Research on Task Scheduling Algorithms for Cloud-Edge Collaboration

Shuai Lu[1,2], Haibo Zhou[1,2], Shuaishuai Zhao[1,2], Wangbei Xu[1,2(✉)], and Kai Fang[3]

[1] National Demonstration Center for Experimental Mechanical and Electrical Engineering Education (Tianjin University of Technology), Tianjin 300384, China
xtjut2014@163.com

[2] Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, School of Mechanical Engineering, Tianjin University of Technology, Tianjin 300384, China

[3] College of Mathematics and Computer Sciences, Zhejiang A&F University, Hangzhou 311300, China

Abstract. With the advent of the 5G era, the development of IoT technology has been accelerated. Due to the continuous increase in the amount of data waiting to be processed from the edge, edge nodes may struggle to handle such a vast amount of data. Therefore, the technology of cloud-edge collaboration has emerged, and how to achieve cloud-edge collaborative task scheduling has become a current research hotspot. This article provides a detailed exposition of the relevant work on task scheduling in the cloud-edge environment, and outlines the common optimization objectives in the cloud-edge collaboration scenario. The methods used to solve task scheduling problems are classified and summarized, including heuristic, heuristic algorithm based on linear programming, and meta-heuristic algorithms. The advantages and disadvantages of each algorithm are analyzed. Finally, the development trends of large-scale task scheduling in the cloud-edge environment are discussed, providing valuable insights for achieving real-time performance, efficiency, and energy conservation in the Internet of Things.

Keywords: Cloud-Edge Collaboration · Optimization Objective · Task Scheduling

1 Introduction

The global number of mobile devices has increased from 8.8 billion in 2018 to 13.1 billion in 2023 [1]. With the rapid development of mobile networks and 5G technology [2], an increasing number of mobile devices are joining the Internet, forming a network space of interconnected mobile devices. However, relying solely on the computing resources of mobile devices cannot meet the fast-paced demands of the Internet of Things, leading to a decline in user experience [3, 4]. Eric Schmidt proposed the concept of cloud computing [5], which involves receiving, aggregating, and processing tasks in a centralized data center with powerful computing and storage capabilities. Users can obtain resources without space and time limitations. However, cloud computing faces

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 158–166, 2024.
https://doi.org/10.1007/978-3-031-65123-6_12

challenges in responding quickly and effectively to requests from the network edge. Edge computing [6] has emerged to address the long response latency problem of cloud computing by placing computing resources at the network edge to reduce network transmission delay and congestion [1, 7]. Cloud-edge collaboration refers to the cooperation between cloud computing and edge computing, which leverages the advantages of both to distribute computing resources between the cloud and the edge, thereby improving computing efficiency and enhancing user experience. Task scheduling, as a crucial link in current cloud-edge collaborative computing models, has become a major research focus and challenge. This article elaborates on the architecture of cloud-edge collaborative systems and, based on optimization objectives, outlines the research status of task scheduling algorithms and summarizes the development trends of task scheduling problems in cloud-edge environments.

2 Cloud Edge Collaboration System

Before the concept of cloud-edge collaboration [8, 9] was introduced, the computing problem of terminal devices mainly relied on its own computing power. However, with the explosive growth of massive data, the computing power of mobile devices is obviously insufficient. In the 1990s, Akamai proposed the Content Delivery Network (CDN) [10], which designed network nodes that can store static content near the terminal locations to place content closer to users and also act as a load balancer (see Fig. 1). This reduces the load on the source station and shortens the distance between users and services at the network level, thus enabling high-speed transmission of interactive information such as text, images, and audio [11].

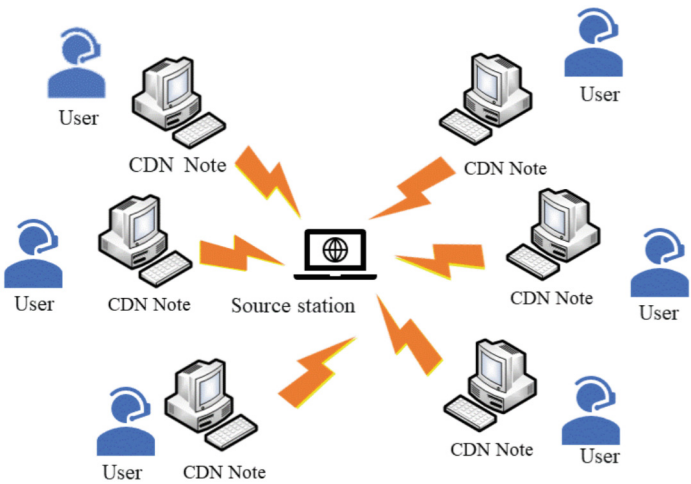

Fig. 1. CDN Structure

In 2006, Eric Schmidt, the CEO of Google, first introduced and elaborated on the concept of "cloud computing" at the Search Engine Conference. Since then, related

products and services have continuously emerged, and the business model of cloud computing has gradually taken shape, marking the nascent stage of cloud computing. In 2009, the research team at Carnegie Mellon University proposed the concept of Cloudlet [12], which combines cloud computing and Internet of Things devices. Cloudlet can serve as a data center located at the network edge, providing sufficient computing resources for mobile devices. In 2013, LaMothes [13] first proposed the concept of edge computing, where users offload computing tasks to edge nodes for processing in edge networks, resulting in low-latency and low-energy consumption service experience. With the explosive growth of mobile data, edge computing has rapidly developed, with computing gradually shifting from the cloud to the edge, and the concept of cloud-edge collaboration [14] being widely discussed.

In cloud-edge collaboration, cloud computing and edge computing have a mutually beneficial relationship [15]. Edge nodes collect data from user ends and perform small-scale processing and analysis using their own computing power. Data is offloaded to the edge for computing, greatly reducing the workload of the cloud and ensuring the privacy of user data. Cloud computing manages edge nodes and analyzes the edge environment after receiving data processed by the edge. It utilizes its powerful computing power to assist the edge in processing large-scale data. Therefore, cloud-edge collaboration can achieve collaborative resource management, resource scheduling, information collection, and service updates. The architecture of a cloud-edge collaborative system is shown (see Fig. 2).

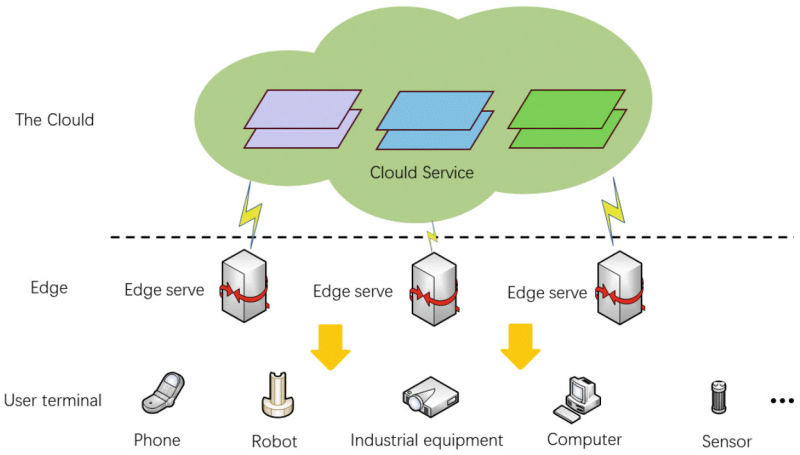

Fig. 2. System framework

3 Optimization Objective

With the development and popularization of cloud computing and edge computing, cloud-edge collaboration has become an important computing paradigm. However, due to the diversification of application scenarios and the continuous increase of computing

workloads, task scheduling in cloud-edge collaboration faces numerous challenges. By implementing rational task allocation and scheduling strategies, optimizing objectives such as low latency, low energy consumption and low latency and low energy consumption, and meeting the requirements of different application scenarios, the performance and sustainability of cloud-edge collaboration can be enhanced.

3.1 Low Latency

It is common to improve the efficiency of cloud-edge collaborative task scheduling by reducing the running time. Miao et al. [16] adopted the LSTM algorithm to predict computing tasks, and proposed a task prediction-based mobile terminal computing offloading strategy and a task migration-based edge-cloud scheduling scheme to assist in optimizing the edge computing offloading model. With the increase of data and sub-tasks, this algorithm can effectively reduce the total delay of tasks. Lin et al. [17] considered the data distribution characteristics of edge computing and cloud computing, as well as the influencing factors such as data bandwidth between cloud and edge, the number of edge data centers, and the storage capacity of edge data centers. They proposed an adaptive particle swarm optimization algorithm with a hybrid genetic operator, which uses genetic algorithm's crossover and mutation operators to avoid the premature convergence of traditional particle swarm optimization algorithm, and enhances the diversity of population evolution, effectively reducing the data transmission delay between the cloud and edge in the workflow scheduling. Zhan et al. [18] proposed a distributed offloading algorithm based on deep reinforcement learning (DRL-DU). This method describes the task scheduling process as a Markov decision process, uses sequence-to-sequence deep neural networks to represent scheduling policies, and trains them with proximal policy optimization (PPO) to learn the optimal actions to be taken in different states, so as to maximize the rewards obtained. This method can effectively optimize task scheduling in edge computing, thereby reducing application execution time and minimizing delay.

3.2 Low Energy Consumption

Gai et al. [19] constructed an energy-aware heterogeneous cloud management (EA-HCM) model to address the energy waste caused by task allocation to remote cloud servers and edge nodes. The core algorithm of this model is the heterogeneous task allocation algorithm (HTA2), which dynamically assigns tasks to various edge nodes and remote clouds, thus reducing the total energy consumption of the system. Wu et al. [20] proposed a green energy-saving scheduling algorithm for cloud data centers based on dynamic voltage and frequency scaling (DVFS) technology. This algorithm controls the power supply voltage and frequency of servers in cloud computing, and can reduce the energy consumption of servers in idle mode or light load, thus achieving the goal of energy saving.

3.3 Low Latency and Low Energy Consumption

The multi-objective optimization of reducing latency and energy consumption has been receiving widespread attention from both academia and industry [21]. Ma et al. [22]

balanced both objectives by optimizing the weighted system cost and latency in cloud-assisted mobile edge computing (CAME) systems, using Karush-Kuhn-Tucker conditions to minimize both delay and cost and obtain a policy for task allocation to mobile devices, edge nodes, or clouds. Shi et al. [23] proposed a cloud computing task scheduling model that optimized time, cost, and workload as objectives, and used an improved whale algorithm for task scheduling. The initialization of the population was performed using an improved Bernouilli Shift chaotic mapping to increase diversity, while adaptive factors were used to balance local and global search capabilities. Individual updates were performed using the differential algorithm to alleviate the drawbacks of long task scheduling time and high cost. Hu et al. [24] proposed a cloud-edge collaborative task scheduling algorithm based on simulated annealing algorithm for multi-objective optimization of time and energy consumption. This algorithm optimized both completion time and energy consumption while satisfying the completion time constraints, using continuous dynamic voltage regulation technology to reduce the energy consumption of mobile devices and improve the execution speed of workflows.

4 Task Scheduling

Task scheduling algorithms for cloud-edge collaboration is an algorithm that allocates cloud and edge computing resources in a rational way, aiming to optimize the performance and efficiency of the entire system. It is mainly applied in scenarios that require data processing and large-scale computation, such as intelligent manufacturing, industrial Internet, smart cities, and other fields.

4.1 Heuristic Algorithm

Heuristic algorithm is a class of non-strict mathematical methods based on experience and heuristic rules to solve optimization problems, mainly used to solve complex optimizations such as NP-hard problems. The advantage of heuristic algorithms lies in their flexibility and adaptability, allowing them to handle various problem types and sizes, but they do not guarantee finding the optimal solution.

Lin et al. [25] proposed a heuristic task scheduling algorithm based on fruit fly optimization, which redefines the computing requests and uses multi-objective functions to search for the optimal strategy in the cloud, optimizing power consumption and operation speed by transferring tasks from intelligent devices to cloud computing. Huang et al. [26] proposed a simulated annealing algorithm with a rearrangement strategy task scheduling algorithm, which rearranges tasks by using the rearrangement strategy of simulated annealing algorithm when the terminal device sends the relevant task to the cloud. This rearrangement process lasts for a certain number of rounds, and each task has the opportunity to be reassigned to one of the edge servers. If this task is rearranged to another edge server to obtain a smaller global makespan (the maximum transmission time of the edge server), it is considered as a better state movement and is accepted; otherwise, it is reassigned.

Yu et al. [27] used genetic algorithms to optimize task allocation in cloud computing, considering the load of CPU and memory and formulating an allocation strategy to

maintain load balance among physical machines in cloud computing. The optimal task allocation strategy was obtained by selecting, crossing, and mutating steps, but the algorithm took a long time to converge to the optimal solution. Juan et al. [28] proposed a Pareto-based list scheduling heuristic algorithm, which extends the Heterogeneous Earliest Finish Time (HEFT) algorithm for optimizing workflow completion time to achieve multi-objective workflow scheduling. In each iteration, the algorithm maps the tasks to be executed to all possible resources and adds the new schedule to a new temporary collection. Only the best solution is saved at the end of the iteration. Su et al. [29] designed a cost-aware dual-task heuristic scheduling algorithm to balance efficiency and cost. The concept of Pareto dominance was used to generate a cost-effective task schedule based on the execution time of key tasks and the cost of Virtual Machine (VM). Non-critical tasks were minimized in cost by extending their execution time to achieve the goal of reducing cost.

4.2 Heuristic Algorithm Based on Linear Programming

Heuristic algorithm based on linear programming is an algorithm used to solve optimization problems. Its core idea is to solve the model based on linear programming and improve the model in a heuristic way to find the optimal solution of the problem faster. However, it cannot handle problems with nonlinear constraint conditions or objective functions very well.

Li et al. [30] abstracted the problem of minimizing task scheduling execution time in cloud-edge environment as a mixed-integer linear programming (MILP) model, considering the heterogeneity of transmission rate and processing capability. They proposed an optimization method combining the logic-based Benders decomposition (LBBD) principle with the MILP model. This method can quickly obtain a scheduling scheme and realize the task offloading of intelligent toys in edge cloud computing. Bahreini et al. [31] described the offline problem as a MILP and designed an efficient heuristic online algorithm. They also proposed a mixed-integer linear programming model considering the dynamic nature of user location and network capacity for multi-component application layout problems. The algorithm is based on an iterative matching process, followed by a local search stage, which has a small execution time and can obtain an approximate optimal solution. Wang et al. [32] proposed a dynamic task scheduling algorithm based on a weighted bipartite graph model. This algorithm considers the dynamic nature of tasks, converts the scheduling problem into a maximum weighted bipartite matching problem, and establishing an integer programming model to achieve the best scheduling of dynamic tasks.

4.3 Metaheuristic Algorithm

Metaheuristic algorithms are a class of approaches that combine multiple heuristic algorithms. By integrating and adjusting different heuristic algorithms, this method enables the comprehensive utilization of their respective advantages, thereby enhancing the efficiency and quality of problem-solving.

Shen et al. [33] proposed a genetic algorithm called Energy-Performance Genetic Algorithm (E-PAGA), which first models virtual machine energy from the perspective

of cloud task scheduling. Then, two fitness functions are designed to adaptively adjust the energy and performance objectives before task allocation in the cloud by adjusting weights, to meet the scheduling requirements of different users. Wang [34] proposed an improved fireworks algorithm, which quantifies tasks and resources into computational, storage, and bandwidth-sensitive types. The load balancing of devices within the cloud-edge system is used as the optimization objective. By introducing a fireworks explosion radius detection mechanism and setting a minimum threshold for the explosion radius and adaptively adjusting the radius parameter, the algorithm's local search capability is improved, and the overall effect is improved. However, the algorithm uses Euclidean distance as the fireworks distance, ignoring the differences between task-node pairs, which affects the scheduling effect. Rodriguez et al. [35] proposed a scheduling strategy based on a metaheuristic particle swarm optimization (PSO) algorithm. The PSO algorithm is used to search for the optimal solution. Each particle represents a potential scheduling scheme, where each element in its position vector corresponds to a task in the workflow, and its value indicates the task should be assigned to the corresponding edge. In each iteration calculation, the fitness value is calculated based on the particle's position, and the scheduling scheme is optimized through reward and punishment mechanism, dynamically configuring and scheduling computing resources at the cloud and edge. Xu et al. [36] proposed a multi-priority queue genetic algorithm (MPQGA) and treated cloud and edge nodes as processing units in a heterogeneous computing system. The MPQGA algorithm is used to assign priorities to each subtask, and the earliest finish time (EFT) method is used to search for the solution of task-to-processor mapping. This method can effectively coordinate the task scheduling between the cloud and edge nodes, improve computing efficiency, and reducing latency.

5 Conclusion

This paper provides an overview of the development of cloud-edge collaboration, outlines the optimization objectives in cloud-edge environments, and classifies task scheduling algorithms into heuristic algorithms, heuristic algorithm based on linear programming and metaheuristic algorithms. The research status of the above algorithms is described in detail, and the advantages and limitations of various algorithms are compared and analyzed. For future multi-objective and large-scale task scheduling problems in cloud-edge environments, the following work will be carried out:

- Task scheduling in large-scale cloud-edge collaborative computing systems is extremely complex and requires consideration of more factors, such as time response, energy consumption, resource constraints and load balancing, etc. Therefore, further exploration and improvement of scheduling algorithms will be necessary to meet the needs of large-scale systems.
- In practical applications, cloud-edge collaborative systems need to be adjusted and optimized according to different application scenarios and requirements. Therefore, adaptive algorithms and mechanisms will need to be explored in order to enable the system to better adapt to different environments.

Acknowledgement. This research was funded by Tianjin multi-investment fund project key project (21JCZDJC00870), Tianjin Natural Science Foundation key project (17JCZDJC30400) and Tianjin graduate research innovation project (2022SKYZ253).

References

1. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
2. Qian, M., Wang, Y., Zhou, Y., Tian, L., Shi, J.: A super base station based centralized network architecture for 5G mobile communication systems. Digit. Commun. Netw. **1**(2), 152–159 (2015)
3. Ren, J., Yu, G., He, Y., Li, G.Y.: Collaborative cloud and edge computing for latency minimization. IEEE Trans. Veh. Technol. **68**(5), 5031–5044 (2019)
4. Dinh, T.Q., Liang, B., Quek, T.Q.S., Shin, H.: Online resource procurement and allocation in a hybrid edge-cloud computing system. IEEE Trans. Wireless Commun. **19**(3), 2137–2149 (2020)
5. Helali, L., Omri, M.N.: A survey of data center consolidation in cloud computing systems. Comput. Sci. Rev. **39**, 100366 (2021)
6. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: a survey. IEEE Internet Things J. **5**(1), 450–465 (2018)
7. Cheng, J., Chen, W., Tao, F., Lin, C.: Industrial IoT in 5G environment towards smart manufacturing. J. Ind. Inf. Integr. **10**, 10–19 (2018)
8. Chen, M., Hao, Y., Hu, L., Hossain, M.S., Ghoneim, A.: Edge-CoCaCo: toward joint optimization of computation, caching, and communication on edge cloud. IEEE Wirel. Commun. **25**(3), 21–27 (2018)
9. Abbasi, M., Mohammadi-Pasand, E., Khosravi, M.R.: Intelligent workload allocation in IoT–Fog–cloud architecture towards mobile edge computing. Comput. Commun. **169**, 71–80 (2021)
10. Vakali, A., Pallis, G.: Content delivery networks: status and trends. IEEE Internet Comput. **7**(6), 68–74 (2003)
11. Chen, X.F., Zhou, Y.M., Ao, Q.Y., Bai, Y.C.: Design and implementation of java and web-based distributed real-time network monitoring system. Comput. Eng. **28**(6), 139–140 (2002)
12. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
13. Hu, Y.C., Patel, M., Sabella, D., Sprecher, N., Young, V.: Mobile edge computing a key technology towards 5G. ETSI White Paper **11**(11), 1–16 (2015)
14. Jing, Z., Feng, H., Yuanyi, C.: Power customer-side IoT dispatching system based on cloud-side-end collaboration. Electric Eng. **1**, 173–175 (2022)
15. Kong, L.N., Guo, H.M., Jiao, H.: A cloud-edge collaboration framework for data collection. Digit. Technol. Appl. **39**(2), 165–167 (2021)
16. Miao, Y., Wu, G., Li, M., Ghoneim, A., Al-Rakhami, M., Hossain, M.S.: Intelligent task prediction and computation offloading based on mobile-edge cloud computing. Futur. Gener. Comput. Syst. **102**, 925–931 (2020)
17. Lin, B., et al.: A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. IEEE Trans. Industr. Inf. **15**(7), 4254–4265 (2019)
18. Zhan, W.H., Wang, J., Zhu, Q.X., Duan, H.C., Ye, Y.L.: Deep reinforcement learning based offloading scheduling in mobile edge computing. Appl. Res. Comput. **38**(1), 241–245 (2021)

19. Keke, G., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2017)
20. Wu, C., Chang, R., Chan, H.: A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. Futur. Gener. Comput. Syst. **37**, 141–147 (2014)
21. Wang, T., Cheng, L., Zhang, K., Liu, J.: Energy-aware service composition algorithms for service-oriented heterogeneous wireless sensor networks. int. J. Distrib. Sens. Netw. **10**(3), 217102 (2014)
22. Ma, X., Zhang, S., Li, W., Zhang, P., Lin, C., Shen, X.: Cost-efficient workload scheduling in cloud assisted mobile edge computing. In: ACM 25th International Symposium on Quality of Service, Vilanova i la Geltrú, Spain, pp. 1–10. IEEE (2017)
23. Shi, Z.H.: Study of cloud computing task scheduling based on improved whale algorithm. Bull. Sci. Technol. **37**(2), 67–71 (2021)
24. Hu, H.Y., Liu, R.H., Hu, H.: Multi-objective optimization for task scheduling in mobile cloud computing. J. Comput. Res. Dev. **54**(09), 1909–1919 (2017)
25. Lin, K., Pankaj, S., Wang, D.: Task offloading and resource allocation for edge-of-things computing on smart healthcare systems. Comput. Electr. Eng. **72**, 348–360 (2018)
26. Huang, Y., Zhu, Y., Fan, X., Ma, X., Wang, F., Liu, J.: Task scheduling with optimized transmission time in collaborative cloud-edge learning. In: 27th International Conference on Computer Communication and Networks, Hangzhou, pp. 1–9. IEEE (2018)
27. Huang, Y.L., Li, Z.X.: A GA-based resource management algorithm for smart living applications requiring intensive computing power. In: IEEE International Conference on Consumer Electronics, Taiwan, pp. 259–260. IEEE (2017)
28. Durillo, J.J., Prodan, R.: Multi-objective workflow scheduling in Amazon EC2. Clust. Comput. **17**(2), 169–189 (2014)
29. Su, S., Li, J., Huang, Q., Huang, X., Shuang, K., Wang, J.: Cost-efficient task scheduling for executing large programs in the cloud. Parallel Comput. **39**(4–5), 177–188 (2013)
30. Li, S., Chen, W., Chen, Y., Chen, C., Zheng, Z.: Makespan-minimized computation offloading for smart toys in edge-cloud computing. Electron. Commer. Res. Appl. **37**, 100884 (2019)
31. Bahreini, T., Grosu, D.: Efficient placement of multi-component applications in edge computing systems, pp. 1–11. ACM, New York (2017)
32. Wang, T., Wei, X., Liang, T., Fan, J.: Dynamic tasks scheduling based on weighted bi-graph in mobile cloud computing. Sustain. Comput. Inform. Syst. **19**, 214–222 (2018)
33. Shen, Y., Bao, Z., Qin, X., Shen, J.: Adaptive task scheduling strategy in cloud: when energy consumption meets performance guarantee. World Wide Web **20**(2), 155–173 (2017)
34. Wang, S., Zhao, T., Pang, S.: Task scheduling algorithm based on improved firework algorithm in fog computing. IEEE Access **8**, 32385–32394 (2020)
35. Rodriguez, M.A., Buyya, R.: Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Trans. Cloud Comput. **2**(2), 222–235 (2014)
36. Xu, Y., Li, K., Hu, J., Li, K.: A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. Inf. Sci. **270**, 255–287 (2014)