Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7873131, 10 pages
https://doi.org/10.1155/2022/7873131

Research Article

Virtual Machine Resource Allocation Optimization in Cloud Computing Based on Multiobjective Genetic Algorithm

Feng Shi[1] **and Jingna Lin**[2]

[1]*Department of Computer Science and Technology, Taiyuan University, Taiyuan 030032, China*
[2]*Shanxi Vocational University of Engineering Science and Technology, Taiyuan 030031, China*

Correspondence should be addressed to Jingna Lin; linjingna@bcsxu.edu.cn

Received 26 November 2021; Revised 17 January 2022; Accepted 4 February 2022; Published 10 March 2022

Academic Editor: Qiangyi Li

Copyright © 2022 Feng Shi and Jingna Lin. 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.

Cloud computing is an important milestone in the development of distributed computing as a commercial implementation, and it has good prospects. Infrastructure as a service (IaaS) is an important service mode in cloud computing. It combines massive resources scattered in different spaces into a unified resource pool by means of virtualization technology, facilitating the unified management and use of resources. In IaaS mode, all resources are provided in the form of virtual machines (VM). To achieve efficient resource utilization, reduce users' costs, and save users' computing time, VM allocation must be optimized. This paper proposes a new multiobjective optimization method of dynamic resource allocation for multivirtual machine distribution stability. Combining the current state and future predicted data of each application load, the cost of virtual machine relocation and the stability of new virtual machine placement state are considered comprehensively. A multiobjective optimization genetic algorithm (MOGANS) was designed to solve the problem. The simulation results show that compared with the genetic algorithm (GA-NN) for energy saving and multivirtual machine redistribution overhead, the virtual machine distribution method obtained by MOGANS has a longer stability time. Aiming at this shortage, this paper proposes a multiobjective optimization dynamic resource allocation method (MOGA-C) based on MOEA/D for virtual machine distribution. It is illustrated by experimental simulation that moGA-D can converge faster and obtain similar multiobjective optimization results at the same calculation scale.

1. Introduction

Cloud computing is an emerging technology in the field of computers. Although it has been applied, its development is not perfect. Many technical standards have not yet formed, and are still in the period of growth. As long as we grasp the rare opportunity, we can occupy a place in the future cloud computing field [1–3]. In this critical period of the formation and development of new technology, whoever holds the commanding heights of technology will hold the initiative of future technology strategy [4]. In addition to commercial applications, many excellent characteristics of cloud computing make it an essential aspect in military education, meteorology, astronomy, and other fields. Therefore, the development of cloud computing technology has been included in the national science and technology strategy of many countries [5–7].

In cloud computing applications, Infrastructure as a Service (IaaS) is an important service model, through which cloud providers can provide massive computing resources to cloud users through the Internet. In addition, the price fluctuation and the uncertainty of demand and other factors make this problem more complicated. Therefore, in order to meet the resource usage requirements and minimize user expenses, the VM allocation scheme must be optimized. This problem is called VM allocation optimization (shown as Figure 1) [8, 9].

The emergence of cloud computing platform virtualization technology brings new opportunities to realize energy saving in the cloud computing environment. The use of

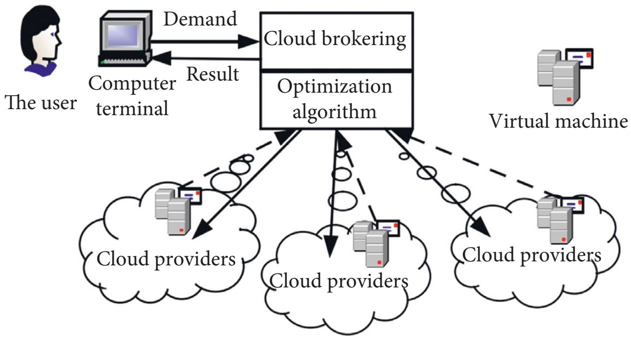

Figure 1: VM resource allocation in cloud computing.

virtualization technology enables multiple users to share the computing resources pool according to the actual demand. The emergence of this technology has greatly improved the resources utilization. With the emergence of a shared resource pool, the number of servers in the cloud environment has reduced, thereby increasing the income of the cloud service provider [2, 10, 11]. And, resource allocation problems ultimately boils down to multiple application mapping relations between the virtual machine and the server, through reasonable allocation, adjusting the distribution of application virtual machines on the physical nodes, making full use of the service condition of the server idle resources, so as to reduce the amount of activation server, and thereby achieve the purpose of reducing energy consumption [12]. At present, there are three basic service modes in the application of cloud computing technology: Software as a Service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS), as shown in Figure 2. Each service mode corresponds to certain system levels.

At present, many energy-saving resource scheduling methods of cloud platform focus on how to reduce the number of active servers to achieve energy saving and green computing. However, the above resource scheduling schemes often ignore the dynamic change of application load caused by the change of user demand in the cloud environment, and they also fail to consider the stability of application load distribution on each physical node. Although the distribution of application VMS among physical nodes in the current state makes fewer physical nodes active, the change of user requirements will lead to changes in the load of each application; in this case, the dynamic change of the required load will lead to resource hot spots on these physical nodes in the future with the change of the application load. The status of these physical nodes may reappear in the near future, which will directly trigger a new round of dynamic resource configuration requirements [13]. To solve the above problems and to put forward a new thinking of physical nodes on the stability of the energy-saving resource scheduling method, how to find a virtual machine distribution that activates the stable state of each physical node and how to reduce the energy consumption of virtual machines from the old state to the new state are urgent and necessary [14].

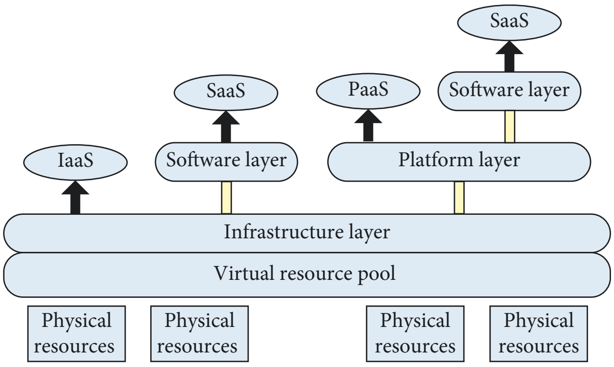

Figure 2: Framework diagram of the cloud computing service model.

2. Related Works

With the development of cloud computing, people find that although cloud computing brings convenience, its energy consumption is also huge compared with the traditional service mode. Now, some suppliers in order to meet user requirements, which can respond to dynamic changes in user needs. However, in the actual environment, this will cause a great waste of resources and increase unnecessary energy consumption. With the change of user demand, the application load request resources will change accordingly. Therefore, how to reduce energy consumption while ensuring service quality is a problem to be solved. In a cloud computing environment, the optimal VM resource distribution scheme can effectively save energy by properly deploying VM resources on physical nodes [15, 16]. Virtual machine deployment strategy is mainly considering the distribution and use of computing resources in data center, so as to select the best distribution scheme, according to the distribution plan to make sure the virtual machine and the mapping relations between physical nodes. If the new deployment scheme satisfies the constraints of the collection, then optimal allocation can be achieved. However, the allocation of multiple virtual machines to multiple physical nodes proved to be a problem, namely NP problem. Therefore, which strategy is adopted to realize the placement of virtual machines to achieve the purpose of energy saving has always been a hot research topic in the field of cloud computing [17, 18].

At present, the research of cloud computing is still in the initial stage. In terms of the development direction of the future mainstream cloud computing industry, there are also two different camps among the big companies that master the core technology of cloud computing. Emerging companies represented by Google and Amazon are pushing the Internet cloud first, while traditional IT enterprises represented by IBM and Microsoft are pushing the IT cloud. In addition, due to commercial interests, many high-end cloud computing technologies are not disclosed as business secrets of enterprises [19]. Google is recognized as the leader in cloud computing technology, occupying a great technological advantage, but many of its technologies are not disclosed. At present, scholars mainly study cloud computing in two ways: one is based on part of the macroscopic technical data disclosed by Google and combined with traditional parallel computing-based grid computing virtualization technology [20]. The other is based on the open source project Hadoop for communication and application of different ideas and different application environments, resulting in the development of many different branches of technology. Therefore, related research on cloud computing is still in a stage of contention, and the technical system has not yet been formed.

As mentioned above, the widespread interest in cloud computing stems from its successful commercial application, so the earliest cloud computing research institutes are mainly concentrated in several major Western IT technology companies. However, with the gradual increase of the influence of cloud computing, its many excellent characteristics make people see its application in other fields such as military scientific research. Cloud computing technology began to receive attention from various countries and has become an important aspect for countries to compete for the commanding heights of future science and technology [2]. In China, in recent years, due to the strong support of the state, as well as the extensive promotion of business plans, cloud computing has gradually become a popular research content, has achieved rapid development, and its influence is also increasing. In 2010, Beijing successfully hosted the world cloud computing conference. Lenovo, Sohu, Baidu, and other domestic companies have also begun to develop business based on cloud computing. In addition, various domestic universities began to set up the relevant research direction and discipline curriculum. Cloud computing research revealed a strong momentum, and has great development potential. However, there is still a big gap between China and western countries, and more investment and policy planning are needed [21].

Resource management and allocation is a key problem in cloud computing. In the IaaS mode, resource allocation is a virtual machine allocation problem. Virtual machine allocation, like other cloud computing technologies, is in a state of constant discovery. Literature [22] proposed an algorithm based on cloud agent, which aims to provide virtual machine allocation scheme for physical server. Literature [23] proposed an MMS virtual machine allocation algorithm based on min-max and shares features provided by the hypervisor. However, these algorithms do not take into account the uncertain factors of demand and the minimization conditions of price. A dynamic integration mechanism based on constraint programming is proposed in literature [24]. In literature [25], demand and price factors are considered, but response time and other issues are not taken into account. These algorithms are based on different application scenarios, and have their own characteristics. The existing algorithm, however, is mainly to study the internal allocation of resources on a single virtual machine. It is used to improve the utilization of vm physical resources and meet users' computing requirements by dividing physical computing resources [26]. In addition, most algorithms are put forward in specific demand scenarios, without considering various uncertainties in practical applications.

From the above analysis, we know that the above methods have studied virtual machine resource allocation optimization in cloud computing to some extent, but some problem still exists. On the other hand, no scholar has applied multiobjective genetic algorithm to this field till now, so the research here still draws a blank, which has great theoretical research and practical application value.

The contributions of this paper are as follows:

(1) The proposed evolutionary algorithm provides a new solution idea for solving NP problems, and the emergence of multiobjective optimization algorithm brings a new solution idea for complex multiobjective optimization practical problems.

(2) Because of its good development prospects, the multiobjective optimization problem has become a research hotspot in the academic world. In this paper, MOGANS- and MOEA/D-based multiobjective optimization scheduling schemes are proposed according to the solution idea of multiobjective optimization.

This paper consists of five parts. The first and second parts give the research status and background. The third part is the MOEA/D-based virtual machine resource allocation in cloud computing. The fourth part shows the experimental results and analysis. The experimental results of this paper are introduced, compared, and analyzed with relevant comparison algorithms followed. Finally, the fifth part summarizes the full paper.

3. MOEA/D-Based Virtual Machine Resource Allocation in Cloud Computing

3.1. Conventional Particle Swarm Optimization Algorithm. Conventional particle swarm optimization (PSO) is an intelligent bionic method, which was jointly proposed by social psychologists Kennedy and Dr. Eberhart in 1995. It has become the mainstream intelligent optimization processing method at present and has been comprehensively applied in the academic and industrial circles. Its optimization performance is mainly reflected in convergence [27, 28].

$$\chi = \frac{2}{\sqrt{\left(2 - l - \sqrt{l^2 - 4l}\right)^2}}, \quad l = c_1 + c_2, l > 4, \tag{1}$$

where χ is the convergence factor of particle swarm. For the particle in each iteration, the particle changes its velocity and position according to the following formula:

$$v_{\mathrm{id}}^{k+1} = v_{\mathrm{id}}^{k} + \chi\left[c_1 r_1\left(F_{\mathrm{id}} - Q_{\mathrm{id}}^{k}\right) + c_1 r_2\left(H_{\mathrm{gd}} - Q_{\mathrm{id}}^{k}\right)\right], \tag{2}$$

$$Q_{\mathrm{id}}^{k+1} = Q_{\mathrm{id}}^{k} + v_{\mathrm{id}}^{k+1}, \tag{3}$$

$$F_{\mathrm{id}} \geqslant H_{\mathrm{gd}} n, \tag{4}$$

where $d = i = 1, 2, \ldots, m$, k is the number of iterations, and c_1, c_2 are decay acceleration factors.

The (PSO) method here is determined by cloud user U and cloud VM, that is, the utility function of cloud user U is used to obtain the optimal position of particle search, and VM is used to obtain the optimal position of overall particle swarm search, that is, one particle represents one cloud user, The whole particle represents the virtual machine resources in the cloud computing environment of the whole field, which corresponds to the particle swarm optimization method. In this way, not only can each cloud user get the maximum benefit but also can make the whole cloud computing environment VM get global optimization.

At present, many scholars have put forward many excellent algorithms for solving multiobjective problems, among which the representative ones are NSGA-II, PSO, etc. However, these algorithms do not adopt decomposition thought in solving the MOP problem. These algorithms mainly deal with the MOP problem through some algorithms, and finally integrate the MOP into a whole. Then, it is solved by single objective optimization problem. But the downside of this is that it becomes very difficult to allocate fitness. However, many of the target values are independent, so there are many different cases of how to judge the superiority of a solution, and there are many other cases of the relationship between the solutions besides being dominant. Therefore, it is difficult to specify suitable fitness for these solutions to compare the advantages and disadvantages of these solutions in order to perform selection operations. In view of this problem, the decomposition idea is put forward, and the decomposition strategy idea is to transform multiple objectives into solving the problem of each single objective, mainly by solving the problem with multiple objectives. It is decomposed into a single solution optimization problem. By optimizing the solution of each sub-objective space, the solution of the whole solution set can be optimized to achieve the purpose of multiobjective optimization, and finally the global optimization solution based on Pareto is obtained. Therefore, MOEA/D comes into being.

3.2. Decomposition Strategy of MOEA/D Algorithm. The decomposition idea of MOEA/D is that, unlike other multiobjective optimization algorithms, the multiobjective problem is not processed as a whole through integration. Instead, it transforms the multiobjective solution into solving the single-objective optimization problem in N subspace through decomposition of the solution space. Among them, the solution in the subspace transforms a multiobjective problem into a single-objective problem by means of aggregation method, and the common aggregation methods include Chebyshev method, weighted sum method, and so on. In this paper, the aggregation function is implemented by Chebyshev:

$$\text{minimize } g^{\mathrm{te}}\left(\frac{x}{y}, z^{*}\right) = \max_{1\leq i\leq x}\left\{\lambda_i\left|f_i(x) - z_i^{*}\right|\right\}, \qquad \text{subject to} \quad x \in \Omega n. \tag{5}$$

And $\sum_{i=1}^{m} \lambda_i = 1\,(\forall i, \lambda_i \geq 0)$, $z_* = (z_1^*, z_1^*, \ldots z_m^*)^T$ are the reference points. And, z^* is the optimal value of each target value in all solutions.

3.3. Implementation of MOGA-D Resource Optimization Scheme. The main characteristics of MOEA/D algorithm are as follows:

(1) MOEA/D algorithm does not transform the multitarget values into a single-target progressive solution through the traditional method but decomposes the multitarget space, thus transforming the multitarget solution problem into the optimization problem of solving the sub-target space solution, and the sub-target space solution can be transformed into a single-target solution problem according to the vector of the solution space.

(2) An excellent core idea of MOEA/D is to update the new solution by exchanging information of neighborhood solutions. This strategy greatly speeds up the search of understanding space, and has lower time and space complexity compared with some other multiobjective optimization solutions

Because the resource scheduling problem in cloud computing environment can finally be transformed into solving a multiobjective optimization problem through evolutionary algorithm, the resource scheduling scheme based on MOEA/D algorithm is proved by experiments to reduce the time complexity of the algorithm compared with MOGANS, so as to solve the shortage of MOGANS in computing performance.

The algorithm mainly consists of four parts: target space decomposition, population classification, crossover, and mutation. The core of the multiobjective optimization problem is to decompose the solution space into N sub-target space, where OMEGA represents the target space, so omega can be decomposed into a sub-target space composed of $(\Omega_1, \Omega_2, \ldots, \Omega_N)$. Individuals in the population can be mapped into different subspaces according to the generated sub-target space, so as to achieve the classification of the population. The vector composed of each target value of individuals in the initial population is mapped into N sub-target space to ensure that each sub-target space has a solution. The division of target space and population classification can be realized by the following formula:

$$p^{\mathrm{i}} = \left\{ x | x \in \mathrm{POP}, \Delta\left(F(x), \lambda^{i}\right) = \max_{1 \le j \le N} \left\{ \Delta\left(F(x), \lambda^{j}\right) \right\} \right\}, \tag{6}$$

$$\Delta\left(F(x), \lambda^{i}\right) = \frac{\lambda^{i} * (F(x) - Z)^{T}}{\left\| \lambda^{i} \right\| * \|F(x) - Z\|}, \quad i = 1, \ldots, m, \tag{7}$$

$$\Omega_{\mathrm{i}} = \left\{ F(x) | x \in \Omega, \Delta\left(F(x), \lambda^{i}\right) = \max_{1 \le j \le N} \left\{ \Delta\left(F(x), \lambda^{j}\right) \right\} \right\}, \tag{8}$$

where Z is a reference point and $\Delta(F(x), \lambda^{i})$ is the optimal solution of each target vector. It is also the cosine of the angle. In this way, POP population is divided into N classes. So there is a solution in every subspace and you have a variety of systems that help maintain the solution. Then, the multiobjective solving problem can be transformed into a single-objective optimization problem by the following aggregation function, which is in the form of the following:

$$\text{minimize}\, \mathrm{g}^{te}\left(x | y, Z^{*}\right) = \max_{1 \le j \le m} \left\{ \frac{\left| f_{j}(x) - z_{j}^{*} \right|}{\lambda_{j}} \right\}. \tag{9}$$

And the letters in formula (9) are the same as (6)–(8).

3.4. Optimization of VM Allocation Problems. In the optimization process, two schemes are proposed for users to choose. One is the scheme without considering the time factor, which only optimizes the cost and minimizes it while ignoring the time factor. It is suitable for users who do not care about time. Another kind is considering two factors of cost and time; the scheme firstly sets up a relatively small default time threshold. If the user has higher requirements, processing time can be dedicated to provide users with a time threshold that can be set up, after the cost optimization. The linear programming method is adopted, with the minimum cost as the optimization objective, and the processing time of resource price, virtual machine quantity, and supplier's supply capacity as the optimization conditions.

Formula (10) represents the optimization goal, namely, the overall cost; formula (11) indicates that the number of VMS obtained from the supplier in each usage phase does not exceed the number of VMS reserved. Equations (12)–(15) indicate that the total amount of basic resources obtained from each supplier should not exceed the upper limit of the maximum resources that the supplier can provide. Formula (15) indicates that the number of VMS in each stage of optimization is a nonnegative integer:

Minimize:

$$\sum_{p_j \in P} \sum_{v_i \in V} c_{j,i}^{[e]} X_{j,i}^{[e]} + \sum_{D \in \overline{D}} \sum_{p_j \in P} \sum_{v_i \in V} p(D)\left(c_{j,i}^{[u]} X_{j,i}^{[u]}(D) + c_{j,i}^{[a]} X_{j,i}^{[a]}(D) \right), \tag{10}$$

Subject to:

$$X_{j,i}^{(u)}(D) \le X_{j,i}^{(e)}, v_i \in V, p_i \in P, D \in \overline{D}, \tag{11}$$

$$\sum_{v_i \in V} n_{j,i}^{(a)}\left(X_{j,i}^{[u]}(D) + X_{j,i}^{[a]}(D) \right) \le s_j^{(a)}, \quad D \in \overline{D}, \tag{12}$$

$$\sum_{v_i \in V} n_{j,i}^{(b)}\left(X_{j,i}^{[u]}(D) + X_{j,i}^{[a]}(D) \right) \le s_j^{(b)}, \quad D \in \overline{D}, \tag{13}$$

$$\sum_{v_i \in V} n_{j,i}^{(c)}\left(X_{j,i}^{[u]}(D) + X_{j,i}^{[a]}(D) \right) \le s_j^{(c)}, \quad D \in \overline{D}, \tag{14}$$

$$X_{j,i}^{(e)}, X_{j,i}^{[u]}(D), X_{j,i}^{[a]}(D) \in \{\text{int eger}\}, \quad v_i \in V, p_i \in P, D \in \overline{D}n. \tag{15}$$

4. Experimental Results and Analysis

4.1. Introduction to Experimental Environment and Data Set. It includes a control node and four computing nodes [15] but due to limited hardware resources, this paper cannot provide a better cloud environment in the authentication algorithm, so a data center composed of 17 physical units is simulated. P1 serves as the controller node, and the other four compute nodes p2-P5 provide resources for users through virtualization [25]. As the algorithm in this paper optimizes the revenue of the cloud service provider and takes the maintenance cost of the cloud server into account in the virtual machine configuration stage, the cost quotation of the cloud server is set according to the pricing of Amazon EC2 virtual machine, as shown in Table 1.

4.2. Experimental Results Analysis. Figure 3 shows the experimental results of the difference in total revenue between the algorithm in this chapter and the utility optimization algorithm based on non-cooperative game and resource reserve calculation method in cloud service providers. As the time period increases, the total revenue of single bit time is equal to that of the comparison algorithm.

The main reason is that the algorithm makes a reasonable planning of resources in the virtual machine configuration stage in this chapter, so as to maximize the revenue of cloud services and optimize resource allocation. However, the utility optimization algorithm based on cooperative game cannot guarantee the optimal game. With the increase of time cycle, both the algorithm in this article and the game-based utility optimization algorithm maintain good returns under the effect of utility function optimization, while the resource reservation allocation algorithm starts to allocate on-demand resources with higher prices.

Through the simulation experiment, the algorithm in this article is compared with the noncooperative game-based utility optimization algorithm, resource reservation algorithm, and user utility on the unit revenue of cloud data center. 0.4 is selected as the interval, and the statistical utility value is between [1, 5]. The experimental results are shown in Figures 4 and 5, with user utility on the horizontal axis and unit revenue on the vertical axis Revenue represents the revenue generated per request processed.

The experimental results show that with the increase of user utility, unit income also increases gradually, but the increasing trend gradually becomes slow, indicating that the utility function has a certain promoting effect on the increase of unit income, but this promoting effect will begin to fail after reaching a certain peak. At the stage of stimulative effect, the increase of yield per unit area is manifested. After reaching the peak, the stimulative failure is mainly related to the resource itself. Therefore, resources will not increase the unit infinite revenue of the non-cooperative game optimization algorithm with the increase of the utility function, the profit per unit increases steadily at first and then falls sharply, indicating that on-demand resources play a decisive role in the allocation process.

In addition, the user utility difference between the algorithm in this article and the virtual machine instance based on noncooperative game utility optimization algorithm and resource reservation algorithm of the same user request is also compared as shown in Figure 6. The experimental results of actual satisfaction and maximum satisfaction ratio of users show that there is no significant difference between the proposed algorithm and other algorithms according to different users. Although the utility value differs greatly, the proposed algorithm is still effective on the whole. Combined with the analysis of the previous experimental results, it is proved that the algorithm in this article has higher revenue and resource utilization of cloud service providers and better overall performance without affecting user utility.

The experiment compares the performance of OpenStack random scheduling algorithm and the algorithm in this article in computing resource CPU utilization, as shown in Figure 7. The horizontal axis represents the number of random user tasks, and the vertical axis represents the CPU resource utilization. This kind of performance results from the large amount of idle resources in the initial stage of resource allocation, which are used by users with the increase of user tasks. The reason is that the algorithm in this article classifies user tasks and physical machines, and computationally intensive users are in the utility function during resource allocation. In this way, computing resources can be fully utilized to improve computing resource utilization.

Table 1: Cloud data center parameters.

Node	CPU (MIPS)	CPU cores	Memory (GB)	Bandwidth (M)
P1	3000	2	4	100
P2	2000	1	2	100
P3	1800	2	1	100
P4	2700	1	1	100
P5	2300	2	2	100

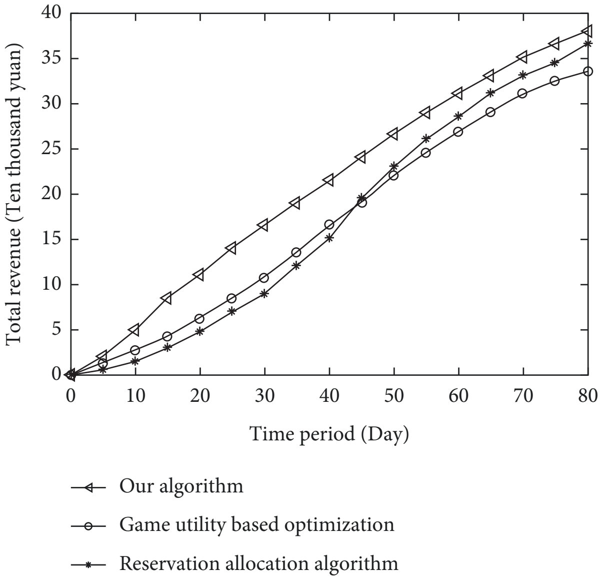

Figure 3: Total revenue comparison of cloud service providers.

The experimental results of memory resource utilization performance of OpenStack random scheduling algorithm and the algorithm in this article are compared as shown in Figure 8. The results in the Figure show that with the increase of the number of user tasks, resource utilization will increase steadily. The algorithm has a higher utilization of computing resources, but the difference between the two is not very large. The reason is that the host filtering in the Nova component of OpenStack gives priority to memory resource allocation and takes memory resource as the measurement standard. Therefore, the fluctuation of memory resource in the allocation process is smaller than that of the computing resource. Memory resource utilization has been improved to some extent.

Under the same initial conditions (the same initial state of VM distribution and the same load change of each application VM), the four algorithms are run to obtain the new distribution states of each VM, respectively. The number of migration times in the stable time and the number of idle physical nodes of these new distribution states are shown in Tables 2 and 3.

As can be seen in Tables 2 and 3, MOGANS makes a good balance between the stability of virtual machine distribution and the cost of virtual machine migration required

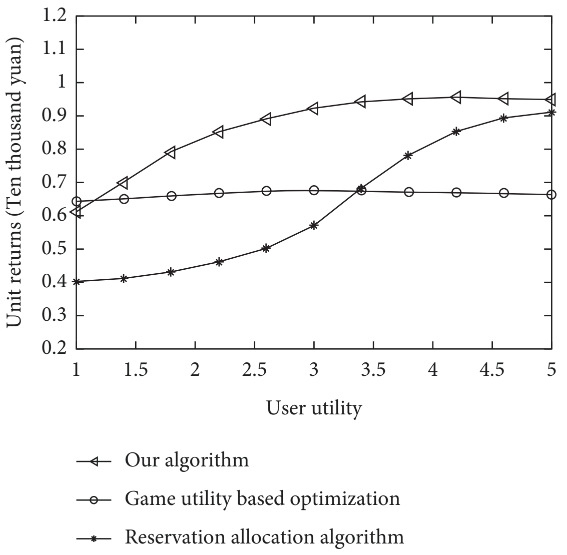

Figure 4: Comparison of unit revenue of different algorithms.

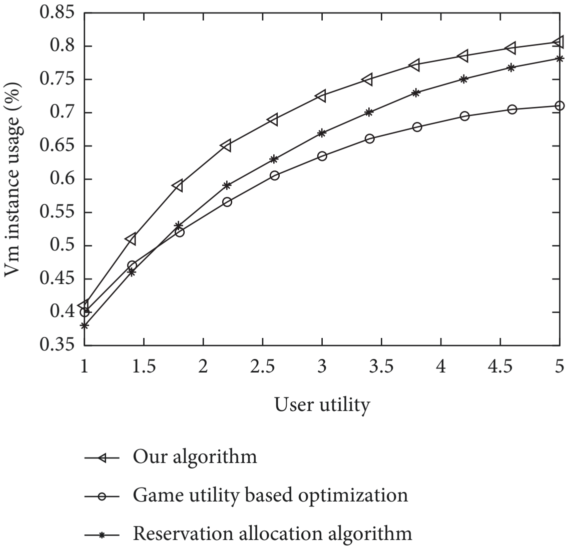

Figure 5: Comparison of VM instance usage.

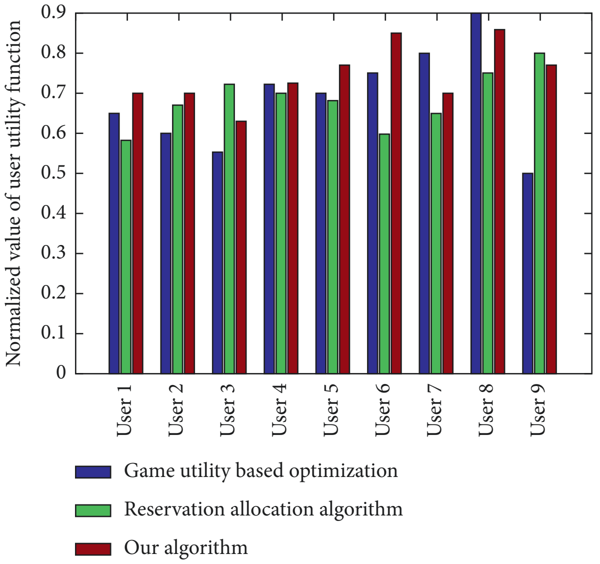

Figure 6: Comparison of utility values of different users under different algorithms.

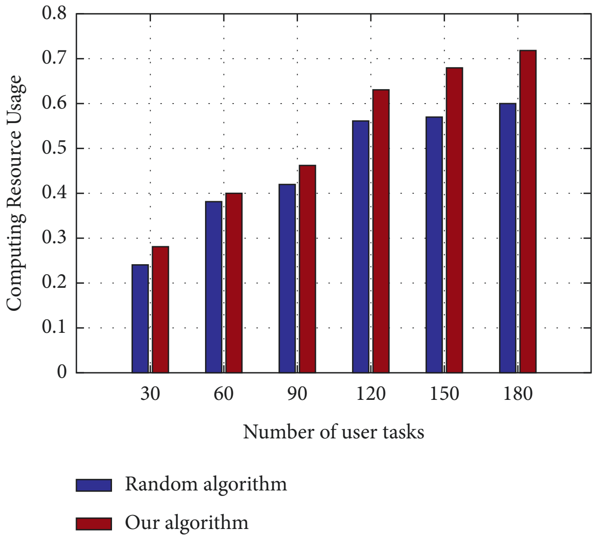

Figure 7: Comparison of computing resource usage.

by the old and new states, ensuring that the new multivirtual machine distribution state not only has a long stability time but also requires fewer virtual machine migration times from the current virtual machine distribution state to the new virtual machine distribution state. Although GA-N can change to a new VM distribution only after a minimum number of VM migrations, the new VM distribution can only maintain stability for a short time. Once the VM distribution becomes unstable, new dynamic resource configuration requests will be triggered, resulting in a new round of VM migration. Ga-NN takes green calculation into account, and the new virtual machine distribution state obtained uses the least active physical nodes, and the remaining number of free physical nodes is the largest, which is about 8 times the number of free physical nodes of MOGANS. Ga-NN takes green calculation into account, and the new virtual machine distribution state obtained uses the least active physical nodes, and the remaining number of free physical nodes is the largest, which is about 8 times the number of free physical nodes of MOGANS. This indicates that the new virtual machine distribution state obtained by GA-NN is more energy saving, but this energy saving state cannot be maintained for a long time. Once the distribution state becomes unstable, additional new dynamic resource allocation cost will be added, and the energy saving effect will be greatly reduced.

In order to compare the computational overhead between MOGANS algorithm and MOGAD algorithm, the experimental data tested in this section are basically the same as the parameters as the former experiment, including two variables: population size and population total algebra. The values of population size and population total algebra are constantly changed to measure the running time of the

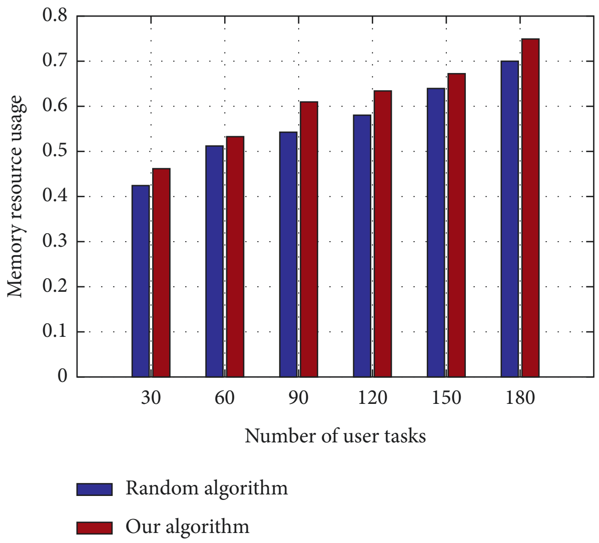

Figure 8: Comparison of memory resource usage.

Table 2: Algorithms' performance (16 physical nodes, 42 virtual machines).

Algorithms	Number of migration	Stable time	Number of idle physical nodes
MOGANS	21	15	1.6
GA-S	6.8	3.8	0.5
GA-N	9.2	1.2	3.7
GA-NN	22.5	0.9	5.6

Table 3: Algorithms' performance (32 physical nodes, 84 virtual machines).

Algorithms	Number of migration	Stable time	Number of idle physical nodes
MOGANS	68	15.2	1.4
GA-S	81.5	13.8	0.9
GA-N	19.2	1.2	5.7
GA-NN	72.5	0.8	15.6

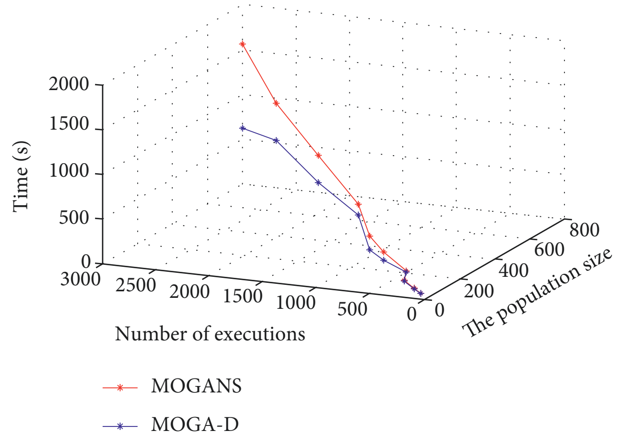

Figure 9: Comparison of running time cost between MOAGNS and MOGA-D.

algorithm, respectively. The experimental results are shown in Figure 9:

It can be seen from Figure 9 results that when the amount of data is small, the execution time cost of MOGANS and MOGA-D algorithm has little difference. When the amount of data increases sharply, the time cost of MOGA-D algorithm increases little. Therefore, when the amount of data is large, the calculation cost of MOGA-D algorithm is obviously superior as MOGANS has greatly improved the computational performance of the former.

5. Conclusions

Aiming at the deficiency of some current resource scheduling schemes of energy-saving resource scheduling in the current cloud computing environment, this paper proposes a multiobjective optimization scheme MOGANS based on NSGA-II multiobjective optimization.

However, MOGANS has insufficient computing performance, especially when the amount of data to be calculated increases, the performance of MOGANS is not satisfactory. After the test of the same amount of computation, the energy-saving resource scheduling scheme of the new opportunistic MOEA/D algorithm, MOGA-D, performs well. Compared with MOGANS, the computational performance is significantly improved with the same calculation scale. The similar multiobjective optimization results can also be obtained.

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 conflicts of interest.

Acknowledgments

This study was supported by Shanxi Province Educational Science "Thirteenth Five-Year Plan" Project (GH-19136) and Shanxi Province Higher Education Reform and Innovation Project (J2020380).

References

[1] L. Wang, G. Von Laszewski, A. Younge et al., "Cloud computing: a perspective study," *New Generation Computing*, vol. 28, no. 2, pp. 137–146, 2010.

[2] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, "Cloud computing virtualization of resources allocation for distributed systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 3, pp. 98–105, 2020.

[3] S. Namasudra, "Data access control in the cloud computing environment for bioinformatics," *International Journal of Applied Research in Bioinformatics*, vol. 11, no. 1, pp. 40–50, 2021.

[4] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[5] H. Hassan, A. I. El-Desouky, A. Ibrahim, E.-S. M. El-Kenawy, and R. Arnous, "Enhanced QoS-based model for trust assessment in cloud computing environment," *IEEE Access*, vol. 8, pp. 43752–43763, 2020.

[6] Z. S. Ageed, R. K. Ibrahim, and M. A. M. Sadeeq, "Unified ontology implementation of cloud computing for distributed systems," *Current Journal of Applied Science and Technology*, pp. 82–97, 2020.

[7] S. Namasudra, R. Chakraborty, and A. Majumder, "Securing multimedia by using DNA-based encryption in the cloud computing environment," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 3s, pp. 1–19, 2020.

[8] N. Rasouli, R. Razavi, and H. R. Faragardi, "EPBLA: energy-efficient consolidation of virtual machines using learning automata in cloud data centers," *Cluster Computing*, vol. 23, no. 4, pp. 3013–3027, 2020.

[9] V. Bogatyrev and A. Derkach, "Evaluation of a cyber-physical computing system with migration of virtual machines during continuous computing," *Computers*, vol. 9, no. 2, p. 42, 2020.

[10] C. Li, C. Wang, and Y. Luo, "An efficient scheduling optimization strategy for improving consistency maintenance in edge cloud environment," *The Journal of Supercomputing*, pp. 1–28, 2020.

[11] M. M. Shahriar Maswood, M. R. Rahman, A. G. Alharbi, and D. Medhi, "A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment," *IEEE Access*, vol. 8, pp. 113737–113750, 2020.

[12] S. Lou, J. He, H. Li et al., "Current knowledge and future directions for improving s quality and reducing energy consumption in conservation fields," *Agriculture*, vol. 11, no. 7, p. 575, 2021.

[13] S. Wilson and K. Platts, "The role of resource configuration on mix flexibility requirements," *Production Planning & Control*, vol. 20, no. 8, pp. 769–784, 2009.

[14] A. Khudoyberdiev, S. Ahmad, I. Ullah, and D. Kim, "An optimization scheme based on fuzzy logic control for efficient energy consumption in hydroponics environment," *Energies*, vol. 13, no. 2, p. 289, 2020.

[15] B. Baron, M. Campista, P. Spathis et al., "Virtualizing vehicular node resources: f," *Vehicular Communications*, vol. 4, pp. 39–46, 2016.

[16] S. Rawas, "Energy, network, and application-aware virtual machine placement model in SDN-enabled large scale cloud data centers," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 15541–15562, 2021.

[17] S. Namasudra, R. Chakraborty, S. Kadry, G. Manogaran, and B. S. Rawal, "FAST: fast accessing scheme for data transmission in cloud computing," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2430–2442, 2021.

[18] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information Fusion*, vol. 36, pp. 10–25, 2017.

[19] L. Wieclaw, V. Pasichnyk, and N. Kunanets, "Cloud computing technologies in "smart city" projects[C]//2017 9th IEEE International Conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)," *IEEE*, vol. 1, pp. 339–342, 2017.

[20] H. Shukur, S. R. M. Zeebaree, and A. Jamil Ahmed, "A state of art survey for concurrent computation and clustering of parallel computing for distributed systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 148–154, 2020.

[21] A. Kaur, P. Singh, and A. Nayyar, *Fog Computing: Building a Road to IoT with Fog Analytics*, Springer, Singapore, 2020.

[22] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications*, vol. 66, pp. 106–127, 2016.

[23] R. Meier, A. Rigo, and T. R. Gross, "Virtual machine design for parallel dynamic programming languages," *Proceedings of the ACM on Programming Languages*, vol. 2, pp. 1–25, 2018.

[24] M. E. Karim, M. M. S. Maswood, S. Das, and A. G. Alharbi, "BHyPreC: a novel Bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine," *IEEE Access*, vol. 9, pp. 131476–131495, 2021.

[25] J. Parmar, P. Das, and S. M. Dave, "Study on demand and characteristics of parking system in urban areas: a review," *Journal of Traffic and Transportation Engineering*, vol. 7, no. 1, pp. 111–124, 2020.

[26] D. B. Christie III, H. M. Nemec, A. M. Scott et al., "Early outcomes with utilization of tissue plasminogen activator in COVID-19-associated respiratory distress: a series of five cases," *Journal of Trauma and Acute Care Surgery*, vol. 89, no. 3, pp. 448–452, 2020.

[27] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 58, Article ID 100718, 2020.

[28] X. Zhang, H. Liu, and L. Tu, "A modified particle swarm optimization for multimodal multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 95, 2020.