



An efficient and scalable hybrid task scheduling approach for cloud environment

Sunita Rani¹ · P. K. Suri²

Received: 23 August 2017 / Accepted: 13 April 2018 / Published online: 19 April 2018
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2018

Abstract Cloud Computing is the new style of computing in the field of IT. Cloud computing is being vital on the internet as it shares various computing resources instead having personal devices to manage data and applications over internet. There is a lot of data stored on cloud and various resources requests for the same. The data and applications are maintained in the cloud computing by making the use of internet. It requires computing facilities on a large scale that depends on usage and to provide services in a very adjustable manner which may move up and down according to user demand. For cloud service providers, to provide the resources to the users in time is one of the tedious tasks. To comply this reason a proper node is to be selected that can complete the tasks for the users while maintaining quality of service. This paper proposed a hybrid algorithm by merging the gravitational search concept in ant colony optimization algorithm. The main idea behind proposed algorithm is its unique search approach that is being used for achieving task scheduling in cloud computing by allocating the incoming tasks to the virtual machines, thereby decreasing the makespan. The CloudSim toolkit package is used to simulate the algorithm and the results revealed better performance than the basic ant colony optimization and basic gravitational search algorithm.

Keywords Cloud computing · Task scheduling · Ant colony optimization · CloudSim · Gravitational search algorithm

1 Introduction

Cloud computing has emerged as an advancement in the field of IT. Cloud has data and resources saved on a shared pool that can be accessed by end users on demand following the model of pay per usage. The end users just have to pay the fees for the services they desire through the Internet.

In cloud computing environment three types of clouds: public cloud, private cloud and hybrid cloud exists to provide the services to the users. The cloud in which the data and services are used by the public over the internet is known as public cloud. The cloud in which services are used privately by some particular organization is called Private cloud. The combination of two-public and private is the Hybrid cloud. It combines the features of both by providing multiple types of data and resources to the users [1].

In the situation, when the users submit a large number of task requests, the virtual machines in the cloud environment have to not only handle immense tasks but also have to satisfy the demand of user's service [2].

While handling the users' requests sometimes in cloud computing the efficiency and utilization of resources is declined whereas completion time and cost are increased. Task scheduling is done to optimize these parameters and to increase the systems' throughput. The process by which resources are assign tasks is termed as scheduling. The main aim of scheduling is to allot the resources to the tasks in the best possible way. This is done to optimize the

✉ Sunita Rani
sunita.bpsmv@gmail.com

P. K. Suri
pksturikuk@gmail.com

¹ Department of CSE and IT, BPSMV, Khanpurkalan, Sonapat, Haryana, India

² Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India

service provided by the cloud. This also enhances the efficiency of the cloud. Scheduling involves how, when and which task will be executed using which resources [3].

The authors have organized the paper in six sections. Section 2 includes the background and the related work. Section 3 describes the CloudSim tool kit. Section 4 explains the details of the proposed algorithm. Highlights of the experimental and simulation results are covered in Sect. 5. Section 6 concludes the work done in this paper.

2 Related work

In this section, authors have discussed the different variants of ant colony optimization and gravitational search technique based task scheduling approach for cloud computing proposed by the other researchers. Details are as follows:

Ant colony optimization for load balancing has been discussed in [4]. It works by searching the node with minimum load. The main feature of load balancing is to make sure that all the processor in the system or network is assigned the equal quantity of work at any particular moment. The above said load balancing technique has been tested using the simulator named Cloud Analyst. When experimental results were studied for a sample application, they were better than the usual approaches like Stochastic Hill Climbing (SHC) and First Come First Serve (FCFS).

Raut et al. [5], have introduced an algorithm based on the ant colony optimization (ACO) that have shown improved performance of cloud computing along with optimally utilizing the resources. This technique detects the overloaded nodes on data centre and distributes it by taking threshold value. It distributes the load to the other nodes by finding the nearest under loaded node using travelling salesman problem (TSP). The proposed algorithm gives a better response in less time. If the machine's RAM and Processor is good, then this algorithm can perform a better load balancing compared to only using ACO. The proposed technique works to ensure availability, to fully utilize the available resources, to handle the maximum number of requests by cloud and to minimize the time required to serve multiple requests. If the requests are already in temporary memory, then it executes in less waiting time.

Li et al. [6] have achieved task scheduling with load balancing. Authors have projected the Load Balancing Ant Colony Optimization (LBACO) algorithm. When the algorithm was tested for the gradually increasing number of tasks, the results shows that the proposed algorithm assigns equal load to the entire system, irrespective of the sizes of the tasks. The results show that LBACO has also performed significantly better than FCFS and ACO algorithms in the field of cloud computing. The investigations made in this paper was done assuming that all tasks are

mutually independent, i.e., there is no precedence constraint between tasks and it was also assumed that tasks are thoroughly calculated, which is not practical in cloud environment. So, this area is yet to be analyzed.

Taufeek et al. [7] have presented an ACO algorithm, a unique search approach that is being used for achieving task scheduling in cloud computing by allocating the incoming tasks to the virtual machines. The best results were obtained on experimental analysis of the algorithm taking the increasing number of tasks on various parameters. Like LBACO the simulation results have shown that ACO algorithm also performed better than Round-Robin (RR) algorithms and first come first serve (FCFS) algorithm. The effect of priority between the tasks and load balancing and the comparison between our search approach and other existing approaches is yet to be carried out.

Mishra et al. [8] have proposed a precise algorithm based on ant colony optimization to distribute the load in cloud computing environment. The load was balanced by the pheromone (chemical secreted by ants to guide others) update mechanism. Ant colony optimization algorithm was used to optimize the performance parameters like load on the processing device, capacity of the memory, and waiting time. Besides the fact that this technique minimizes the make span of the cloud computing based services and portability of servicing the request also has been achieved but the issues related to fault tolerance are not considered by this technique.

Jain et al. [9] have proposed a new load balancing approach for cloud computing. Proposed approach has used the concept of biased random walk. Biasing has been achieved through task size, and available capacity of virtual machine. Proposed approach has not only improved the load balancing but also improved the reliability of the system.

Kalra et al. [10] have compared and analyzed various scheduling algorithms based on Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), league Championship Algorithm (LCA) and BAT algorithm. Analysis of the discussed algorithms includes the comparison among the techniques in order to find the optimal solution in short span of time. The background researches show that the data centers consume a lot of energy along with liberating CO₂. For future work the authors have accepted the challenge to reduce energy consumption of data centers without degrading performance and violating SLA constraints.

Sharma et al. [11] have proposed a hybrid of Differential algorithm with Gravitational search algorithm to overcome the problem of weak local searching of Differential algorithm. Although DE algorithm can produce the optimum results while scheduling the tasks but rate of congregation is slow that has been improved by using GS algorithm. The

comparative analysis of the experimental results with other algorithms show that the proposed DE-GSA algorithm has improved performance in terms of the minimum completion time of tasks along with the optimum results in task scheduling. In future, the proposed algorithm can be used to optimize other parameters and to solve the issues related to task limit by changing the objective function.

Along with the load balancing, the tasks finishing time and the required delivery dates are other issues related to scheduling that have to be resolved. Tavakkolai et al. [12] have proposed a method using gravitational search algorithm to solve the above stated problems in limited resources. Mathematical analysis of the proposed model reveals the excellent results over GA algorithm. As a future work the advantages of this algorithm can be combined with other algorithms to form a combinatorial algorithm. The finishing time of the tasks and the utilization of resources in the best possible way is very important for solving the task scheduling problem in workflow workshop environment keeping the sum of tasks delays minimized and on required times for in advance reservation of resources.

Rawal et al. [13] discussed the study of various CPU task scheduling algorithms. The main aim to propose this gravitational search based algorithm is to optimize the cost parameters related to task scheduling. Analysis have revealed the better performance when compared with traditional GA algorithm as the total cost, waiting time and delay comes out to be very less in case of GSA than GA by keeping the number of processors and execution time constant.

Kumar et al. [14] presented a hybrid load balancing approach for cloud environment by combining the best feature of join idle queue, join shortest queue and minimum completion time approach. Moreover, authors have added the prior overloading checking mechanism. Authors have tested the proposed approach on cloud analyst simulator and it has been found that proposed hybrid approach JIMC has outperformed all the basic approach on all the relevant parameter.

3 Simulator-CloudSim toolkit [7, 15]

CloudSim is free and open source software available at <http://www.cloudbus.org/CloudSim/> that is used for the simulation purpose. Simulation is a process in which designer simulates the behavior of different components by identifying the interaction among the different units through mathematical equations.

3.1 Cloud simulator characteristics

CloudSim offers the following characteristics which has made the life of researcher easy.

- It supports large scale Cloud computing model
- Act as a base for modeling data centers, service brokers, scheduling, and allocations policies.
- It enhances the services by allowing them to be space-shared and time-shared
- Cloud supports VM provisioning at two levels:
 - At the host level—It is possible to specify how much of the overall processing power of each core will be assigned to each VM.
 - At the VM level—The VM assigns a fixed amount of the available processing power to the individual application services (task units) that are hosted within its execution engine (Known as VM Scheduling).

3.2 The CloudSim platform

The different components of cloudsim simulator and their connectivity has been shown in Fig. 1.

- *Cloud Information Service* A unit which maintains the list of available resources on this cloud.
- *Data center Broker* Broker is another entity which assigns cloudlet (task) to data center.

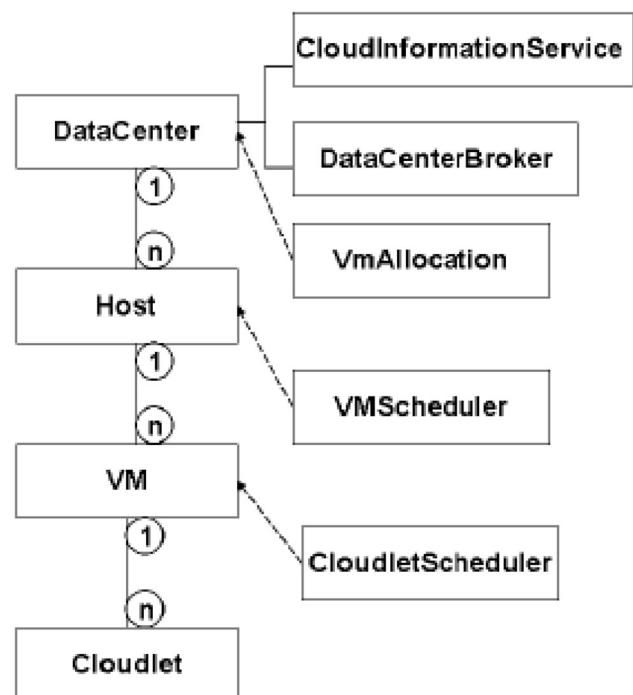


Fig. 1 Main parts of Cloudsim

- *Host* It represents a physical computing machine or server in a Cloud data center.
- *Vm* is managed and hosted by a Cloud host component.
- *Vm allocation* It is used to create different instances of VM on a host
- *Data center* It manages several hosts that in turn manage VMs during their life cycles.
- *VmScheduler* This is processed by a Host component that allocates processor cores to VMs.
- *Cloudlet* It represent the task.
- *CloudletScheduler* It divides the processing power among various tasks.

4 Proposed algorithm

ACO is an evolutionary algorithm which can be used to solve NP-hard optimization problems. In 1996, Dorigo has given this approach by simulating the food searching technique of ants. While searching the best food source, ant keep on leaving the pheromone on the path traced by them. Size of pheromone indicates the quality and distance of food source from their source. While searching the food source, if ant does not find any pheromone then it selects the path randomly. Follower ant trace that path on which maximum number of good quality pheromones are available. Quantity of path will keep on increasing on the path which is approaching to food source. Pheromone acts as positive feedback mechanism for the follower ants.

Now authors have used the same concept for task scheduling in cloud computing. But the problem associated with this basic approach is that it results in overloading of the single source and other sources are not utilized as efficiently. This problem is called premature convergence. So to overcome this inherent problem of ACO, authors have combined the ACO with gravitational search algorithm. Food source has been replaced by virtual machines and concept of pheromone help in finding the best virtual machine. But in proposed approach, pheromone does not convey the information as in traditional ACO. To update the pheromone, gravitational search algorithm has been used. This avoids the overloading of any specific machine and results in equal distribution of load.

4.1 Notations

$\tau_{i,j}(t)$ represents the quality of path between i -th and j -th machine. In the beginning all paths have the same quality. This quality is similar to pheromone and its value may depend upon quality of food/services at j .

4.2 Steps of proposed ACO with GSA approach is as follows:

- Step 1 Initialize the value of heuristic factors and pheromone. In the beginning $\tau_{i,j}(t)$ will depend upon the computing capacity of machines. So $\tau_{i,j}(t) = C_j$.
- Step 2 In the beginning when no pheromones is placed, random path is traced to map task with machine.
- Step 3 Using gravitational search algorithm, update the pheromone. Use of GSA with ACO at this step will make this algorithm different from basic ACO. As pheromone express the quality and distance of machine from that point where it is lying. In tradition ACO, no of pending tasks allotted to machine and capacity of machine is not considered. But here, we have used the concept of gravitational search algorithm to calculate the power of virtual machine. This will consider the number of tasks in the queue as well as their size. Pheromone will consider all these values, to update itself. In this way proposed approach avoids the premature convergence.

$$\tau(t+1) = (1-\rho)\tau_{i,j}(t) + \Delta\tau_{i,j}(t),$$

where ρ represents the factor of power consumed. Higher the value of this factor, lower will be the chances of task receiving by the machine. Value of ρ is calculated using gravitational search algorithm.

$$\rho = \left\{ \frac{\text{Sum of all tasks size in queue of } j\text{-th machine}}{\text{Processing capacity of } j\text{-th machine}} \right\}.$$

- If no task is there in queue then $\rho = 0$.
- Step 4 Trace the path from the pheromone till it reaches at highest value pheromone.
- Step 5 Map the task with the virtual machine corresponding to selected pheromone.
- Step 6 Go back to step 3.

Algorithmic representation of steps are as follows:

```

Algorithm ACO_GSA()
{
    Find the list of available virtual machine and
    store the count in N

    For (j=1 to N)
    {
        Identify the following particulars and
        store them for j th machine
        > Processing Capacity
        > No of processing cores
        > Storage capacity
        > Ram
    }

    Using the Poisson distribution, generate the M
    number of tasks

    For (j=1 to M)
    {
        Initialize the following task
        parameter for the j-th task
        > Task size
        > Arrival time
        > Resource requirement
    }

    Launch the Pheromones.

    While (pending task is there in the data center)
    {
        For (j=1 to M)
        {
            • Using Gravitation
            search algorithm,
            power consumed
            factor  $\rho$  for each
            machine is
            calculated

            • Pheromone values
            are updated using
            the following
            formula:

             $\tau(t+1) = (1 - \rho) \tau_{ij}(t) + \Delta\tau_{ij}(t)$ 
        }

            • Now to identify the best
            available virtual machine
            based upon its configuration
            and no of tasks in queue and
            their size, path from lower
            value pheromone to higher
            value is traced, till we reach
            at a point where we reach at
            the highest value
            pheromone.
            • Virtual machine is identified
            corresponding to highest
            value pheromone.
            • Task is assigned to the
            selected virtual machine.
        }
    }
}
    
```

$$\rho = \left\lfloor \frac{\text{Sum of all tasks size in queue of } j - \text{th machine}}{\text{Processing capacity of } j - \text{th machine}} \right\rfloor$$

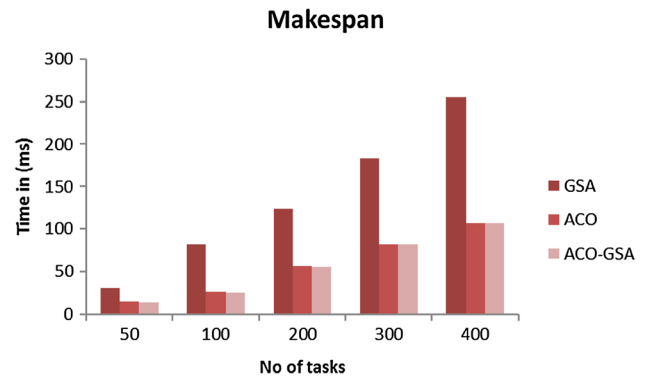


Fig. 2 Comparison on the scale of makespan

5 Result and analysis

Proposed hybrid approach has been tested on CloudSim simulator under different number of tasks while keeping the number of resources same. Results have been analyzed on the relevant parameters like makespan, throughput, load balancing percentage and percentage of resource utilization. Detail comparative analysis of proposed hybrid of ACO and GSA with the basic ACO and GAS are as follows:

5.1 Analysis

If we analyze the results shown in Figs. 2, 3, 4 and 5, it can be concluded that Hybrid approach is performing very well related to GSA and perform marginally well relative to ACO.

In Fig. 6, performance of proposed approach on the scale of makespan with the varying number of tasks has been analyzed. It has been found that makespan is increasing in the same ratio in which the number of tasks has increased.

In Figs. 7, 8 and 9, performance of proposed approach with varying number of tasks has been analyzed on the

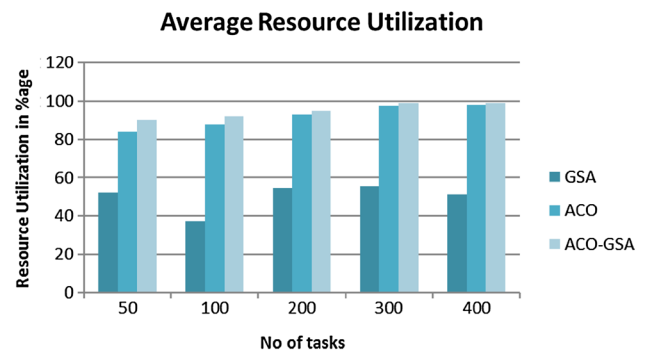


Fig. 3 Comparison on the scale of resource utilization

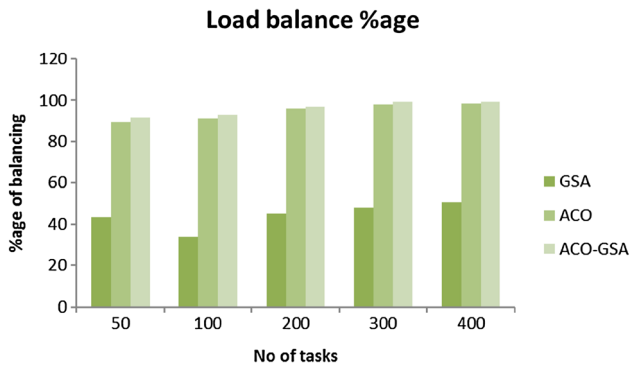


Fig. 4 Comparison on the scale of load balance percentage

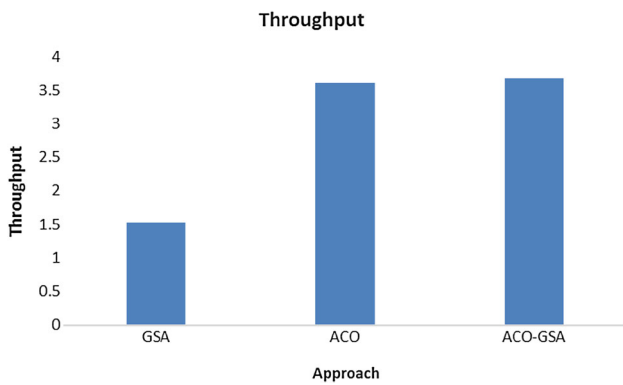


Fig. 5 Comparison on the scale of load balance percentage

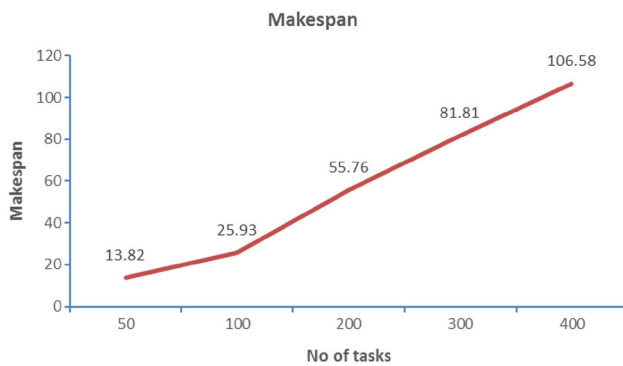


Fig. 6 Performance of proposed approach on the scale of makespan with the change of tasks

scale of resource utilization, percentage of load balancing and throughput. It has been found that performance of proposed approach is upgrading with the increase in number of tasks. This indicates the stable, scalable and distributed nature of proposed approach which makes it suitable for large distributed computing environment like cloud.

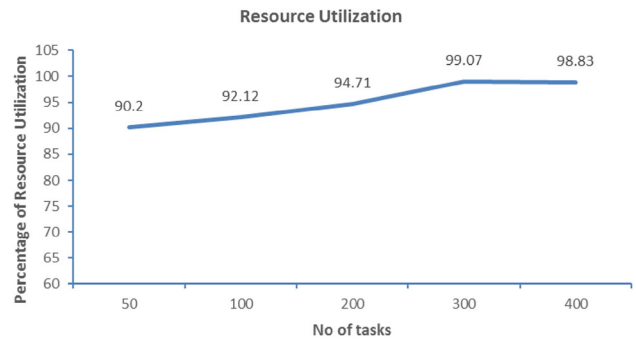


Fig. 7 Performance of proposed approach on the scale of resource utilization with the change of tasks

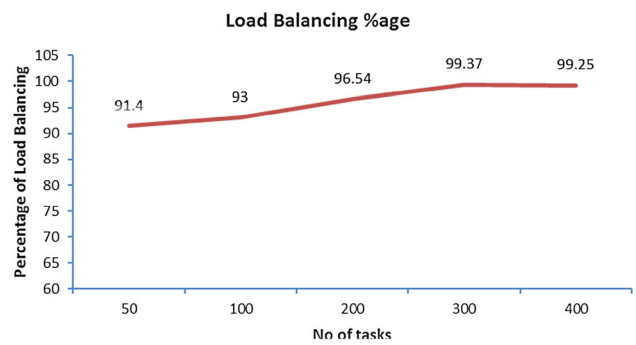


Fig. 8 Performance of proposed approach on the scale of load balancing percentage with the change of tasks

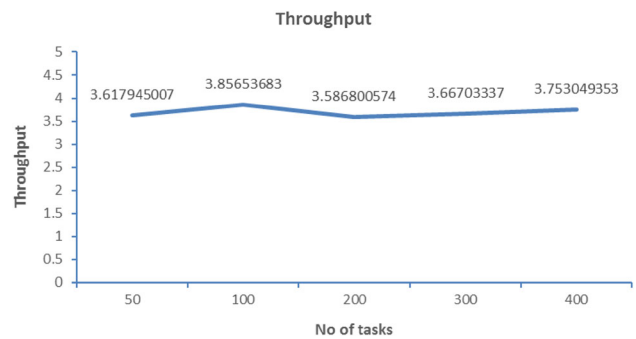


Fig. 9 Performance of proposed approach on the scale of throughput with the change of tasks

6 Conclusion

The algorithm proposed in this paper, is perfect blend of ant colony optimization and gravitational search algorithm to solve the issues of load balancing. It has been found that proposed hybrid approach is fruitful than basic approach. This has not only avoided the premature convergence issue of ant colony optimization but also achieve the distributed nature which was not available in gravitational search

algorithm. Using CloudSim the simulation results so obtained proves that this proposed algorithm is an improvement in terms of decreasing the task completion time and also in assigning approximately equal loads to each virtual machine which creates a balance in the load of virtual machines, thereby reducing the consumption of resources. Also proposed hybrid algorithm can check the capability of the machines and can allocate them the tasks in efficient manner, thereby increasing their efficiency.

As future work, the proposed algorithm has to be compare with other models so as to verify the consistency of the proposed algorithm. Secondly, some parameters such as cost of the proposed algorithm are yet to be realized.

References

- Jain A, Kumar R (2014) A taxonomy of cloud computing. *Int J Sci Res Publ* 4(7):1–5
- Sasikala P (2011) Cloud computing: present status and future implications. *Int J Cloud Comput* 1(1):23–36
- Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2008) Cloud computing and emerging IT platforms: vision, type, and reality for delivering computing as the 5th utility. *Future Gener Comput Syst* 25(6):599–616
- Dam S, Mandal G, Dasgupta K, Dutta P (2014) An ant colony based load balancing strategy in cloud computing. In: *Advanced computing, networking and informatics—volume 2, Part of the smart innovation, systems and technologies book series (SIST, vol 28)*, pp 403–413
- Raut H, Wasnik K (2015) Load balancing in cloud computing using ant colony optimization. *Int J Innov Res Comput Commun Eng* 3(12):12832–12837
- Li K, Xu G, Zhao G, Dong Y, Wang D (2011) Cloud task scheduling based on load balancing ant colony optimization. In: *Sixth Annual China Grid Conference, IEEE*, pp 3–9
- Tawfeek M, El-Sisi A, Keshk A, Torkey F (2015) Cloud task scheduling based on ant colony optimization. *Int Arab J Inf Technol* 12(2):129–137
- Mishra R, Jaiswal A (2012) Ant colony optimization: a solution of load balancing in cloud. *Int J Web Semant Technol (IJWest)* 3(2):33–50
- Jain A, Kumar R (2016) Scalable and trustworthy load balancing technique for cloud environment. *Int J Eng Technol (IJET)* 8(2):1245–1251
- Kalra M, Singh S (2015) A review of metaheuristic scheduling techniques in cloud computing. *Egypt Inf J* 16:275–295
- Sharma A, Tyagi S (2016) Differential evolution-GSA Based optimal task scheduling in cloud computing. *Int J Eng Sci Res Technol* 5(7):1447–1451
- Tavakkolai H, Hosseinabadi AAR, Yadollahi M, Mohammadpour T (2015) Using gravitational search algorithm for in advance reservation of resources in solving the scheduling problem of works in workflow workshop environment. *Indian J Sci Technol* 8(11):1–16
- Rawal P, Rani S (2016) CPU task scheduling using gravitational search algorithm. *Int J Eng Sci Comput* 6(6):6768–6870
- Jain A, Kumar R (2017) Hybrid load balancing approach for cloud environment. *Int J Commun Netw Distrib Syst* 18(3–4):264–286
- Buyya R, Ranjan R, Calheiros RN (2009) Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. In: *High performance computing and simulation, HPCS'09. International Conference*, pp 1–11