[1] Dr. T. Saravanan
[2] Dr. B. Senthilkumaran
[3] Dr. John T Mesia Dhas
[4] Dr. A. Muthukrishnan

An Ant Colony Optimization Algorithm for Virtual Machine Placement in Cloud Computing.

Journal of Electrical Systems

Abstract: - The cloud computing (CC) environment is defined as the separation of workload, computing characteristics, and more. In the future, companies will manage workload or supply requests by allocating resources to computers to schedule tasks. , network or server. In this research, a new planning model is proposed as a hybrid of ant colony optimization techniques (ACO+). This technique increases the ant population to reduce the search space and allows the ACO technique to identify extensive paths accordingly. The Ant Colony Optimization (ACO) algorithm works by selecting the best virtual machine at the shortest possible distance from a point to a straight line. Use point-to-line spacing. The best virtual machine is selected from this. The submitted ACO+ implements an efficient method for identifying the best VMs that consume the least energy and improves fundamental tools such as overall response time for proper resource VM placement and optimization tasks. The results of this coupling simulation show that the submitted "Anti-Colony-Optimization-Plus" shows effective performance compared to other algorithms. Simulation Results: The Ant Colony Optimization Plus algorithm achieves effective results with a minimum energy consumption of 72%. This is superior to all comparison algorithms like K-Means clustering and Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), etc.

Keywords: Cloud Computing, Task Scheduling, Ant Colony Optimization Plus (ACO+), Cloudsim Stimulation, Virtual Machine Placement, Particle Swarm Optimization, Ant Colony Optimization, K-Means Clustering.

1. INTRODUCTION

Cloud computing, saving computing and storage capacity to support a group of people or end users. The term is derived from the use of the cloud-shaped figure as a concept for different communications that each node in structural figures contains. Cloud computing guarantees service support with user data, computation over a wide network and software. Basically, there are three main types of cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Customers can also pay for application software and databases through Software as a Service. The cloud service provider takes care of the infrastructure and location on which the application runs. Assistants leverage cloud-based applications via web browsers, lightweight desktop or mobile applications, as well as critical data storage software and user data loaded onto cloud servers in remote locations. End users claim that cloud computing can help companies launch and manage applications faster, with better management and less storage space. In addition, cloud systems enable the IT industry to quickly adapt resources to unpredictable and changing business needs.

The most explicit task processing tasks are delegated to different virtual machines or servers. These are usually triggered immediately during the activity. In addition to editing work, scientists also conduct research, comprehensive research-oriented editing work, or I-a-a-S cloud frameworks. Be that as it may, most existing frameworks have limitations in terms of cost, complexity and advanced information base design.

Distributed computing offers companies the opportunity to work on data innovation by transferring responsibility for programming and the associated help and support obligations to cloud specialist collaborations. The ability to add capacity as needed reduces costs and converts CAPEX into OPEX. However, such major changes in the way infrastructure is managed come with risks. To ensure the benefits are realized, it is important to make strong changes during the transition. This article discusses the placement of virtual machines geographically within a connected network to protect the data from natural disasters and utilize the nearby virtual machine. This saves a lot of energy to connect a virtual machine, and the performance of the machine is also very high. Several experiments are conducted using "n" VMs in Cloudsim and the tasks are scheduled for each existing and future

[1] Assistant Professor, Department of Artificial Intelligence, M.Kumarasamy College of Engineering (affiliated to anna university, Chennai, Tamilnadu), Karur - 639113, India. [1]drsaravanant.ml@gmail.com
[2]Assistant Professor, Department of Computer Science and Engineering, School of Computing, Veltech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology (deemed to be university, Estd. u/s 3 of UGC Act, 1956)
[3,4] Associate Professor, Department of Computer Science and Engineering, School of Computing, Veltech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology (deemed to be university, Estd. u/s 3 of UGC Act, 1956)

algorithm. The scores obtained show that the proposed algorithm consumes much less energy by using the closest path of the ACO algorithm.

The rest of the book is organized as follows: Section II provides an overview of the VMP problem model and its ongoing application. The third section describes the ACO+ algorithm. Section IV then applies to a variety of cloud computing environments, including 15 identical and 15 multi-server VMP displays that utilize interfering resources. Section IV then applies to different cloud computing environments where 15 identical server VMP displays and 5 multiple server VMPs exploit sources of interference. The performance and efficiency of particle mass optimization (PSO) [30], ACO-based methods [10] and horticultural methods such as hybrid ACO and PSO + [37] and ACO + are compared. Check. Finally, the results are presented in Section V.

2. RELATED WORK

Cloud computing (CC) offers the opportunity to facilitate information technology operations by delegating ownership of devices and software, as well as associated support and maintenance tasks, to cloud service providers. Cloud computing resources can be requested and used by multiple cloudlets simultaneously. End users can access centralized cloud computing resources via the Internet anytime, anywhere.

Z. H. Zhan, J et al [1] followed It introduces the principle of the ASS problem in the form of a sequence problem. It proposes a new solution framework, the first attempt to solve it using the Ant Colony System (ACS) algorithm based on Retreat Horizon Control (RHC). The result is that the RHC-enhanced ACS algorithm (known as the RHC-ACS-ASS algorithm) for the ASS problem is relevant and proven to be robust. The ACS algorithm with robust global search capability is effective and efficient. Not only for these types of NP-difficult problems, but RHC methods can also divide the problem into time-frame regression to reduce the calculation and improve the quality of the solution.

R.Buyya, S.Pandey, and C.Vecchiola, [2] followed the below listed principles. (1) It presents Forecasting and Identifying Various Information Technology Paradigms for the 21st Century that are expected to provide computers as an application. (2) Defining the framework for creating a marketable cloud computing environment using technologies such as virtual machines. (3) Provides market surveys. A basic resource management strategy that incorporates customer-driven service management and computational risk management to maintain SLA-based resource allocation. (4) Presenting work done as a component of another distributed computing drive called Cloudburst.

Y. Q. Gao, et al, A typical and home framework calculation for virtual machine position issues. The goal is to effectively get a package of non-dominant solutions (parallel packages) while minimizing the number of wasted resources and power consumption. The proposed algorithm has been tested in some examples in the literature. Solution performance is comparable to existing multi-purpose genetic algorithms and two single-purpose algorithms, a well-known bin-packing algorithm and a max-min ant system (MMAS) algorithms.

M.Stillwell, D.Schanzenbach, F.Vivien, and H. Casanova, [4] propose Creating resource allocation issues on a shared hosting site for constant workloads provided by the server with multiple types of resources. Our development supports a combination of optimal effort and QoS programs and utilizes well-defined objective functions to improve performance, fairness and cluster utility. In addition, this formula allows to calculate the optimal limit of resource allocation.

S.Chaisiri, B et al [5] propose an optimal virtual machine placement (OVMP) calculation. Given the vulnerability of future interest and evaluation, the calculation might diminish the expense of a venture to have virtual machines in different cloud supplier conditions. The OVMP calculation will settle on choices in view of the standard arrangement of standard whole number programming (SIP) for leasing assets from cloud suppliers. The exhibition of the OVMP calculation is assessed by mathematical examinations and reproductions. The outcomes plainly show that the proposed OVMP calculation will lessen the client's spending plan.

Table 1: Summary of related works.

Reference	Objective	Algorithm
1	New solution framework that makes the main attempt at using an ant colony system (ACS) algorithm based on the receding horizon control (RHC) to solve it.	Ant Colony System (ACS) algorithm
2	A vision for computing in the 21st century and identifies various information technology paradigms that promise computing as an application.	PSO algorithm
3	The objective is to productively get many When Pareto sets to lessen complete asset wastage and power utilization.	Multi-objective ant colony system algorithm
4	Resource allocation issues on shared hosting sites that use static workloads on servers serving multiple resources.	Ant Colony Optimization (ACO) algorithm

5	Future demand and price uncertainty can reduce the cost of a project to host virtual machines in multiple cloud hosting environments.	Optimal virtual machine placement (OVMP) algorithm

Recent work has been done in this area of research [4, 5, 6, 7, 8], where research [4] proposes ACO, a new way of planning tasks in a variety of categories, such as large, medium, and small, using clients with the lowest number of energy. Table 1 summarizes the review work on its objectives and basic mechanisms. Although there are many works in the literature, some of these methods do not consider that the energy parameters of the CC environment should improve performance in many ways.

3. PROPOSED SYSTEM

The proposed ACO + algorithm includes the Hybrid Any colony Optimization (ACO) algorithm. The proposed ACO + algorithm has lower energy consumption, more efficiently reduces production time, and strengthens basic functions such as system resource utilization and work response time. The following subsections describe the steps in the proposed algorithm. The placement of virtual machines within an interconnected network is being distributes geographically in order to safeguard the data from natural disasters and to utilize the nearby virtual machine. So that the energy consumed to connect virtual machine is being saved a lot and the performance of the machine is also very high. Several experiments are conducted with the help of 'n' number of VM's in cloudsim and the job are scheduled against each existing and proposed algorithms. The obtained results shows that proposed algorithm consume very less energy by utilizing the nearest path of ACO algorithm.

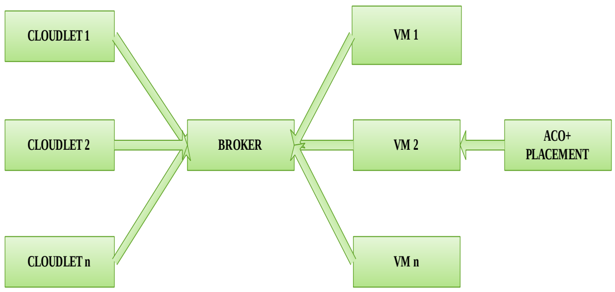

Fig 1: Architecture Diagram

A) K-Means Clustering Algorithm

Clustering is perhaps the most utilized exploration information investigation strategy to give an intuitive comprehension of information structure. It is characterized as the errand of recognizing subgroups in information. Hence, the relevant items of a solitary subgroup (bunch) are practically the same, and the important elements of various groups are altogether different. Euclidean tries to track down comparable subgroups in the information as indicated by the level of comparability, for example, distance or contact-based distance, making the main items in each bunch as indistinguishable as possible expected. Figuring out which similitude ought to be utilized is application dependent.

The K-Means calculation is a calculation that attempts to isolate a bunch of information into pre-characterized, detached subgroups (groups) where every information point has a place with a similar gathering. Attempt to make the important informative items in the bunch as uniform as expected while keeping the group as various (a ways off) as could be expected. Dole out information focuses to the group to diminish the number of squares of the distance between the main items and the focal point of the bunch (math mean of all information guides having a place toward the group). The less variety inside a bunch, the more (indistinguishable) focuses the information focuses will be inside a similar group.

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

If the information point xi has a place with group k, then, at that point, wik = 1; Otherwise, wik = 0. Moreover, μk is the centroid of the xi group.

In other words, the distance from the cluster centroid is determined by the sum of the squares, assigning data point's xi to the nearest cluster.

And M-step is:

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

This means recalculating the center of each cluster to reflect the new allocation.

$$\frac{1}{m_k}\sum_{i=1}^{m_k} \|x^i - \mu_{c^k}\|^2$$

Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a biologically inspired algorithm that easily finds the optimal solution at the solution site. Unlike other optimization algorithms, only objective functionality is required. It does not depend on the desired inclination or different shape. Also, there are virtually no hyper parameters.

Kennedy and Ever hard proposed particle mass optimization in 1995. As noted in the original paper, social biologists believe that groups of fish and birds can "benefit from the experience of all other members." In other words, for example, when the birds are flying and roughly searching for food, all the birds in the herd share their finds and help the whole flock to hunt better.

In the Particle Swarm Optimization (PSO) Algorithm Proposed by Kennedy and Anytime hard is a meta-heuristic calculation in light of the idea of multitude knowledge, which tackles complex numerical issues in designing. It is critical to take note of that PSO execution enjoys numerous upper hands over other improvement calculations, as there are less boundaries to tune. What should set up is generally talked about in the writing

Consider the mass of particles P. For each of the I particles that make it up, and there is a level vector Ait = xi1 xi2 xi3… AinT and a speed vector Xit = Xi1 Xi2 Xi3 ... XinT. These vectors are invigorated by aspect j as indicated by the accompanying condition:

Aijt+1=wAijt+c1r1tpbestij−Vijt+c2r2tgbestj−Xijt

And

Aijt+1=Aijt+Xijt+1F2

where i = 1,2,…,P and j = 1,2,…,n.

Equation (1) denotes since three distinct commitments to molecule movement in a redundancy, three unique terms should be additionally clarified. On the other hand, Equation (2) updates the particle position.

B) Ant Colony Optimization(ACO) Algorithm

Ant Colony Optimization (ACO), developed by Marco Torico in 1992, was the first method based on mass intelligence. Essentially, ACOs simulate the eating behaviour of swarming ants, while pheromones are used to simulate local interactions and communication between ants (Torico, 1992). Each ant deposits pheromones, which gradually evaporate over time. The exact shape of the evaporation model depends on the variation and shape of the ACO used in the implementation. Both gradual pheromone deposition and rapid decomposition are widely used.

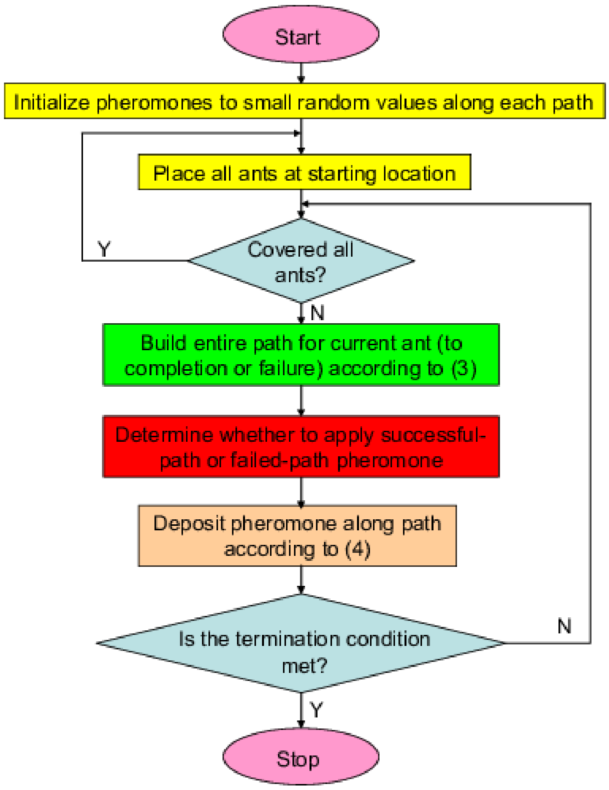

Fig 6: Flowchart of ACO Algorithm

The ACO algorithm is robust and has a well-distributed system technique. ACO can be easily integrated with other methods. Excellent at solving complex optimization problems. ACO has an enhanced pheromone path and improves the probability of changing the region and the problem by moving these ants into the search area according to a simple aggregate pheromone formula. At every cycle, the ACO makes a worldwide insect and works out its feasibility. Further develop pheromones and powerless zone edges. Assuming you feel good, move your neighborhood insects to a superior area. In any case, select another irregular pursuit heading. Invigorate subterranean insect pheromones and vanish subterranean insect pheromones. Consistent ACO depends on neighborhood and worldwide hunt. Nearby subterranean insects can move towards potential regions involving the best answer for region k change probabilities.

Equation: 1

Where, Total pheromones in t_k (t) furthermore, n is the number of ant insects.

Pheromones are updated using the following formula.

$$t_i\ (t+1) = (1-r)\, t_i\ (t)$$

Equation: 2

Where r is the pheromone evaporation rate.

C) Ant Colony Optimization Plus (ACO+)

An ACO + algorithm is designed to enhance the search engine by combining the tabu search algorithm. Provides all the best free virtual machines for ACO + output. Henceforth, the output will be considered as the input of the block list, where the trainer will find a virtual machine capable of performing that particular task. This behavior provides better output at the energy and production intervals.

ACO + is particularly suitable for personal optimization issues. For example, during a routing problem, the path or path is encrypted as a solution. As ants explore different pathways, the pathways studied are marked by pheromones that evaporate and deposit over time. Path fit or quality (solution) is related to the pheromone concentration in the path. The path with the highest pheromone concentration will be the preferred path or the path with the highest probability at the intersection will be selected. Like GA, ACO is a hybrid program with many types and applications (Dorigo, 1992).

$$p(ij) = \frac{\tau_{ij} \cdot \eta_{ij}^{\beta}}{\sum_{g \notin \mathscr{B}} \tau_{ig} \eta_{ig}^{\beta}},$$

Equation: 1

Where A horoscope called Visibility selects the nearest town ηij = 1 / dij. Pheromone enhancement rules

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij} \quad \text{with} \quad \Delta\tau_{ij} = \sum_{k=1}^{m} \Delta_{ij}^{k}.$$

Equation: 2

The pheromone contribution of ant k is

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ used edge } (i,j) \text{ in its last tour,} \\ 0 & \text{otherwise.} \end{cases}$$

4. Result and Discussion

4.1 Performance Validation

A) The proposed ACO + calculation was mimicked utilizing the Cloudsim device. All assignments have login and run time, are allowed to get everything rolling and are given utilizing workstation programming. To compare, we used a set of algorithms. Particle Mass Optimization (PSO), Ant Colony Optimization (ACO), Ant Colony Optimization (ACO +). The measured values of the study results are energy consumption time and production time, respectively.

B) Implementation setup

Table 2 shows the sizes of the different types of planned tasks and the number of tasks associated with them. As you can see from the table, the magnified tasks will contain 800-1000 tasks, with task sizes ranging from 100,000 to 200,000 MI. Similarly, large tasks can contain 600-700 tasks, with task sizes ranging from 700 to 10000 MI. Similarly, medium jobs will have 400-500 jobs with 50,000-70000 MI work sizes. Similarly, small tasks can contain 100-200 tasks, of which 30000-50000 MI level. In addition, the task size is generated approximately at run time and the size is defined by millions of algorithms (MIs). All of these types of tasks are performed and tested in a cloud environment where the VM size is set to 50.

Table 2: Summary of different type of tasks.

Tasks Types	No. of tasks	Size of tasks (MI)
Extra-large tasks	700 – 1000	100000-200000
Large tasks	500 – 700	50000–100000
Medium tasks	300 – 500	40000–70000
Small tasks	100 – 200	20000–50000

C) The results of the production gap

Figure 7 provides a detailed comparison of Mac voltages between different scheduling methods and ACO+ algorithms. The graphic shows that ACO+ delivers better results than comparable planning methods. If you want to measure results with very large tasks and Mac spans, keep in mind that the PSO and ACO algorithms require Mac spans of up to 630432 ms and 412427 ms, respectively. On the other hand, the ACO algorithm requires an MS margin of 511629. However, the proposed ACO+ algorithm is efficient, with a minimum margin of 412427 ms. Remember that the lifespan of your Mac gradually increases with the number of tasks it performs.

Table 3: Make span Calculation.

Number of tasks / Algorithms	K-Means	PSO	ACO	ACO+
200	432427	630432	511629	412427
400	4621617	5417622	4940019	4621617
600	33274450	36862460	34709654	33274450
800	41018263	44210268	42295065	41018263
1000	81647271	86637276	83643273	81647271

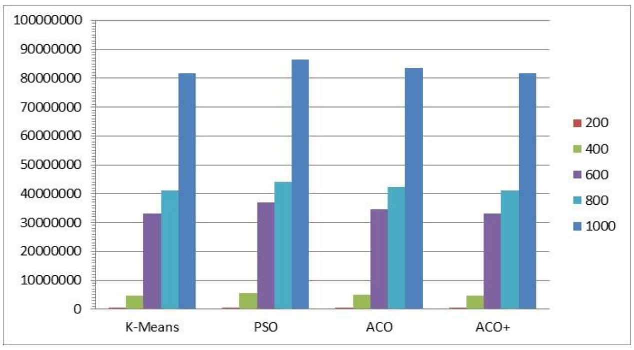

Fig 10: Comparative results analysis of Mac span.

Analysis of results in terms of energy

Figure. 8 Provides a detailed comparison of the power of different programs methods and ACO + algorithms. This figure shows that ACO + gives better results than comparable planning algorithms. Keep in mind that the PSO and ACO algorithms require a maximum Mac span of 938 and 910, respectively, when measuring the results of very large tasks and energies. On the other hand, the ACO algorithm requires a competitive production interval of 511629. However, the proposed ACO + algorithm is valid at the minimum production interval required at 412427. It is necessary to note that the production interval gradually increases with the number of tasks.

Table 4: Energy Calculation.

Energy Consumed (Joules)/ Algorithms	K-Means	PSO	ACO	ACO+
200	986.90	910.00	933.70	938.70
400	1949.70	1885.20	1843.90	1856.90
600	5695.30	5672.20	5695.20	5585.80
800	3784.29	3766.39	3782.00	3787.09
1000	4725.20	4697.59	4702.90	4732.30

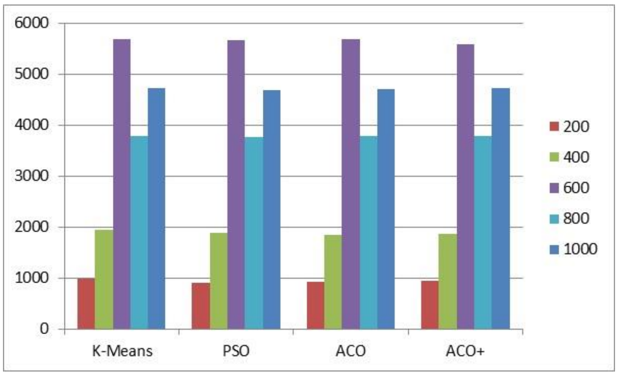

Fig 11: Comparative results analysis in terms of energy.

5. Conclusion

This paper has introduced Cloud climate energy saving arranging calculation utilizing ACO + calculation. The proposed calculation consolidates the upsides of the ACO calculation and the untouchable inquiry calculation. The proposed ACO + calculation empowers and improves the normal power compelling for making basic estimations, for example, ideal energy use and reaction season of the undertaking. For this test, the arrangement of estimations used to break down the outcomes is the development and utilization of energy. Reenactment results show that the proposed ACO + model works better contrasted with the connection procedure. The simulation results show that the ACO + algorithm achieved effective results with a minimum average production interval of 412,427, energy consumption of 72%, and all other comparisons with a production interval of 412,427. There are ACO + algorithm improvements to use the detection algorithm for future purpose.

REFERENCES

[1] Z. H. Zhan, K. J. Du, J. Zhang, and J. Xiao, “Extended binary particle swarm optimization approach for disjoint set covers problem in wireless sensor networks,” in Proc. Conf. Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 2012. pp. 327-331.

[2] R. Buyya, S. Pandey, and C. Vecchiola, “Cloudbus toolkit for market-oriented cloud computing,” in Proc. 1st International Conference on Cloud Computing, 2009, pp. 24–44.

[3] Y. Q. Gao, H. B. Guan, Z. W. Qi, Y. Hou, and L. Liu, “A multiobjective ant colony system algorithm for virtual machine placement in cloud computing,” Journal of Computer and System Sciences, vol. 79, no. 8, pp. 1230-1242, Dec. 2013.

[4] M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova. “Resource allocation algorithms for virtualized service hosting platforms”, Journal of Parallel and Distributed Computing, vol. 70, no. 9, pp. 962-974, September 2010.

[5] S. Chaisiri, B. S. Lee, and D. Niyato, “Optimal virtual machine placement across multiple cloud providers”, in Proc. IEEE Asia- Pacific Services Computing Conference, 2009, pp. 103–110.

[6] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. Chung, and Y. H. Shi, “Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems,” IEEE Transactions on Cybernetics, vol. 43, no. 2, pp. 445-463, April. 2013.

[7] M. Shen, Z. H. Zhan, W. N. Chen, Y. J. Gong, J. Zhang, and Y. Li, “Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks,” IEEE Transactions on Industrial Electronics, in press 2014.

[8] E. Feller, L. Rilling, and C. Morin, “Energy-aware ant colony based workload placement in clouds”, in Proc. IEEE/ACM International Conference on Grid Computing (GRID), 2011, pp. 26-33.

[9] H. H. Li, Y. W. Fu, Z. H. Zhan, and J. J. Li, “Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling,” in Proc. IEEE Congr. Evol. Comput., 2015, pp. 870-876.

[10] Z. H. Zhan, X. F. Liu, H. Zhang, Z. Yu, J. Weng, Y. Li, T. Gu, and J. Zhang, “Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version,” IEEE Trans. Parall. Distribut. Systs, DOI: 10.1109/TPDS.2016.2597826, 2016.

[11] X. F. Liu, Z. H. Zhan, J. H. Lin, and J. Zhang, “Parallel differential evolution on distributed computational resources for power electronic circuit optimization,” in Proc. Genetic Evol. Comput. Conf., 2016, pp. 117-118.

[12] H. H. Li, Z. G. Chen, Z. H. Zhan, K. J. Du, and J. Zhang, “Renumber coevolutionary multiswarm particle swarm optimization for multi-objective workflow scheduling on cloud computing environment,” in Proc. Genetic Evol. Comput. Conf., 2015, pp. 1419-1420.

[13] Z. G. Chen, Z. H. Zhan, H. H. Li, K. J. Du, J. H. Zhong, Y. W. Foo, Y. Li, and J. Zhang, “Deadline constrained cloud computing resources scheduling through an ant colony system approach,” in Proc. Int. Conf. Cloud Computing Research and Innovation, 2015, pp. 112-119.

[14] R. De S. Couto, S. Secci, M. E. Mitre Campista, and L. H. M. Kosmalski Costa, “Network design requirements for disaster resilience in IaaS clouds,” IEEE Commu. Magazine., vol. 52, no. 10, pp. 52-58, Oct. 2014.

[15] R. K. Gupta and R. K. Pateriya, “Energy efficient virtual machine placement approach for balanced resource utilization in cloud environment,” Int. J. of Cloud-Computing and Super-Computing, vol. 2, no. 1, pp. 9-20, 2015.

[16] Y. W. Foo, C. Goh, H. C. Lim, Z. H. Zhan, and Y. Li, “Evolutionary neural network based energy consumption forecast for cloud computing,” in Proc. Int. Conf. Cloud Computing Research and Innovation, 2015, pp. 53-64.

[17] M. H. Ferdaus, M. Murshed, R. N. Calheiros, R. Buyya, “Virtual machine consolidation in cloud data centers using ACO metaheuristic,” in Euro-Par 2014 Parallel Processing, vol. 8632, F. Silva, I. Dutra, and V. S. Costa, Eds. Springer International Publishing, 2014, pp. 306-317.

[18] B. B. J. Suseela and V. Jeyakrishnan, “A multi-objective hybrid ACO-PSO optimization algorithm for virtual machine placement in cloud computing,” International Journal of Research in Engineering and Technology, vol. 3, no. 4, pp. 474-476, 2014.

[19] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, “Using ant colony system to consolidate VMs for green cloud computing, ” IEEE Trans. Services Comput., vol. 8, no. 2, pp. 187-198, Mar. 2015.

[20] X. F. Liu, Z. H. Zhan, K. J. Du, and W. N. Chen, “Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach”, in Proc. ACM Genetic Evol. Comput. Conf., 2014, pp.41-48.

[21] B.A. Al-Maytami, P. Fan, A. Hussain, T. Baker, P. Liatsis, A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing, IEEE Access 7 (2019) 160916–160926.

[22] S. Belguith, N. Kaaniche, M. Hammoudeh, T. Dargahi, PROUD: verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IoT applications, Future Gener. Comput. Syst. (2019).

[23] Y. Wang, Y. Guo, Z. Guo, T. Baker, W. Liu, CLOSURE: A cloud scientific workflow scheduling algorithm based on attack–defense game model, Future Gener. Comput. Syst. (2019).

[24] Q. Xu, R. V. Arumugam, K. L. Yong, Y. Wen, Y. S. Ong, and W. Xi (2015). Adaptive and scalable load balancing for metadata server cluster in cloud-scale file systems. Frontiers of Computer Science, 9(6), 904-918.

[25] R. M. Singh, S. Paul and A. Kumar, “Task scheduling in cloud computing”, International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014.

[26] G. Xie, G. Zeng, R. Li & K. Li, ”Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing,” IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 62-75, 2017.

[27] Y. Chen, C. Lin, J. Huang, X. Xiang, & X. Shen, ”Energy efficient scheduling and management for large-scale services computing systems,” IEEE Transactions on Services Computing, no. 1, pp. 1-1, 2017.

[28] G. Xie, G. Zeng, R. Li & K. Li, ”Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing,” IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 62-75, 2017.

[29] S. Sahoo, B. Sahoo, & A. K. Turuk, ”An Energy-Efficient Scheduling Framework for Cloud Using Learning Automata,” In 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-5, 2018.

[30] P. Zhang, & M. Zhou, ”Dynamic cloud task scheduling based on a two-stage strategy,” IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, 2018.
[31] X. Li, X. Jiang, P. Garraghan, & Z.Wu, ”Holistic energy and failure aware workload scheduling in Cloud datacenters,” Future Generation Computer Systems, vol. 78, pp. 887-900, 2018.
[32] S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, & E. Dutkiewicz, ”Sustainable Service Allocation using Metaheuristic Technique in Fog Server for Industrial Applications,” IEEE Transactions on Industrial Informatics, 2018.
[33] H. Chen, X. Zhu, G. Liu and W. Pedrycz, ”Uncertainty-Aware Online Scheduling for Real-Time Workflows in Cloud Service Environment,” IEEE Transactions on Services Computing, pp. 1-1, 2018.
[34] M. Ranjbari, & J. A. Torkestani, ”A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers,” Journal of Parallel and Distributed Computing, vol. 113, pp. 55-62, 2018.
[35] A. A. Rahmanian, M. Ghobaei-Arani, & S. Tofighy, ”A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment,” Future Generation Computer Systems, vol. 79, pp. 54-71, 2018.