Improved SLA based Scheduling Algorithm for Cloud

Akhil Kumar Sharma [1]*, Shayog Sharma[2] , Navneet Verma[3]
[1]M. Tech. Student, Computer Science Engineering Department,
[2]Assistant Professor, Computer Science Engineering Department,
[2]Head of Department , Computer Science Engineering Department,
Geeta Engineering College, Panipat, Haryana, India.

Abstract: *We propose a cloud service scheduling model that is referred to as the Task Scheduling System (TSS). In the user module, the process time of each task is in accordance with a general distribution. In the task scheduling module, we take a weighted sum of makes pan and flow time as the objective function and use an Ant Colony Optimization (ACO) and a Genetic Algorithm (GA) to solve the problem of cloud task scheduling. Simulation results show that the convergence speed and output performance of our Genetic Algorithm-Ant Colony Optimization (GA-ACO) are optimal.*

Keywords: AODV, OSLR, DOS, DSLR.

I. INTRODUCTION

In the present era technology play a very tremendous role in the human life. It makes the human life so easy. Now technology is the crucial part of human life. Without air, water and food human can't live his life now technology makes a space for him in this list. In the technology computer science and internet technology play a very wide and serious role. Every day new internet technology and computing devices are come in the market and proved his working style for human kind. [1] When Sir Tim Burners Lee was made his first web page in 1992 he didn't think that he will be inventor of new modern world which have power of internet technology. Now these days every work is perform by the help of internet technology and the services, which is providing using the help of internet technology.

Today a new technology spread in the IT market, which delivered the computing services to the client on demand in pay per use manner any time anywhere, but client must be connected to the high-speed internet connection to enjoy these type of services. This technology known as [2] "Cloud Computing". Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software and information are provided to users over the network. Cloud computing providers deliver application via the Internet, which are access from web browser, while the business software and data are stored on servers at a remote location. Cloud computing can provide three kinds of service modes, including IaaS, PaaS and SAAS [3]. Where SAAS is a service provided to client in terms of applications running on the cloud-computing infrastructure hosted by the service providers. PaaS refers to services, which provide high-level integrated environment to design, build, run, test, deploy and update the applications created by client using development language and tool say Java, python, .net etc. provided by the service providers to the cloud infrastructure. Iaas refers to the services provided to the users is to lease the processing power, storage, network and other basic computing resources, with which users can deploy and run any software including operating systems and applications. Beauty of cloud computing is that any time anywhere any one can access the services by paying the subscription charges to the service provider. Client has no need to worry about the service maintenance, updated version etc. They just subscribe the package and use it instantly without any separate hardware support but he must be his own computing device and high-speed internet connection.

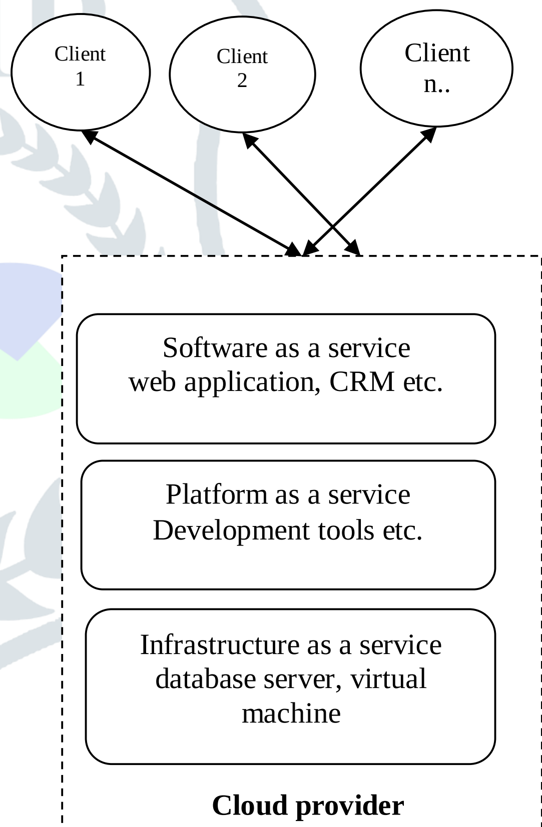

Fig 1. Cloud Provider Layout

Before the cloud-computing user have need to purchase the software in which the client want to work and then install it on a compatible machine and then use the software services and client have to maintain the software itself or by technology expert and update the software according to the need. In this process client lost the focus to his work and busy with the lower management task. Using the cloud computing technology user have no need to worry about these lower management task. They have to start from the main work without any worry about lower management task.

The rest of paper is design as follows. The overall past work is describe in Section II. Section III describes the framework of the implementation used for proposed work. Result discussion describe in section IV. Finally, Section V describes the conclusion of paper.

II. LITERATURE REVIEW

Many scholars and researchers studied a lot on Cloud scheduling, different researcher's does different works using variety of simulators and code to support their proposed studies.

Chen, C. H., Lin, J. W., & Kuo, S. Y., (2018) [8], Map Reduce is a software framework for processing data-intensive applications with a parallel manner in cloud computing systems. Some Map Reduce jobs have the deadline requirements for their job execution. The existing deadline-constrained Map Reduce scheduling schemes do not consider the following two problems: various node performance and dynamical task execution time. In this paper, we utilize the Bipartite Graph modelling to propose a new Map Reduce Scheduler called the BGMRS. The BGMRS can obtain the optimal solution of the deadline-constrained scheduling problem by transforming the problem into a well-known graph problem: minimum weighted bipartite matching.

Juarez, F., Ejarque, J., & Badia, R. M., (2018) [9], Traditional scheduling solutions attempt to minimize processing times without taking into account the energetic cost. One of the methods for reducing energy consumption is providing scheduling policies in order to allocate tasks on specific resources that impact over the processing times and energy consumption. In this paper, we propose a real-time dynamic scheduling system to execute efficiently task-based applications on distributed computing platforms in order to minimize the energy consumption. Scheduling tasks on multiprocessors is a well known NP-hard problem and optimal solution of these problems is not feasible, we present a polynomial-time algorithm that combines a set of heuristic rules and a resource allocation technique in order to get good solutions on an affordable time scale. The proposed algorithm minimizes a multi-objective function which combines the energy-consumption and execution time according to the energy-performance importance factor provided by the resource provider or user, also taking into account sequence-dependent setup times between tasks, setup times and down times for virtual machines (VM) and energy profiles for different architectures.

Latiff, M. S. A., Madni, S. H. H., & Abdullahi, M., (2018) [11], In cloud computing, resources are dynamically provisioned and delivered to users in a transparent manner automatically on-demand. Task execution failure is no longer accidental but a common characteristic of cloud computing environment. In recent times, a number of intelligent scheduling techniques have been used to address task scheduling issues in cloud . In this research article, we proposed a dynamic clustering league championship algorithm (DCLCA) scheduling technique for fault tolerance awareness to address cloud task execution which would reflect on the current available resources and reduce the untimely failure of autonomous tasks.

Experimental results show that our proposed technique produces remarkable fault reduction in task failure as measured in terms of failure rate. It also shows that the DCLCA outperformed the MTCT, MAXMIN, ant colony optimization and genetic algorithm-based NSGA-II by producing lower makespan with improvement of 57.8, 53.6, 24.3 and 13.4% in the first scenario and 60.0, 38.9, 31.5 and 31.2% in the second scenario, respectively. Considering the experimental results, DCLCA provides better quality fault tolerance aware scheduling that will help to improve the overall performance of the cloud environment.

Li, K., (2018) [12], Highlights Addressing energy and time constrained scheduling of precedence constrained parallel tasks. Deriving lower bounds for optimal solutions. Developing pre-power-determination algorithms and post-power-determination algorithms.Considering multiple manycore processors with continuous or discrete speed levels. Evaluating the performance of these algorithms analytically and experimentally When multiple many core processors in a data center for cloud computing are shared by a large number parallel tasks simultaneously, we are facing the problem of allocating the cores to the tasks and scheduling the tasks, such that the system performance is optimized or the energy consumption is minimized. Furthermore, such core allocation and task scheduling should be conducted with energy constraints or performance constraints. The problems of energy and time constrained scheduling of precedence constrained parallel tasks on multiple manycore processors in a cloud computing environment are defined as optimization problems. Lower bounds for optimal solutions are generalized from a single parallel computing system to multiple parallel computing systems. Our approach in this paper is to design and analyze the performance of heuristic algorithms that employ the equal-speed method

Wei, W., Fan, X., Song, H., Fan, X., & Yang, J., (2018) [13], Existing static grid resource scheduling algorithms, which are limited to minimizing the makespan, cannot meet the needs of resource scheduling required by cloud computing. Current cloud infrastructure solutions provide operational support at the level of resource infrastructure only. When hardware resources form the virtual resource pool, virtual machines are deployed for use transparently. Considering the competing characteristics of multi-tenant environments in cloud computing, this paper proposes a cloud resource allocation model based on an imperfect information Stackelberg game (CSAM-IISG) using a hidden Markov model (HMM) in a cloud computing environment. CSAM-IISG was shown to increase the profit of both the resource supplier and the applicant. Firstly, we used the HMM to predict the service provider's current bid using the historical resources based on demand. The proposed resource allocation model can support synchronous allocation for both multi-service providers and various resources. The simulation results demonstrated that the predicted price was close to the actual transaction price, which was lower than the actual value in the game model. The proposed model was shown to increase the profits of service providers and infrastructure suppliers simultaneously.

Basu, S., & Anand, A., (2019) [14], Cloud computing is one of the highly debated topics in todays IT as well as academic research domain. The balance between performance and security has always been a vital issue when adapting cloud-based services. Task scheduling is a major research area of cloud computing, where efficient algorithms are devised for service providers to allocate a task to a particular virtual machine, thereby enhancing the performance as well as execution time of the tasks. Though QoS has been kept in mind while designing such schemes, the aspect of security has been majorly ignored in the existing works. Here a secured task scheduling methodology has been devised by using the test environment of CloudSim 3.0.3.

Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z., (2019) [15], Mobile cloud computing (MCC) as an emerging and prospective computing paradigm, can significantly enhance computation capability and save energy for smart mobile devices (SMDs) by offloading computation-intensive tasks from resource-constrained SMDs onto resource-rich cloud. However, how to achieve energy-efficient computation offloading under hard constraint for application completion time remains a challenge. To address such a challenge, in this paper, we provide an energy-efficient dynamic offloading and resource scheduling (eDors) policy to reduce energy consumption and shorten application completion time.. Finally, we provide experimental results in a real testbed and demonstrate that the eDors algorithm can effectively reduce EEC by optimally adjusting CPU clock frequency of SMDs in local computing, and adapting the transmission power for wireless channel conditions in cloud computing.

Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z., (2019) [16], Mobile cloud computing (MCC) as an emerging and prospective computing paradigm, can significantly enhance computation capability and save energy for smart mobile devices (SMDs) by offloading computation-intensive tasks from resource-constrained SMDs onto resource-rich cloud. However, how to achieve energy-efficient computation offloading under hard constraint for application completion time remains a challenge. To address such a challenge, in this paper, we provide an energy-efficient dynamic offloading and resource scheduling (eDors) policy to reduce energy consumption and shorten application completion time. We first formulate the eDors problem into an energy-efficiency cost (EEC) minimization problem while satisfying task-dependency requirement and completion time deadline constraint.

Li, Z., Li, Y., Yuan, T., Chen, S., & Jiang, S., (2019) [17], With the development of virtualization technologies, cloud data centers are faced with more and more virtual machines (VMs) requests. How to realize an efficient virtual machine placement (VMP) becomes a hot research topic. The optimal resource consumption and the utilization rate of a physical machine in the whole cloud data center can be realized by optimizing the process from the virtual machine to the physical machine. VMP is a combinatorial optimization problem which was demonstrated to be NP-hard. The proposed algorithms are compared with other optimal placement strategies, namely Cuckoo Search Optimization (CSO), Reordered Grouping Genetic Algorithm (RGGA), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD). Experimental results show that the proposed CVP and CVV give better performance comparing with the other compared algorithms in terms of resource consumption and resource utilization. In term of scalability, the proposed CVV algorithm benefits from the high computational speed and performs well when there are a large number of virtual machine scheduling requests in the cloud data center.

Naik, K., Gandhi, G. M., & Patil, S. H., (2019) [18], In cloud Infrastructure as a Service (IaaS) environment, selecting the Virtual Machines (VM) from different data centers, with multiple objectives like reduction in response time, minimization in cost and energy consumption, is a complex issue due to the heterogeneity of the services in terms of resources and technology. The existing solutions are computationally intensive; rely heavily on obtaining single trade-off solution.

The author describe the new hybrid multi objective heuristic algorithm based on Non-dominated Sorting Genetic Algorithm-II (NSGA-II) and Gravitational Search Algorithm (GSA) called as NSGA-II & GSA to facilitate selection of VM for scheduling of an application. The simulation results show that the proposed algorithm outperforms and fulfills the prescribed objective as compared to other multi objective scheduling algorithms.

III. Proposed Methodology

3.1. Genetic Algorithm (GA). For an optimization problem, a number of candidate solutions that are called individuals and whose abstract representations are named chromosomes, are improved in each iteration and eventually evolve to the optimum solution. The Genetic Algorithm (GA) uses a selection operation to determine the survival for each individual. That is, the fitness of the whole population is evaluated. Multiple individuals are chosen randomly from the current population based on their fitness, and individuals with high fitness are inherited in the next generation with a higher probability; on the contrary, for low fitness individuals, the probability of being inherited in the next generation is smaller. For each generation, a new population is generated according to the natural selection process described above and mutation.

3.1.1. Selection Operation. The roulette selection [20, 21] method is the most commonly used method for selecting individuals in the natural selection process just outlined above. As shown in Figure 3, the basic idea of a selection operator, also known as proportional selection, is to set the probability of an individual being selected in proportion to its fitness function value. If we imagine that the population size is $N$and the fitness of an individual xi is (xi), then probability xi that will be selected by our operator is

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

Assuming that every individual's chromosome represents a small sector on a wheel and its size is proportional to the fitness of the chromosome, then individuals with better adaptability are more likely to be chosen.

3.1.2. Genetic Operations. In order to promote evolution of populations and speed up convergence towards the optimal solutions, as well as to increase the diversity of the population, we need to perform genetic operations such as crossover and mutation on the individuals selected according to the roulette selection discussed previously.

(a) Crossover operation

Two chromosomes interchange part of their genes with each other based on the crossover probability, forming two new individuals.

Before crossover:

01000|0100001000
11000|1111000010

After crossover:

01000|1111000010
11000|0100001000

(b) Mutation operation

Mutation changes the code string of a chromosome of an individual to a new value of certain gene. The mutation operation is a good method to maintain population diversity.

Before mutation:

01000|111100**0**010

After mutation

01000|111100**1**010

Step 1. Initialization (configures parameters, initialize VCN)
Step 2. While iteration t < maximum iterations do:
Calculate fitness value of every individual;
Select individuals using roulette rotation method;
Crossover and mutate individuals which has been selected to produce new individuals;
Step 3. End while
Step 4. Return best solution

Fig 2 Genetic Algorithm

Step 1. Initialization (configures parameters, initialize QOT, VCN, and initialize the initial position of all ants)
Step 2. While iteration t < maximum iterations do:
Calculate the next position of ants according to the equation (8).
Calculate the fitness value of every candidate solution.
Update pheromone matrix according to equation (10).
Step 3. End while
Step 4. Select the optimal solution and return.

Fig 3 ACO Algorithm

The two algorithm ACO & genetic algorithm are given in figure 2 & 3 respectively.

IV. Result Analysis

The result of cloud scheduling is concluded in terms of different weight task , execution time , estimated cost etc.

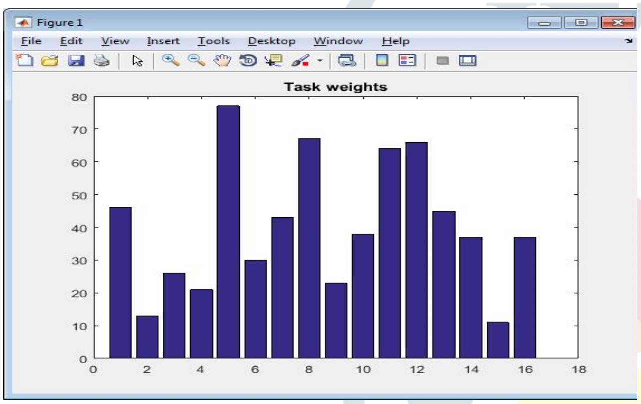

Fig 4 Different Weight Task

There are 16 parallel task are to be performed for cloud scheduling. The different weight task is shown in form of bar graph as shown in the fig 4. The graph indicates that 5 task have highest weight & 17 task having minimum weight.

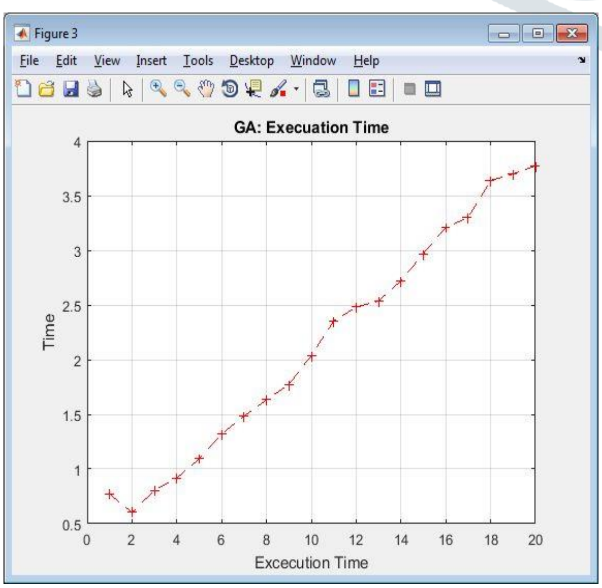

Fig 5. Genetic Algorithm Execution Time

The execution time of cloud scheduling is shown in fig 5. It indicates that execution time is decrease using genetic algorithm.

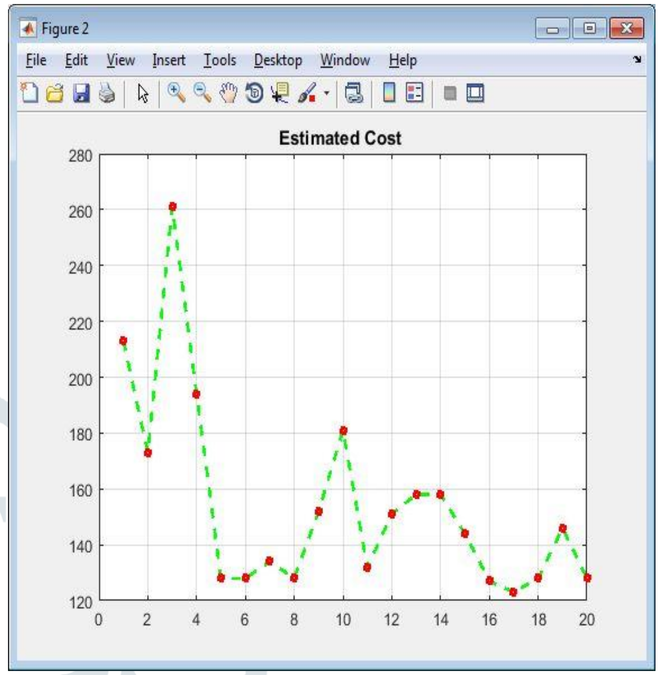

Fig 6. Estimated Cost for cloud scheduling

Estimated cost for cloud scheduling for 20 iterations is shown in fig 6. Maximum estimated cost is 260. & minimum is 120.

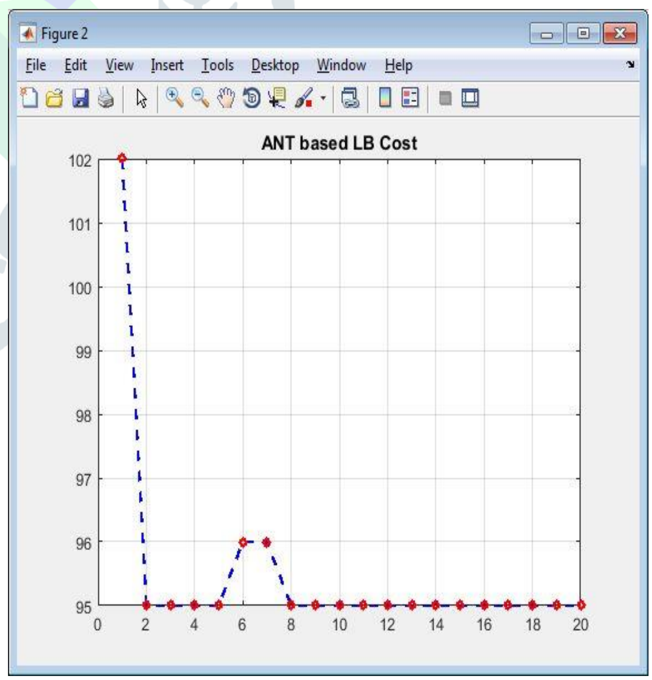

Fig 7 ANT based LB Cost

ANT based LB Cost performance is as shown in figure 7. The performance is defined as shown in figure. The point of indications is shown in figure.

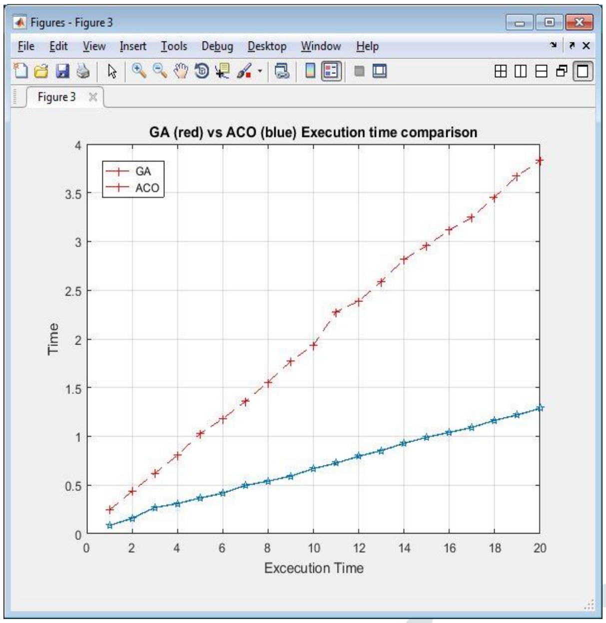

Fig 8 Comparative Analysis between ACO & GA for Execution time

The comparative analysis of both the algorithm is show in fig 8. The performance of Genetic & Ant colony optimization is as shown in fig 8.

V. Conclusion

In this paper, we present a cloud service task scheduling model TSS, which consists of three modules: a user module, a data center module, and a task scheduling module. In the user module, we assume that the tasks model is a Poisson process. In the task scheduling module, because of the deficiencies of GA and ACO, we have proposed a GA-CACO algorithm, Which merges GA and CACO. And finally, we compared GACACO, GA, and ACO for several different task sizes. The results indicate that GA-CACO algorithm is optimal for the optimization of the objective function and also has acceptable convergence speed.

References

[1] Chen, C. H., Lin, J. W., & Kuo, S. Y. (2018). MapReduce scheduling for deadline-constrained jobs in heterogeneous cloud computing systems. IEEE Transactions on Cloud Computing, 6(1), 127-140.

[2] Juarez, F., Ejarque, J., & Badia, R. M. (2018). Dynamic energy-aware scheduling for parallel task-based application in cloud computing. Future Generation Computer Systems, 78, 257-271.

[3] Juarez, F., Ejarque, J., & Badia, R. M. (2018). Dynamic energy-aware scheduling for parallel task-based application in cloud computing. Future Generation Computer Systems, 78, 257-271.

[4] Latiff, M. S. A., Madni, S. H. H., & Abdullahi, M. (2018). Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. Neural Computing and Applications, 29(1), 279-293.

[5] Li, K. (2018). Scheduling parallel tasks with energy and time constraints on multiple manycore processors in a cloud computing environment. Future generation computer systems, 82, 591-605.

[6] Wei, W., Fan, X., Song, H., Fan, X., & Yang, J. (2018). Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing. IEEE Transactions on Services Computing, 11(1), 78-89.

[7] Basu, S., & Anand, A. (2019). Location Based Secured Task Scheduling in Cloud. In Information and Communication Technology for Intelligent Systems (pp. 61-69). Springer, Singapore.

[8] Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z. (2019). Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing. IEEE Transactions on Mobile Computing, 18(2), 319-333.

[9] Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z. (2019). Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing. IEEE Transactions on Mobile Computing, 18(2), 319-333.

[10] Li, Z., Li, Y., Yuan, T., Chen, S., & Jiang, S. (2019). Chemical reaction optimization for virtual machine placement in cloud computing. Applied Intelligence, 49(1), 220-232.

[11] Naik, K., Gandhi, G. M., & Patil, S. H. (2019). Multiobjective virtual machine selection for task scheduling in cloud computing. In Computational Intelligence: Theories, Applications and Future Directions-Volume I (pp. 319-331). Springer, Singapore.

[12] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2008.

[13] P. Wang, S. Huang, and Z.-Q. Zhu, "Swarm intelligence algorithms for circles packing problem with equilibrium constraints," in *Proceedings of the 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science (DCABES '13)*, pp. 55–60, IEEE, Los Alamitos, Calif, USA, September 2013.

[14] X. Huang, G. Liu, W. Guo, and G. Chen, "Obstacle-avoiding octagonal steiner tree construction based on particle swarm optimization," in *Proceedings of the 9th International Conference on Natural Computation (ICNC '13)*, pp. 539–543, Shenyang, China, July 2013.

[15] C. Wang, J. Zhang, J. Yang et al., "A modified particle swarm optimization algorithm and its application for solving traveling salesman problem," in *Proceedings of the International Conference on Neural Networks and Brain (ICNN&B '05)*, vol. 2, pp. 689–694, Beijing, China, October 2005.

[16] Y. M. Wang and H. L. Yin, "A two-stage approach based on genetic algorithm for large size flow shop scheduling problem," in *Proceedings of the 10th IEEE International Conference on Mechatronics and Automation (IEEE ICMA '13)*, pp. 376–381, Takamatsu, Japan, August 2013.

[17] J. Li and J. Peng, "Task scheduling algorithm based on improved genetic algorithm in cloud computing environment," *Journal of Computer Applications*, vol. 31, no. 1, pp. 184–186, 2011.

[18] N. Srinivas and K. Deb, "Muiltiobjective optimization using Non dominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.