization mechanism for cloud data center," *Computer Engineering and Design*, vol. 40, no. 8, pp. 2216–2223, 2019.
- [5] J. Liao, S. Qingxiao, and H. Yang, "Research on preemptible virtual machine instance configuration and scheduling methods that meet the execution time limit of workflow," *Computer Engineering and Science*, vol. 42, no. 11, pp. 1956–1964, 2020.
- [6] L. Jiliang, Q. Bing, and L. Wenjiang, "A cloud virtual machine scheduling strategy combining wake-up threshold and semi-sleep mode," *Journal of Yanshan University*, vol. 44, no. 4, pp. 370–378, 2020.
- [7] A. Ghasemi and A. T. Haghighat, "A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning," *Computing*, vol. 102, no. 9, pp. 2049–2072, 2020.
- [8] M. H. Sayadnavard, A. T. Haghighat, and A. M. Rahmani, "A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers," *Journal of Supercomputing*, vol. 75, no. 4, pp. 2126–2147, 2019.
- [9] S. Jin, X. Qie, W. Zhao, W. Yue, and Y. Takahashi, "A clustered virtual machine allocation strategy based on a sleep-mode with wake-up threshold in a cloud environment," *Annals of Operations Research*, vol. 293, no. 1, pp. 193–212, 2020.
- [10] H. Xu, P. Cheng, and Y. Liu, "A fault tolerance aware virtual machine scheduling algorithm in cloud computing," *International Journal of Performability Engineering*, vol. 15, no. 11, 2019.
- [11] M. J. Usman, A. S. Ismail, H. Chizari et al., "Energy-efficient virtual machine allocation technique using flower pollination algorithm in cloud datacenter: a panacea to green computing," *Journal of Bionic Engineering*, vol. 16, no. 2, pp. 354–366, 2019.
- [12] X. Xi, X. Cao, and P. Yang, "Network resource allocation for eMBB payload and URLLC control information communication multiplexing in a multi-UAV relay network," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1802–1817, 2021.
- [13] Z. Li and Z. Ding, "Distributed multiobjective optimization for network resource allocation of multiagent systems," *IEEE Transactions on Cybernetics*, 2020.
- [14] T. Coorevits, S. Kossman, D. Chicot, F. Hennebelle, A. Montagne, and A. Iost, "Virtual machine concept applied to uncertainties estimation in instrumented indentation testing," *Journal of Materials Research*, vol. 34, no. 14, pp. 2501–2516, 2019.
- [15] Z. Liang, Y. Liu, T.-M. Lok, and K. Huang, "Multi-cell mobile edge computing: joint service migration and resource

- allocation,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, 2021.
- [16] H. Qian, J. Yu, and L. Hua, “Relay selection algorithm based on social network combined with Q-learning for vehicle D2D communication,” *IET Communications*, vol. 13, no. 20, pp. 3582–3587, 2019.
- [17] Y. Fan, M. He, L. Su, and X.-H. Zhou, “A smoothed Q-learning algorithm for estimating optimal dynamic treatment regimes,” *Scandinavian Journal of Statistics*, vol. 46, no. 2, pp. 446–469, 2019.
- [18] Y. Huang, B. Yang, M. Wang, B. Liu, and X. Yang, “Analysis of the future land cover change in Beijing using CA-Markov chain model,” *Environmental Earth Sciences*, vol. 79, no. 2, pp. 1–12, 2020.
- [19] L. Ricci, “Asymptotic distribution of sample Shannon entropy in the case of an underlying finite, regular Markov chain,” *Physical Review E*, vol. 103, no. 2, article 022215, 2021.
- [20] M. Zhang, S. Shi, W. Cheng, and Y. Shen, “Self-adaptive hyper-heuristic Markov chain evolution for generating vehicle multi-parameter driving cycles,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6041–6052, 2020.
- [21] F. Brandon and B. Kosko, “Noise can speed Markov chain Monte Carlo estimation and quantum annealing,” *Physical review: E*, vol. 100, no. 5-1, pp. 53309–53309, 2019.
- [22] C. Smith, J. R. Lu, R. K. Thomas, I. M. Tucker, J. R. P. Webster, and M. Campana, “Markov chain modeling of surfactant critical micelle concentration and surface composition,” *Langmuir*, vol. 35, no. 2, pp. 561–569, 2019.
- [23] B. Cai, L. Zhang, and Y. Shi, “Control synthesis of hidden semi-Markov uncertain fuzzy systems via observations of hidden modes,” *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3709–3718, 2020.