Research Article

An Effective Ant Colony Optimization Methodology For Virtual Machine Placement In Cloud Data Centre

A. Arivuselvi[1], Dr. T. Amudha[2], P. Deepan Babu[3]

M.Phil Scholar[1], Associate Professor[2], Assistant Professor[3]
Department of Computer Applications[1, 2], Department of IT & CT[3]
Bharathiar University[1, 2], VLB Janakiammal College of Arts and Science[3]
Coimbatore, Tamil Nadu, India
arivuselvi255@gmail.com[1], amudhaswamynathan@buc.edu.in[2], pdeepan_13@yahoo.co.in[3]

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: A Cloud environment is a type of service-oriented computer application, and its services comprise SaaS, IaaS, PaaS, and XaaS. The cloud user chooses essential services from dust, and those service models fulfill customers' incoming tasks with service contracts. Each task is executed in Physical Machine (PM) deployed in the cloud through Virtual Machine (VM), so VM plays a vital role in the cloud environment. A cloud would be a pay as you go platform, and hence many VMs within the physical servers can be created and destroyed. The effective allocation of virtual machines to physical machines permits the economic sharing of physical devices to obtainable data centers. These allocation strategies facilitate measuring and enhancing the performance of the cloud. The focus of this work is to put forward a VM migration technique in the cloud environment, which improves the CPU and memory utilization with reduced traffic congestion in the cloud data centers using bioinspired algorithms. The proposed approach improves the memory management, reduces the traffic, and maximizes CPU utilization the current state of the VMs and PMs is not interrupted.

Keywords: Cloud Data Centre, Physical Machine, Virtual Machine Placement, Bio-inspired algorithm, Ant Colony Optimization

1. a. Introduction

The term 'Cloud' is taken from the real cloud, the way a natural cloud is formed in different sizes and shapes. It frequently changes its shape. Similarly, cloud computing also connects the various computing devices for a specific job and disappears when work completes. Almost all the industries, whether IT or non-IT, have moved their political process into the cloud, according to the nature of their services, such as using infrastructure, platform, software, or anything. Cloud is also an emerging multidisciplinary field of research to analyze the massive amounts of data and user requests to obtain actionable information. Cloud computing has become the appropriate platform for Big Data processing because of its on-demand conductivity, extremely low latency, or massively parallel computing architecture [1]. Server virtualization, which virtualizes physical cloud resources, underpins the cloud computing system. A virtual cloud environment improves both the system's performance and scalability. A software device is classified in a cloud setting with virtual machines (VMs). These VMs are mapped to specific customer requests, including tasks also to be performed feedback.

As resources are now over-subscribed, strategic planning is much more complicated, and cloud users are not cooperative. To handle these situations, a cloud service provider (CSP) will have to follow a proper scheduling mechanism for services delivered. Technically, a standard of service contract (SLA) is considered an agreement between the CSP and the customer. SLA is also an integral part of the Quality of Service (QoS). All the rest of the paper, we use PM and host, synonymously. There are a great many developments in designing energy-efficient servers or networking tools. To energy efficiency, approximately 20 percent could be achieved through data centers, which additionally save up to 30% on cooling energy requirements [2]. Standard cloud deployment consumes substantial amounts of power, or even in do other, increases carbon dioxide (CO2) levels.

b. Bio-Inspired Computing - Ant Colony Optimization Algorithm

Inspired by the biological computing or, in short, Bio-inspired computing would be a growing computer and information technology as well as a study that loosely knits together subfields related to the topics of connectives, social behavior & plants, and animals emerging. It relies on a massive combination of fields like biology, computer science, and mathematics. In short, it is the use of lifestyle of biological species, their survival and reproduction mechanisms in computer algorithms to improve the performance metrics such as processor utilization, memory utilization, balancing the load of any device, network routing, and so on. Biologically inspired computing is a significant subset with natural computing with various methods like Ant colony optimization, bee colony optimization, swarm intelligence, evolutionary computing, and so on.

Ant colony optimization (ACO) is a form for bio-inspired computing, but its activity is focused on ant colonies' natural behavior or their working ants. And if ants forage, they tend to find the "logical" and "efficient" route among their nest as well as the food source; in several other words, an optimal path seems to be decided. This measured behavior is based upon ACO (Bianchi et al., 2008). Imagine two ants walking from the nest to a source of food utilizing two distinct routes. But as ants walk, pheromones are released that naturally decay over time. An ant that (randomly) chosen the shortest route will start a journey's return leg faster than another ant, thereby strengthening the trace of the pheromone more shortly. Many ants naturally follow the direction of a more potent pheromone, strengthening it and contributing to all of it. According to Bianchi et al. (2008), the ACO algorithm includes three main parts that constitute an algorithm's central optimization loop:

1. Creating solutions for the Ants. That is the methodology whereby "ants" incrementally as well as stochastically establish paths, i.e., results in a broader context of optimization.
2. Pheromone evaporates. That is the method wherein the pheromone is reduced using "local" knowledge for some solutions; Therefore, this stage is sometimes referred to as a local update. This step is crucial for ensuring that perhaps the ACO algorithm does not prematurely converge to a single solution.
3. Behavior by the daemon. Its phase applies to decisions based on global knowledge about the issue for optimization, also associated with comprehensive updates.

The three stages outlined above have been replicated until the optimization problem converges or terminates otherwise through a present termination condition. Depending on the above, it is indeed clear which substantial research on development & analysis has been needed until ACO can be applied to a troubling problem besides the optimization of nonlinear topology. The Fig.1 illustrate the Ant colony optimization algorithm

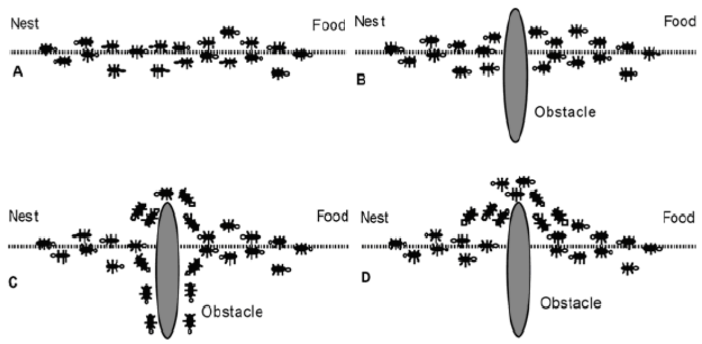

Fig.1. Ant colony optimization algorithm

2. Literature Survey

Research in [3] focused on various proposed systems as well as a complex Virtualised data center problem of energy-aware optimization of virtual machine placement. Particle Swarm Optimization (PSO) has an excellent opportunity to explore a better alternative solution to reduce energy usage. An author of [4] examined a data center's virtual machines that provide multi-dimensional resources and energy consumption, as well as proposed the Jaya algorithm [5] to optimize the data center's positioning and minimize energy consumption.

A VM placement algorithm is proposed in [6] based on the meta-heuristics of the Particle Swarm Optimization (PSO). In comparison to recorded VM placements, which seek to reduce either energy usage or a waste of resources, the proposed PSO algorithm maximizes the efficiency of packaging while minimizing energy consumption. [7] Proposes an alternative multi-objective optimization (MOP) method incorporating the salp swarm as well as sine-cosine algorithms (MOSSASCA) to evaluate the practical solution for virtual machine placement (VMP). A suggested MOSSASCA wants to expand a mean time before the first host shutdown, decrease power usage, and eliminate the level of service violation contracts.

An author of [8] used biogeography-based optimization (BBO) as just a method of optimization, considering server loads, inter-VMs, power consumption, resource wastage, and network storage traffic, to ascertain a solution to the issue. In addition to a study [9], a new genetic algorithm (GA) is used for solving the VMP issue in such a communication network and PMs in a data center. Gao et al. [10] proposed a multi-objective ant colony optimization (ACO) algorithm for resolving VMP problems; these have been used to find efficient, non-deterministic solutions to reduce power output or waste consumption.

Shabeera et al. [11] Presented an ACO dependent VMP solution for selecting confined PMs to locate data and VMs. A set of VMs is located on PMs in such a cloud data center, as well as process data stored in them. As just a host of them for processing, the PMs need to provide the VMs adequate resources. This algorithm chooses PMs in proximity, and the jobs are done throughout the VMs allocated by such a scheme outperform specific allocation schemes. A Clouds tool is being used to simulate such a scenario, as well as the numerical simulations, which have been displayed as follows in 3 cloud data centers: Unicloud, Multicloud, and Pre.

An approach in [12] presents an ACO-based VMP result, using artificial ants based on global search information. Through assigning VMs to a minimum number of active PMs, its algorithm received an optimal solution. The whole approach employed so several VMP problems, homogeneous and heterogeneous, in cloud environments of varying VM sizes. The main goal of this strategy is to reduce power usage and prevent duplication of energy in cloud DC. Throughout the references [13], an author suggested an ACO-based VMP approach for evaluating the appropriate PMs for the migration of VMs. Multi-objective optimization is one of the most significant aspects of cloud center VMP issues. Optimum positioning of a VMs scheme is calculated based on function metrics, multi-objective optimization, and the ACO. We used OpenStack / Nebula tools to simulate such a process, and the obtained results were compared with many other methods, like meta-generalized gradient estimation.

3. Problem Formulation

A VMP is also an NP-hard issue, as well as being a sort of problem of optimization. This same optimization algorithm and approaches are the right solution to handle and solve this type of problem. Virtual machine operation, Virtual Machines (VMs) were virtualized environments of predetermined virtual resources like Processor, operating system and middleware-configured memory space & bandwidth, and one or more applications. As with any PM, VM's do workloads. Cloud service providers offer their computing resources for their customers using only a Service Level Agreement (SLA). The services given are typically throughout the type of VMs which place different PMs to perform various tasks. In addition to enabling service providers to charge their customers based on their use in a pay-as-you-go scheme, a virtualization capability also provides customers with the ability to scale up or down resource usage, even though their needs vary. These advantages were due to part to the fact that virtualization technology allows multiple virtual servers that run on the same PM, leading to better memory & CPU utilization.

CPU utilization

PM CPU use is indeed a measure for the determination with under loaded PMs. PM is regarded as under full while using a minimum amount of resources.

Memory utilization

VM selection is a process of choosing the best one or more VMs from overloaded PMs to move them for limited memory use. Thus, summarizing the formulation:

$$Minimize \sum \quad _{Yi} \tag{1}$$

Subject to:

$$\sum_i r_{ik}.x_{ij} \leq 1 \quad \forall j \tag{2}$$

$$\sum_i x_{ij} = 1 \quad \forall i \tag{3}$$

$$y_i \geq x_{ij} \ \forall i,j \tag{4}$$

Capacity Constraints

The sum of existing resources of all of the VMs placed upon this should be less than or equitable to an installed output accessible through each dimension of the given PM *j*, where r_{ik} will be the fractional resource requirement of even an individual VM *i* along with the dimension *k*. That inequality has been created per each dimension by each PM.

Placement Guarantee Constraints

This should be placed on all virtual servers. So, by each VM *i* it is expected that precisely one of the x_{ij}'s will be one, and another 0. For each PM j, a differential y_j is associated, as seen below, to formulate the objective function. *Yj* will either be 0 or 1 based on whether to use PM *j* or not. When the constraint is met, y_j tells if PM *j* should be used or not, while x_{ij} values start giving PM mapping to a VM.

4. Proposed Methodology

Ant Colony Optimization Technique for Virtual Machine Placement in Cloud Data Center 's total architecture, as given in Fig. 2; Cloud user-submitted VM queue requests for everyone's virtual machine. First come First serve basis processes a request. This same resource broker throughout the cloud will forward requests to the Resource Finder. A resource finder identifies the appropriate resources for placing a virtual machine that meets hardware, software, and QoS limitations. A resource-finder finds a matched set using the Tanimoto coefficient. The placement manager gets information from either a cloud system stored in such a data server. A placement manager would aim to put a virtual machine to maximum effect on physical computers using ACO. An optimization algorithm besides Ant colony incorporates a Placement Manager. Track the status of the physical devices in the Resource Repository regularly after installing a virtual machine on a physical computer, too. Will any physical machine be overloaded if a monitor finds, migrator has been invoked? A migrator migrates the virtual machine placed inside the overloaded physical machine to some other device.

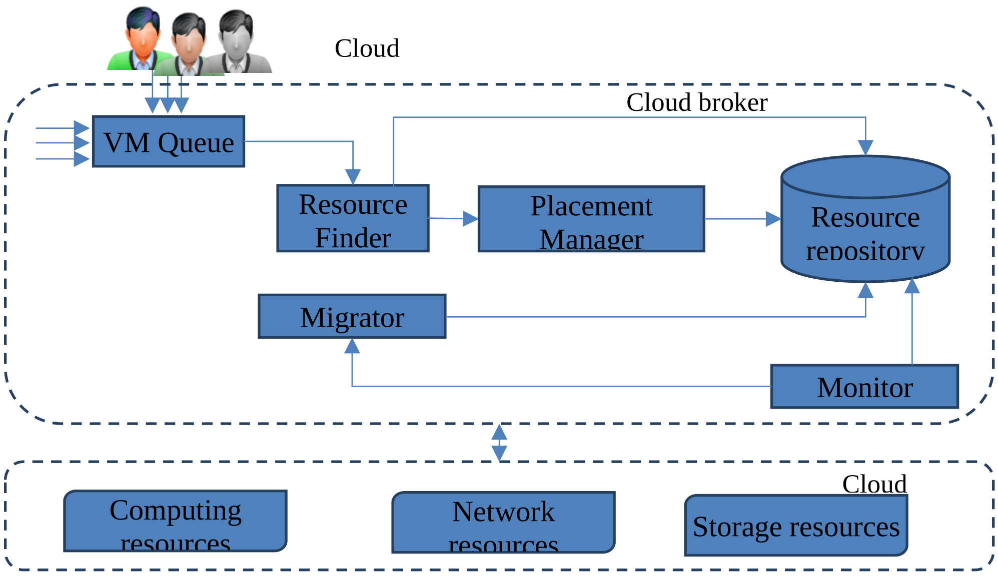

Fig. 2. Proposed System Architecture

5. Working of ACO on Cloud for VM placement

The main idea is to select the best order of memory, CPU allocation, and optimal path discovery through the ant colony algorithm, in which the distribution can be regarded as an ant traversal process. The pheromone is updated according to the performance parameters of the memory, CPU, and optimal path, traffic demands. A physical machine with high pheromone concentration (high performance and low load) is selected to handle the assigned virtual machine. The step by step procedure of the Proposed Algorithm is given in Table.1.

1. Inputs: PMList, VMList
2. Output: Memory Utilization, CPU Utilization, and Optimal path discovery
3. Begin
4. For each ant M antlist
5. MemoryUtilization ←Min
6. AllocatedPM←Null
7. **For each** PM in PMList
8. Do
9. **If** PM has enough resource for VM then
10. Send data to the VM from PM
11. Data ← Datacenter (PM, VM)
12. **If** Data < MaxData then
13. AllocatedPM←PM
14. MaxData←PM
15. End If
16. End If
17. If allocatedPM ≠ null **then**
18. Allocation.add (VM.AllocatedPM)
19. End If
20. For each and K analyst and O Optimalpath
21. CPU_Utilization ←Min
22. AllocatedPM←Null
23. **For each** PM in PMList
24. Do
25. If CPU utilization is min for a resource for VM then
26. VM send data from PM
27. O← Optimal path identification
28. Data ← Datacenter (VM, PM)
29. **If** Data < MaxData then
30. CPU utilization is done
40. End If

```
  End If
End for
End for
End for
For each ant T antlist
      Traffic ←MAX
      AllocatedPM←Congestion
  For each PM in PMList
Do
If Traffic is Maximum for sending data to cloud for VM then
 Ant T find another route for sending data to PM from VM
      PM ← Data (VM, Datacenter)
      If Data send to PM then
      Clear traffic is done
            End If
      End If
End for
End for
End for
```

Optimal Allocation of Virtual Machines Using ACO

Throughout this research, the new approach is being investigated utilizing an ant colony optimization method to find the optimal solution for a traffic intersection. With a different number of VMs as well as tasks, the proposed algorithm has experimented. Once the VMs serve a job, the excess resources will then be stored throughout the individual VM buffers. Also, these tools are being used to meet the demands of a specific mission. VMs start soon executing a task while providing services from the cloud resource pool as well. A resource in ACO suits the VMs that have a higher concentration of pheromones. As most of the time, there will be no perfect resource match due to the increasing level of evaporation of pheromones, and thus ACO takes the same execution time for all instances. An algorithm tries to fit all combinations but instead allocate funds to potential solutions. An ability of ACO to incorporate heuristic information about traffic networks makes it more efficient. By applying the Ant colony optimization algorithm, memory utilization and CPU utilization are improved, and the traffic demand is reduced.

6. Results and discussion

This division provides an implementation solution to show the effectiveness of the proposed Ant Colony Optimization algorithm on memory utilization, CPU utilization, and traffic demand reduction. The dataset for this research work was taken from the benchmark dataset available in the link, http://www.cs.nott.ac.uk/~pszrq/benchmarks.htm. Its connection contains a dataset called "Online datacentre virtual machine placement," which is uploaded through Rong Qu, University of Nottingham's School of Computer Science. The whole link also includes datasets besides real-world applications like Transport Logistic Scheduling, Vehicle Routing, Connected as well as Autonomous Vehicle Cyber Security (CAVs), Personnel Scheduling, Portfolio Optimization, Multicast Network Routing, and Timetabling. The dataset taken for this research work contains the configuration for PM, VM, and traffic demands. It includes 2 types of PM, which are type 1 and type 2 for machine values. The implementation is done optimally, placing the VMs in the available PMs and minimizing the traffic demands among VMs. The results of traffic demand are tabulated in Table 2. From the results, it could be observed that there is a good improvement in CPU utilization, memory utilization, and traffic requirements.

Table 2. Reduction in Traffic Demand through Optimal PM-VM allocation by ACO

Problem Instance	Available		Optimal Allocation By ACO		Traffic Reduction (in %)
	PM	Traffic Demand (in Mbps)	PM	Traffic Demand (in Mbps)	
I1	20	1336.54	7	1060.18	21
I2	20	1184.82	7	1069.74	10
I3	20	1123.08	7	936.48	17
I4	20	1320.48	11	1211.38	8
I5	20	1198.04	12	864.36	28
I6	20	1296.70	11	1270.78	2
I7	20	1758.72	19	1580.56	10
I8	20	1588.74	19	1569.24	2

I9	20	1458.00	19	1434.33	1
I10	40	1715.70	16	1696.02	1
I11	40	1570.60	16	1533.96	2
I12	40	1421.06	16	1270.40	11
I13	40	2669.52	24	2641.74	1
I14	40	1689.72	25	1644.08	3
I15	40	1965.20	24	1562.49	20
I16	40	2068.86	38	2051.86	1
I17	40	2580.24	37	2474.95	4
I18	40	1992.08	37	1925.05	3
I19	60	2455.16	31	2424.24	1
I20	60	1925.12	25	1799.98	7
I21	60	2817.66	23	2725.78	3
I22	60	2776.30	32	2595.66	3
I23	60	2332.20	32	2196.62	6
I24	60	2501.08	32	2485.06	5
I25	60	2797.18	52	2565.27	9
I26	60	3131.64	54	3010.23	4
I27	60	3062.12	54	2858.82	7
I28	80	2891.54	26	2477.88	14
I29	80	3098.64	27	3010.86	3
I30	80	3342.80	26	3151.78	6
I31	80	3376.60	43	3269.24	3
I32	80	3766.48	42	3444.45	9
I33	80	3785.48	42	3591.13	5
I34	80	4080.14	79	4001.20	2
I35	80	3760.50	80	3713.50	7
I36	80	4669.02	80	4538.86	3

Table 2 shows the results obtained by optimal VM-PM allocation by ACO, for instance, I1 to I36 to minimize traffic demands.

Fig.4 shows the optimal memory utilization of 36 instances using ACO. The proposed ACO method has less memory utilization results indicating the optimal VM placement result. It is evident from the figures that memory utilization is particularly useful, with almost 80 % to 99 % of usage in most of the instances considered in this work. Fig.5 shows the optimal CPU utilization of 36 cases using ACO. The proposed ACO method has fewer CPU utilization results indicating the optimal VM placement result. The CPU utilization is also found to be good, with 60 % to 77 % of use in most of the instances. This shows that ACO performs well in VM – PM allocation with optimal CPU utilization and memory utilization. A reason for this is that the algorithm proposed uses less computational approaches to update national and global pheromones.

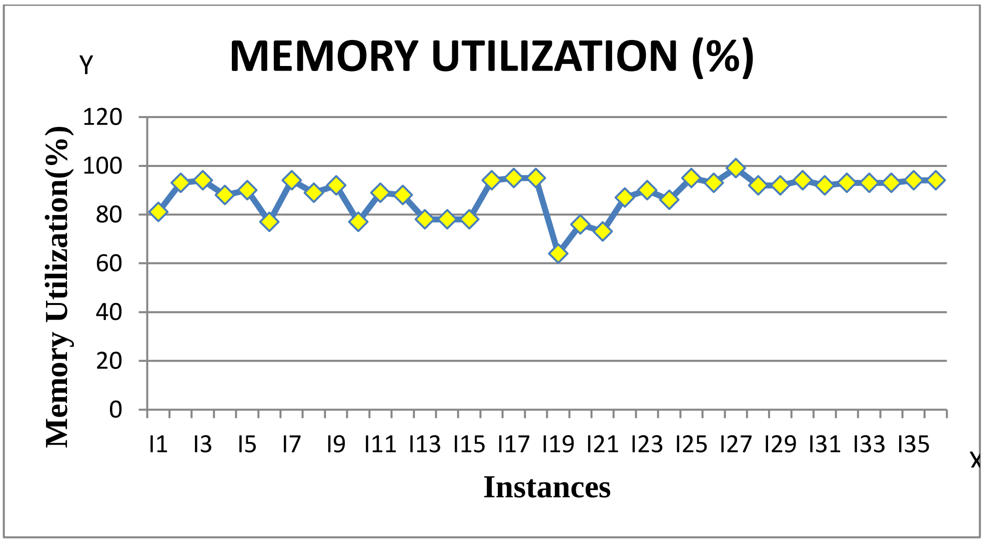

Fig.4. Memory utilization of benchmark instances using ACO

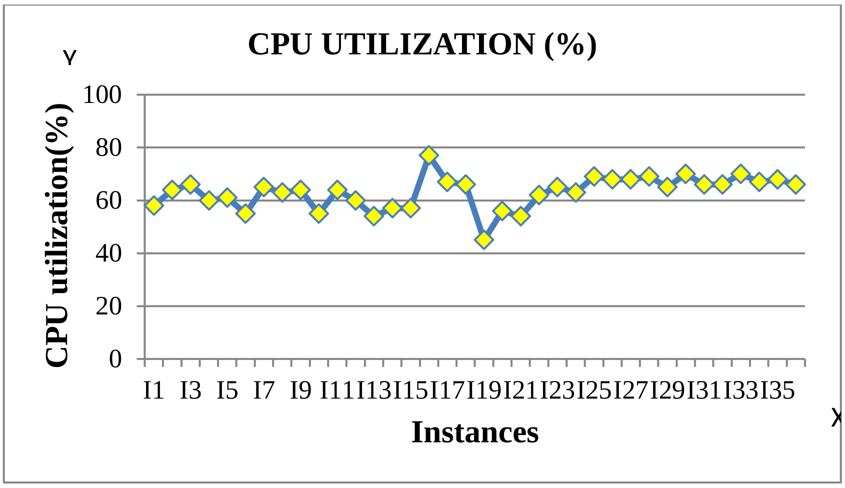

Fig.5.CPU utilization of benchmark instances using ACO

7. Conclusion and Future Work

In this job, the problem of virtual machine positioning was formulated as just an NP-hard problem to obtain effective use of total memory resources and the full use of treatment resources. To solve the problem formulation ant colony optimization algorithm was proposed. In addition to large data centers, a proposed algorithm is built to efficiently deal with possible ample solution space and obtain the collection of non-dominated solutions. A proposed algorithm with several devices ranging from 50 to 300 has also been evaluated. Results show that the proposed algorithm performs excellent for less use of memory and use of CPU. For future ventures, other Metaheuristics techniques will address a VM placement problem and ensure the high performance compared with that of the ACO. The proposed algorithm will also be optimized for handling peak temp between servers, and the carbon emission control would also be considered.

References

1. D. Puthal, B. Sahoo, S. Mishra, S. Swain, Cloud computing features, issues, and challenges: a big picture (2015) 116–123doi:10.1109/CINE.2015.31.
2. Dayarathna, Miyuru, Yonggang Wen, and Rui Fan. "Data center energy consumption modeling: A survey." *IEEE Communications Surveys & Tutorials* 18, no. 1 (2015): 732-794.
3. Wang, Shangguang, Zhipiao Liu, Zibin Zheng, Qibo Sun, and Fangchun Yang. "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers." In *2013 International Conference on Parallel and Distributed Systems*, pp. 102-109. IEEE, 2013.
4. Reddy, M. Amarendhar, and K. Ravindranath. "Virtual Machine Placement Using JAYA Optimization Algorithm." *Applied Artificial Intelligence* 34, no. 1 (2020): 31-46.
5. Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19-34.
6. Braiki, K., & Youssef, H. (2018, June). Multi-objective virtual machine placement algorithm based on particle swarm optimization. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)* (pp. 279-284). IEEE.
7. Alresheedi, S. S., Lu, S., Elaziz, M. A., & Ewees, A. A. (2019). Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. *Human-centric Computing and Information Sciences*, *9*(1), 15.
8. Zheng Q, Li Jia, Dong Bo, Li Rui, Shah Nazaraf, Tian Feng (2016) Multi-objective optimization algorithm based on bbo for virtual machine consolidation problem. In: IEEE international conference on parallel and distributed systems. Piscataway, IEEE, pp 414–421
9. Tang M, Pan S (2015) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Process Lett 41(2):211–221
10. Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. J Comput Syst Sci 79(8):1230–1242
11. Shabeera, T., Kumar, S.M., Salam, S.M., Krishnan, K.M.: Optimizing VM allocation and data placement for data-intensive applications in the cloud using ACO metaheuristic algorithm. Eng. Sci. Technol. Int. J. 20(2), 616–628 (2017)

12. Liu, X.-F., Zhan, Z.-H., Deng, J. D., Li, Y., Gu, T., Zhang, J: An energy-efficient ant colony system for virtual machine placement in cloud computing. In: Proceedings of the IEEE Transactions on Evolutionary Computation (2016)
13. Zhang, L., Wang, Y., Zhu, L., Ji, W.: Towards energy-efficient cloud: an optimized ant colony model for virtual machine placement. J. Commun. Inform. Netw. 1(4), 116–132 (2016)