International Journal of Grid and Distributed Computing
Vol. 11, No. 8 (2018), pp.13-22
http://dx.doi.org/10.14257/ijgdc.2018.11.8.02

Traveling Salesman Problem with Ant Colony Optimization Algorithm for Cloud Computing Environment

Taskeen Zaidi[1*] and Prashanshi Gupta[2]

[1]Department of Computer Science and Engineering, Shri Ramswaroop Memorial University Barabanki, Lucknow, India*
[2]Department of Computer science and Engineering, Shri Ramswaroop Memorial University Barabanki, Lucknow, India
[1]Taskeenzaidi867@gmail.com, [2]Prashanshigupta0@gmail.com

Abstract

Cloud computing or remote computing is one of the patterns in the IT field that offers resources in heterogeneous manner. All the resources are put on cloud and we may get access it from any location as pay per use. In dynamic cloud computing environment, efficient job scheduling with allocation of resources on demand is one of the major challenges. It is also a challenge to provide the different resources in the form of web services to the cloud users. In this work we have used an Ant colony optimization approach advantages for solving NP hard problems like Travelling salesman problem. The paper aims to reduce the job execution and completion time based on pheromone value. The results depicted that an Ant colony optimization is a good approach for scheduling of jobs in a cloud computing environment.

Keywords: *ACO, TSP, Cloud computing, IACO, NGTSP*

1. Introduction

As Real ants are capable of finding the shortest path between the food sources to the nest. Also, they are capable of adapting any changes in the environment, for example, finding a new shortest path if the old one is no longer feasible due to a new obstacle. Ant Colony Optimization targets discrete optimization problems and can be extended to continuous optimization problems which is useful to find approximate solutions. Now a day, a number of algorithms inspired by the foraging behavior of ant colonies have been applied to solve difficult discrete optimization problems. ACO is used to provide an effective solution to many combinatorial problems like Travelling Salesman Problem and this work is organized as describing ACO, illustrate TSP and an algorithm is proposed for TSP and best route and path length is calculated for jobs.

2. Related Work

Medhat *et al.*, [1] uses an Ant Colony Optimization (ACO) algorithm for cloud task scheduling policy and compared with different scheduling algorithms like First Come First Served (FCFS) and Round-Robin (RR), has been presented. The main goal of these algorithms is minimizing the makespan of a given tasks set. Ranjan Kumar [2] has proposed a model based on Ant Colony Optimization for proper utilization and to reduce congestion. Beena *et al.*, [3] has optimized the resources in an hasty manner. Ant Colony Optimization is chosen among one of the different alternative paths to determine the processing order of each resource. The search space is reduced to provide a better solution. Travelling Salesman Problem (TSP) is

Received (February 19, 2018), Review Result (May 14, 2018), Accepted (May 23, 2018)

ISSN: 2005-4262 IJGDC

the application is also used here to find the shortest path to the destination. This reduces the delay in allocating resources to the user by providing a global search technique. The Ant Colony Optimization (ACO) algorithm reduces the amount of energy used by each resource and hence the performance of the system is increased. Effective time utilization is also done using this algorithm. Brintha *et al.*, [4] proposed an optimization technique to optimize the resources through distributed computation. Ant Colony Optimization (ACO) is used to choose different alternative rules to determine the processing order of each resource and one among the different rule is chosen. This reduces the delay in allocating resources to the user by providing an adaptive and global search technique. This approach reduces the total completion time of jobs and also takes in to account the migration time of the process. Hlaing *et al.*, [5] proposed system based on basic Ant Colony Optimization algorithm with well distribution strategy and information entropy which is conducted on the configuration strategy for updating the heuristic parameter in Ant Colony Optimization to improve the performance in solving TSP. Then, Ant Colony Optimization algorithm (ACO) for Traveling Salesman Problem (TSP) has been improved by incorporating local optimization heuristic. Gondhi *et al.*, [6] has proposed modified approach based on ant colony optimization algorithm for better optimization of allocation process of virtual machines to their respective physical counterparts. This greatly increases the efficient usage of current physical resources. Bhatt *et al.*, [7] suggested a new solution for resource scheduling in computational cloud environment. The resource-scheduling algorithm was proposed that utilizes the ACO (Ant Colony Optimization) algorithm for scheduling the resources and manage the resources. The resource-scheduling algorithm optimizes the performance of computational cloud and provides the efficient resources scheduling strategy in execution. Hingrajiya *et al.*, [8] proposed an approach for solving traveling salesman problem based on improved ant colony algorithm. The main objective of this paper is to avoid the stagnation behavior and premature convergence by using distribution strategy of initial ants and dynamic heuristic parameter updating based on entropy. Then emergence of local search solution is provided. The results after experiment and performance comparison showed that the proposed system reaches the better search performance result over Ant Colony Optimization algorithms. The proposed system is more efficient in terms of convergence speed and in finding the better solutions. Mou [9] has proposed a New Generalized Traveling Salesman Problem (NGTSP), and the current Generalized Traveling Salesman Problem (GTSP) is only a special case of the NGTSP. To solve effectively the New Generalized Traveling Salesman Problem (NGTSP), the ant colony system method extended from Traveling Salesman Problem (TSP) to New Generalized Traveling Salesman Problem (NGTSP). For a better result, a local searching technique is introduced. So that this method can speed up the convergence, and a novel parameter adaptive technique is also introduced into this method to avoid locking into local minima.

Qingbin [10] proposed Improved Ant Colony Optimization (IACO) algorithm that improves pheromone factors and inspired factors innovatively based on the existent algorithms. The results indicate that Improved Ant Colony Optimization (IACO) is superior to the conventional ACO and the latest Improved Artificial Bee Colony Algorithm IABC in task executing efficiency. Yang *et al.*, [11] has proposed an improved ant colony algorithm to solve Travelling Salesman Problem TSP. By changing the amount of information and searching for the optimal parameters, it can speed up the convergence velocity. Simulation results verify the effectiveness of the proposed algorithm. Pandey *et al.*, [12] formulates a model for the multi-objective task assignment and describes a particle swarm optimization algorithm in cloud computing environment. The algorithm not only optimizes the time, but also optimizes the cost. The experimental result manifest that the proposed method is more effective and efficient in time and cost. Sun *et al.*, [13] explained that tasks scheduling problem in cloud computing is NP-hard, and it is difficult to attain an optimal solution, so used an intelligent optimization algorithms to approximate the optimal solution, such as ant colony optimization algorithm. To solve the task scheduling problem in cloud computing. In this paper, the period ACO (Ant Colony Optimization) based scheduling algorithm (PACO) has been proposed. PACO uses ant colony optimization algorithm in cloud computing, with the first proposed scheduling strategy and improved pheromone intensity update strategy. Wei *et al.*, [14] extended the task scheduling problem to the mobile cloud computing environments by extending the Cloudlet architecture. Authors have taken each tasks profit into consideration in order to maximize the profit of the system, which is an import target of the task scheduling algorithm in the commercial mobile cloud environment. They designed their proposal based on the hybrid Ant Colony Optimization ACO algorithm, which has been validated by experiments. Dorigo and Gambardella [15] described an artificial ant colony that was capable of solving Travelling Salesman Problem TSP. They described about the real and artificial ants. Nishant *et al.*, [16] discussed about the load distribution of workloads among nodes of a cloud by the use of Ant Colony Optimization (ACO) algorithm. This is a modified approach of ant colony optimization the main of this approach is load balancing of nodes that has been applied from the perspective of cloud or grid network systems. Nonsiri and Supratid [17] discussed about the Ant Colony Optimization ACO that allows the fastest optimal solutions. It is useful in industrial environments where computational resources and time are limited. Marco Dorigo *et al.*, [18] described an Ant Colony Optimization based algorithm known as AntNet for resolving the routing problem in telecommunication. Gao *et al.*, [19] proposed an optimized algorithm for Virtual Machine (VM) placement in cloud computing scheduling based on multi-objective ant colony system algorithm in cloud computing. M. Dorigo *et al.*, [20] proposed an ant colony optimization (ACO) algorithm, a novel bionic evolutionary algorithm, it is a kind of probability based technology used in the graph to find the optimal path. Moharana *et al.*, [21] proposed a load balancing technique that is Load balancers are used for assigning load to different virtual machines in such a way that none of the nodes gets loaded heavily or lightly. The load balancing is need to be done properly because failure in any one of the node can lead to unavailability of data. James[22] proposed a new VM load balancing algorithm and implemented for an IaaS framework in Simulated cloud computing environment; *i.e.*, 'Weighted Active Monitoring Load Balancing Algorithm' using CloudSim tools, to achieve better performance parameters such as response time and Data processing time. In the current work ACO is used with TSP for finding the shortest round trip time by visiting each of the node and the path length and optimal distance is computed.

3. Background

3.1. Cloud Computing

Cloud computing is an emerging platform within the field of information technology (IT) that is widespread as it provides computing resources accessibility such as applications, storage, services, video games, movies and music on demand in such a way that the Cloud users need not have any idea how or from where they are receiving these resources. The only thing they needed is a internet connectivity to the Cloud service provider. Cloud computing provides accessibility in efficient, easy manner, and assigns on-demand network to a shared pool of configurable computing resources (such as, networks, servers, storage, applications, and services, hardware resources, CPU cycles *etc.*,) that can be managed with optimum efforts and rapidly provisioned.

The end-users that are connected in the cloud are not aware about internal details of specific technology while hosting their application, as the Cloud Service Provider (CSP) assigns the service. Users may access services at a rate that is set by their particular demands. This on-demand service can be provided any times from any were. CSP manages all the necessary complex operations on behalf of the user. It may provide the complete system that allocates the required resources for execution of user applications and management of the entire system flow for fulfilling the demand of cloud users.

For making the cloud computing feasible and easily accessible to end users there are various models and services that are working behind the scene. The working models for cloud computing are as follows:

- Deployment Models-: Deployed to provide access to the cloud. Cloud may be accessed as: Public, Private, Hybrid, and Community.
- Service Models-: based on service provisioning. These are classified into three basic service models as –
    - Infrastructure-as–a-Service (IaaS)
    - Platform-as-a-Service (PaaS)
    - Software-as-a-Service (SaaS)

Cloud computing is a virtual collection of resources which are provided to users on demand. Cloud service provider assigns the users virtually unlimited pay-per-use computing resources without worrying about the management of the underlying infrastructure. The objective of cloud computing service providers is to distribute and use the resources in an efficient manner to attain maximum profit. Cloud has another layer known as virtualization layer. Virtualization layer provides creation, execution, management and hosting environment for application services. The VMs used in the above virtual environment are isolated but they may share computing resources processing cores, system bus etc. So, the amount of hardware resources available to each VM is limited by the total processing powers provided such CPU cycles, the memory and system bandwidth, capacity available within the host.

3.2. Ant Colony Optimization

The ant colony optimization algorithm (ACO), was formulated by Marco Dorigo in 1992.It is a Meta heuristic algorithms for optimizing complex problems and it is a probabilistic technique that searches for optimal path based on ants behavior from source to food. ACO algorithm is one of the most successful Meta heuristic approaches and widely recognized algorithmic based on the ant behavior.

- Ants move from nest to food resource.
- They discovered the shortest path based on pheromone deposition.
- The paths are selected randomly.
- If pheromone deposition is more on a path the probability of that path selection will be increased.

When an ant moves from point source to destination, ant makes pheromone trails, to mark these paths. This helps the other ants to find the way passed by their team members as they check pheromone trail and choose, in probability, the paths having greater concentration of pheromone. This algorithm is based on an adaptive approach of adjusting the pheromone on routes at each node. Choice of this node is guided by a probability based selection approach randomly.

3.3. Travelling Salesman Problem (TSP)

TSP is an important operation research problem. It consists of set of cities to be visited by a salesman and the salesman start from the source cities and visited other cities to reach destination only once and come back to the initial point. In this problem it is important to minimize the path length to minimize the cost from source to destination. In an ant colony optimization, the problem is checked on simulating a number of artificial ants moving on a graph that permutated itself by finding a path: in ACO each vertex described to be a city and each edge associates a connection between two cities. A variable pheromone is associated with each edge and can be read and altered by ants. The traveling salesman problem plays a crucial role in ant colony optimization because it was the first one to be attacked by ACO.
The TSP was chosen because with ACO as:

- It is relatively easy to put the ant colony values to it.
- It is NP hard and there is no solution for this problem based on polynomial time.
- It is one of the most studied problems in combinatorial optimization to solve complex research problems.
- It is very easy to describe, and the algorithm behavior is not obscured by too many technical details.

4. Proposed Work

The resources are allocated through ACO as follows.

1. Firstly input data about number of jobs and resources are collected.

2. The ants used for the jobs and the pheromone variable used are initialized. The pheromone variable is used two times first time it is used to find suitable path and then again it is used to find resource.

3. Ants allocate resources randomly which is shown as pheromone and its heuristic. If there is any conflict in process scheduling then probabilistic rule is used.

4. The current best and the global best position are selected separately.

5. After certain number of iterations it is checked whether best solution is achieved. If global best position gives poor solution then the solution is reconstructed to get an optimal solution.

ACO may be applied to the TSP in a significant way, In all ACO algorithms available in literature for the TSP, Tours are projected by the following simple constructive procedure to each ant: (1) choose, according to some criterion, a start city at which the ant should be positioned; (2) pheromone variable and heuristic values to probabilistically construct a tour by iteratively adding cities that the ant has not visited yet, until all cities have been visited till destination; and (3) then go back to the start city. After all ants have finished their tours, they may deposit pheromone on the tours they have visited. In some cases, before adding pheromone variable, the tours constructed by an ant arriving in city i chooses the next city to move to as a function of the pheromone values and of the heuristic values on the arcs connecting city i to the cities j the ant has not visited yet. This high level description applies to most of the published ACO algorithms for the TSP, one not able exception being Ant Colony System in which pheromone trail evaporates with tour construction. This algorithm's scheme after initializing the ant colony parameters and the pheromone trails, iterate through a main loop, in which first all of the ants' tours are projected, and then an optional phase takes place in which the ants' tours are improved by the application of some local search algorithm, and finally the pheromone trails are updated globally. This last step involves pheromone trail evaporation and the update of the pheromone trails by the ants to reflect their search phase.

Ant colony system was first introduced and applied to TSP by dorigo *et al.* At the initial stage each ant is placed on some randomly chosen city. An ant k currently placed at city I choose to directed to city j by applying the following probabilistic transition rule; [12,13], as shown as equation (1)

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} \quad \ldots\ldots(1)$$

where nij is the heuristic visibility of edge (i, j), generally it is a value of 1/dij, where dij is the distance between city i to city j. Jk (i) is a set of cities which remain to be visited when the ant is at city i. α and β are are two positive parameters that control the relative weights of the pheromone trail and of the heuristic visibility. The trade-off between edge length and pheromone intensity appears to be necessary. After each ant completes its tour, the pheromone amount on each path will be adjusted according to equation and *(1-p)* is the pheromone decay parameter ($0<p<1$) where it represents the trail evaporation when the ant chooses a city and decides to move.

All the ants perform the local pheromone update after each step. Each ant applies it only to the chosen city equation (2):

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho.\tau \quad \ldots\ldots(2)$$

After all the ants have travelled through all the cities, and global update only the amount of the pheromone on the optimal path according to equation (3):

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \quad \ldots\ldots(3)$$

5. Algorithm

STEP 1: START

STEP 2: Dis_Covered initially zero

STEP 3: Do
(Loading balancing using ACO)

STEP 4: Initialize ACO parameter

STEP 5: Inputs jobs using probability distribution

STEP 6: At first iteration jobs (i) moves to different node

STEP 7: Repeat

STEP 8: Compute distance

STEP 9: Select another node to be visited according to solution

STEP 10: Local update the path (according to rule 2)

STEP 11: Job (i) completed the execution
End for

STEP 12: Global update the path (rule 3)
STEP 13: Compute path traverse by jobs

STEP 14: End condition

STEP 13: Analyze and find the optimal solution.

6. Experimental Results

In this work we have taken the values as:
In the given
Taumax =2 pheromone deposit
Initial city=0
No. of cities=8
No. of ants=4
Alpha=0.5
Beta=0.8
Rho=0.2
Q=80(double)
Number of ants=4

The steps are as follows:
1. Initialize distance
2. Initialize ants to random trail
3. Best initial trail length
4. Initializing pheromone on trails
5. Pheromone update
7 .best trial found
8. Computed path length of best trail

The ants (cloudlets) traverse the city (VM) in the five iterations:
1. In the first iteration each ants are assigned to different city.
2. Node 0 is the source so it has different option and chooses path probability depending on random number
3. In this case the amount of pheromone have value 0.5 and the pheromone decay=0.2
4. In iteration 2, node 0 has different option and path of each ant and path length is calculated
5. After each iteration the trial of pheromone evaporates.
6. After traversing each city the ants revert to initial city.
7. Best Tour sequence and length is saved
8. Old ants (jobs) die and new ants (jobs) join.
9. The best route and ideal route is 0 7 6 2 4 5 1 3 as shown in Figure 1.
10. The path length is 127.509 as shown in Figure 2.

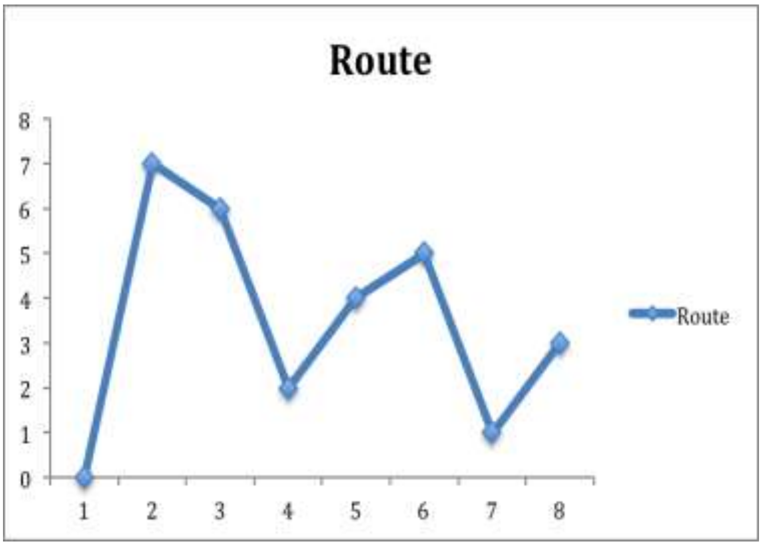

Figure 1. Ideal Route

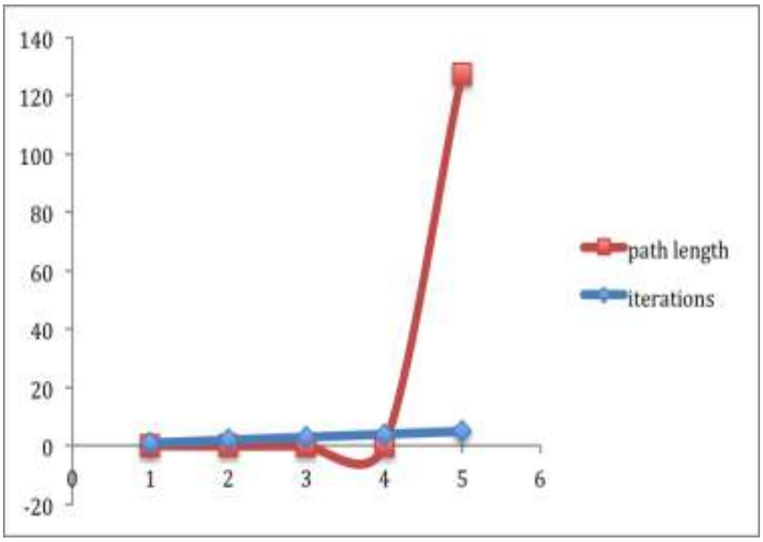

Figure 2. Path Length

7. Conclusion

The current work is based on the effective resources allocation on demand in cloud environment. The energy consumption is one of the prime objectives of this work as due to reduction in energy utilization the performance is increased. The result shows the reduction in completion time of jobs with optimal route. In the future work we will increased the parameters by increasing number of cities and iterations and execution time will be evaluated to justify the ACO as a solution to solve TSP.

References

[1] M. Tawfeek, A. El-Sisi, A. Keshk and F. Torrey, "Cloud Task Scheduling Based on Ant Colony Optimization", The International Arab Journal of Information Technology, vol. 12, no. 2, pp. 129-137, **(2015)** March.

[2] R. Kumar, G. Sahoo and K. Mukherjee, "Performance Analysis of Cloud Computing using Ant Colony Optimization Approach", International Journal of Innovative Research in Science, Engineering and Technology, vol. 2, no. 6, **(2013)** June.

[3] B. M. Beena and H. D. Harshitha, “Ant Colony Optimization for Efficient Resource Allocation in Cloud Computing”, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169, vol. 5, no. 6.
[4] N. C. Brintha, J. T. Winowlin Jappes and S. Benedict, “A Modified Ant Colony Based Optimization for Managing Cloud Resources in Manufacturing Sector”, 2nd International Conference on Green High Performance Computing (ICGHPC), **(2016)**.
[5] Z. Chi Su Su Hlaing and M. Aye Khine, Member, IACSIT, “Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm”, International Journal of Information and Education Technology, vol. 1, no. 5, **(2011)** December.
[6] N. Kumar Gondhi and A. Sharma, “Local Search based Ant Colony Optimization for Scheduling in Cloud Computing”, 2015 Second International Conference on Advances in Computing and Communication Engineering.
[7] P. Mod and M. Bhatt, “ACO Based Dynamic Resource Scheduling for Improving Cloud Performance”, International Journal of Science, Engineering and Technology Research (IJSETR), vol. 3, no. 11, **(2014)** November.
[8] K. H. Hingrajiya, R. Kumar Gupta and G. Singh Chandel, “An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem”, 2011 International Conference on Information Communication and Management IPCSIT vol. 16 (2011) © (2011) IACSIT Press, Singapore.
[9] L. Mou, “An Efficient Ant Colony System for Solving the New Generalized Traveling Salesman Problem”, Proceedings of IEEE CCIS, **(2011)**.
[10] Li Pinghua and Nie Qingbin, “An Improved Ant Colony Optimization Algorithm for Improving Cloud Resource Utilization”, 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.
[11] X. Yang and J.-s. Wang, “Application of Improved Ant Colony Optimization Algorithm on Traveling Salesman Problem”, 978-1-4673-9714-8/16/$31.00_c IEEE, **(2016)**.
[12] S. Pandey, L. Wu, S. Mayura Guru and R. Buyya, “A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments”, 24th IEEE international conference on advanced information networking and applications, **(2010)**, pp. 400-407.
[13] W. Sun, “PACO: A Period ACO_based Scheduling Algorithm in Cloud Computing”, IEEE International Conference on Cloud Computing and Big Data (CloudCom-Asia), **(2014)** July.
[14] X. Wei, J. Fan, Z. Lu and K. Ding, “Application scheduling in mobile cloud computing with load balancing”, Journal of Applied Mathematics, vol. 2013, **(2013)**.
[15] S. Nonsiri and S. Supratid “Modifying Ant Colony Optimization”, IEEE Conference on Soft Computing in Industrial Application, Muroran, Japan, **(2008)**, pp. 95-100.
[16] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. Pratap Singh, Nitin and R. Rastogi, “Load Balancing of Nodes in Cloud Computing Using Ant Colony Optimization”, IEEE, 14th International Conference on Modelling and Simulation, **(2012)**, pp. 03-08.
[17] M. Dorigo and L. Maria Gambardella, “Ant colonies for the traveling salesman problem”, TR/IRIDIA, Université Libre de Bruxelles, Belgium, vol. 3, **(1996)**.
[18] M. Dorigo, M. Birattari and T. Stutzle, “Ant Colony Optimization- Artificial Ants as a Computational Intelligence Technique”, Universeite Libre de Bruxelles, Belgium.
[19] Y. Gao, H. Guan, Z. Qi, Y. Hou and L. Liu, “A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing”, Journal of Computer and System Sciences, vol. 79, no. 8, **(2013)**, pp. 1230-1242.
[20] M. Dorigo and G. D. Caro, “Ant colony optimization: a new meta-heuristic”, Proceedings of the Congress on Evolutionary Computation, vol. 2, no. 4, **(1999)**, pp. 1470-1477.
[21] S. S. Moharana, R. D. Ramesh and D. Powar, “Analysis of Load Balancers in Cloud Computing”, International Journal of Computer Science & Engineering (IJCSE), **(2013)** May, pp. 101-108.
[22] J. James, “Efficient VM Load Balancing Algorithm for a Cloud Computing Environment”, International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 09, **(2012)**, pp. 1658-1663.