



**HAL**  
open science

# Enhanced Ant Colony Optimization Algorithm for Optimizing Load Balancing in Cloud Computing Platform

A. Daniel, N. Partheeban, Srinivasan Sriramulu

► **To cite this version:**

A. Daniel, N. Partheeban, Srinivasan Sriramulu. Enhanced Ant Colony Optimization Algorithm for Optimizing Load Balancing in Cloud Computing Platform. 4th International Conference on Computational Intelligence in Data Science (ICCIDS), Mar 2021, Chennai, India. pp.64-70, 10.1007/978-3-030-92600-7\_6. hal-03772927

**HAL Id: hal-03772927**

**<https://inria.hal.science/hal-03772927v1>**

Submitted on 8 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Enhanced Ant Colony Optimization Algorithm for Optimizing Load balancing in Cloud Computing Platform

A.Daniel<sup>1</sup>, N.Partheeban<sup>2</sup>, Srinivasan Sriramulu<sup>3</sup>

<sup>1</sup>Assistant Professor, Galgotias University,UP.

<sup>2,3</sup>Professor, Galgotias University,UP.

<sup>1</sup>danielarockiam@gmail.com, <sup>2</sup>n.partheeban@galgotiasuniversity.edu.in,

<sup>2</sup>s.srinivasan@galgotiasuniversity.edu.in

**Abstract.** Load balancing among virtual machines (VMs) is significant for conveying the cloud services in advanced manner with least cost paid and all-out time procured to convey the services. In this research work, the different research holes for load balancing optimization in the past writing have been exhibited, which should be tended to for taking care of the load balancing issue in cloud environment. In present work, Hybrid methodology-based asset provisioning and load balancing framework for work processes execution has been proposed to enhance the usage of VMs with uniform load conveyance. The proposed framework depends on binary ant colony optimization algorithm and the results proved the superiority of them over the compared methods.

**Keywords:** Cloud; Load Balancing; Ant Colony; Virtual machine.

Load balancing is important for enhanced utilization of cloud resources (processors, memory, circles) and to get superior of the machines. The resources are allotted and used by VMs facilitating on physical machines. At the point when undertakings are booked on VMs there may emerge a circumstance that a portion of the VMs are over utilized, while others stay under-used.

Load balancing techniques are utilized to ensure that each machine in the cloud datacenter performs around the equivalent number of assignments anytime of time. In cloud computing, the requests of client are exceedingly unique in nature and multi-tenure requires confining diverse clients from one another and from the cloud foundation. Distinctive heuristic and metaheuristic techniques have been utilized for circulating load among accessible VMs and to get ideal execution of VMs.

The computational expense of metaheuristics is higher than heuristics techniques since they have bigger inquiry space and their hunt criteria depends on guided arbitrary look to get scan answer for the given planning issue. Heuristics are utilized to diminish the scan space for metaheuristics to improve their combination rate to achieve the ideal arrangement in least conceivable time. Notwithstanding, the issue of load balancing is multi-objective, with the end goal that the essential target is to convey occupations or errands among accessible resources so the relative lop- sidedness is limited; and the optional goal is to boost the use of resources while lessening makespan and cost.

In this examination paper, a framework for resource provisioning and load balancing has been proposed and actualized. The proposed framework depends on binary ant colony optimization algorithm to accomplish its ideal execution as far as makespan and cost. A framework for

Workflow's execution has been proposed to give ideal load to underutilized VMs and advance their use.

## 2 Literature Review

The load balancing techniques for cloud computing with various heuristic and metaheuristic techniques explicitly for load balancing optimization have been looked into. Different load balancing techniques have been recommended by the analysts. The exploration holes have been found amid the past writing study. [1] proposed Load Balancing Ant Colony Optimization (LBACO) algorithm for balancing the load crosswise over whole framework and minimize makespan.

The outcomes are observed to be superior to fundamental ACO and FCFS algorithms. The undertakings have been viewed as commonly autonomous with no priority requirement, nonprimitive and calculation serious. The creators considered free undertakings as contribution with no priority and spending requirements among them. Additionally, ACO parameters have been instated arbitrarily.

[2] proposed load balance improved min-min scheduling (LBIMM) algorithm which expanded asset usage and diminished makespan. Be that as it may, due dates regarding execution time and cost have not been given significance in their work. Further, [3] thought about makespan of ACO-based cloud task scheduling with FCFS and Round Robin algorithms and presumed that ACO beat the two partner algorithms in any case, reliance limitations among errands have not been considered, [4] proposed ACO-based load balancing optimization to allot balanced load of errands among VMs.

They have taken just finish time algorithm examination with predetermined number of VMs (up to 4) and no other measurement has been utilized to assess its execution. [5] proposed and actualized a dynamic load the executive's algorithm for viable circulation of solicitations among VMs. Be that as it may, the reaction time expanded when underloaded VMs were checked

[6] proposed multi-objective ACO (MO-ACO) based undertaking scheduling and load balancing algorithm with makespan and cost optimization. By and by, non-pre-emptive scheduling has been done on autonomous undertakings which are not appropriate for cloud environment. Since, load balancing alludes to managing hubs or VMs which are used beneath their minimum limit. This outcomes in wastage of computing asset if not used to their ideal point of confinement.

At the point when VMs are over utilized, the all-out time taken by a VM to finish every one of the assignments apportioned to it, i.e., its makespan, additionally increments. Be that as it may, when the VMs are underutilized, in spite of the fact that the makespan decline however it results in expanded expense of asset usage as the accessible resources (VMs) are not broadly used bringing about their wastage. Subsequently, there is a need to balance the load over the VMs with the goal that both the makespan and cost parameters must be controlled and balanced. The abatement in makespan must not result in increment of expense of asset use and the other way around.

In past research works, load balancing optimizations has been finished utilizing diverse metaheuristics yet in practically all the proposed techniques, the meta-heuristics are furnished with irregular seed for their parameters [7]. The arbitrary introduction of metaheuristic parameters can corrupt the execution of the method.

The greater part of the load balancing approaches proposed in writing spotlight on minimizing just the execution time yet don't consider the execution cost [8]. Cost is a significant parameter as the cloud computing services are conveyed on pay per use dependent on web. What's more, the priority request of assignment execution with requirement on execution time and cost has not given due significance in past works to such an extent that the errands which don't satisfy these limitations (due dates) ought not to be executed [9-11].

### 3 Proposed Method

#### 3.1. Solution Construction

In BAS, artificial ants construct solutions by walking on the mapping graph as described in Fig.1. At every iteration  $i$ , a number of ants  $n_a$  cooperate together to search in the binary solution domain, each ant constructs its solution by walking sequentially from node 1 to node  $n + 1$  on the routing graph. At each node, ant either selects the upper path  $i0$  or the selects lower path  $i1$  to walk to the next node  $i+1$ .  $i0$  Selecting means  $x_i=0$ ; and  $i1$  selecting means  $x_i=1$ . The selecting probability is dependent on the pheromone distributed on the paths.

$$p_{is}(t) = \tau_{is}(t), i = 1, \dots, n, s \in \{0,1\} \quad (1)$$

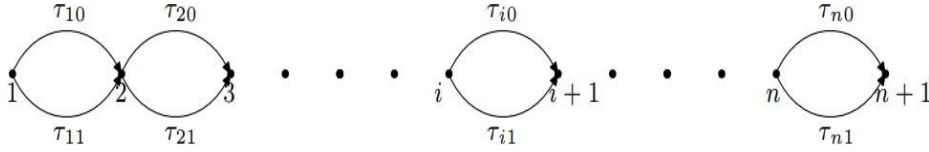


Fig. 1: Routing Diagram for Ants in BAS

The solutions constructed by the ants may not be feasible when tackling the constrained binary optimization problems. A solution repair operator is incorporated to transfer the infeasible solutions to the feasible domain.

#### 3.2 Pheromone Update

Initially, BAS set all the pheromone values as  $\tau_{is}(0) = 0.5$ , which is the same as that of *HCF*[8], but uses a simplified pheromone update rule:

$$\tau_{is}(t+1) \leftarrow (1 - \rho)\tau_{is}(t) + \rho \sum_{x \in S_{upd} | is \in x} w_x \quad (2)$$

here  $S_{upd}$  is the set of solutions to be intensified  $w_x$  are explicit weights for each solution  $x \in S_{upd}$

$$0 \leq w_x \leq 1 \quad \rho = \rho_0$$

which satisfying and

$\sum_{x \in S_{upd}} w_x = 1$  is the evaporation parameter, which is set initially as, but decreases as  $\rho \leftarrow 0.9\rho$  every time the pheromone re-initialization is performed

$S_{upd}$  consists of three components, they are: the global best solution  $S^{gb}$ , the iteration best

	$cf < cf_1$	$cf \in (cf_1, cf_2)$	$cf \in (cf_2, cf_3)$	$cf \in (cf_3, cf_4)$	$cf \in (cf_4, cf_5)$
$w_{ib}$	1	2/3	1/3	0	0
$w_{rb}$	0	1/3	2/3	1	0
$w_{gb}$	0	0	0	0	1

solution  $S^{ib}$ , and the restart best solution  $S^{rb}$ .

Different combinations of  $W_x$  are implemented according to the convergence status of the algorithm. The convergence status is monitored by a convergence factor  $cf$ , which is defined as

$$cf = \frac{\sum_{i=1}^n |\tau_{i0} - \tau_{i1}|}{n} \quad (3)$$

Under this definition, when the algorithm is initialized with  $\tau_{is}(0) = 0.5$  for all the paths,  $cf = 0$ , while when the algorithm gets into convergence or premature,  $|\tau_{i0} - \tau_{i1}| \rightarrow 1$ , thus that  $cf \rightarrow 1$ .

Table 1 describes the pheromone update strategy in different value of  $cf$ , where  $w_{ib}, w_{rb}$  and  $w_{gb}$  are the weight parameters for  $S^{ib}, S^{rb}$  and  $S^{gb}$  respectively,  $cf_i, i = 1, \dots, 5$  are threshold parameters within the range of  $[0, 1]$ .

In BAS, once  $cf > cf_5$ , the pheromone re-initialization procedure is performed according to  $S^{gb}$ :

$$\begin{cases} \tau_{is} = \tau H & \text{if } is \in S^{gb} \\ \tau = \tau L & \text{otherwise} \end{cases} \quad (4)$$

these two parameters satisfying  $0 < \tau L < \tau H < 1$  and  $\tau L + \tau H = 1$ . This kind of pheromone re-initialization will focus more on the previous search experience rather than a total redo of the algorithm

Table 1: Pheromone Update Strategy for BAS

## 4 Results and Discussion

To balance load among the VM's in cloud service is a composite task that needs to be done. The existing and proposed method performance is measured by means of increasing the number of VMs in cloud. Fig. 1 shows the results of the presented method in terms of average make span.

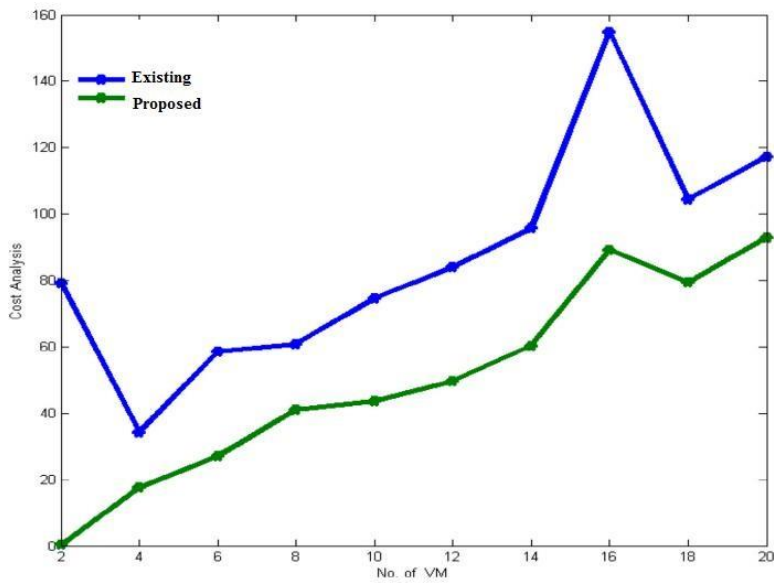


Fig. 2: Average Makespan (ms) Analysis

Fig. 2 shows comparison of existing and proposed method average makespan analysis while running the Workflow of genome in the framework of presented method. For Ligo workflow execution, while 2 VMs are employed as resource for the existing method, the ms achieved the cost analysis of 80 and decreased when deployed by 4 VMs and subsequently increased till 14 number of VMs, finally it increases drastically near 160 and end near 120 when the number of VMs are 20.

When 2 VMs are employed as resource for the existing method, the ms achieved the cost analysis is nearby zero and increased shortly when deployed by 4 VMs and subsequently decreased in 14 number of VMs, finally it increased a little bit near 100 when the number of VMs are 20. From the makespan results computed during execution of three scientific workflows on the proposed are better than existing approach.

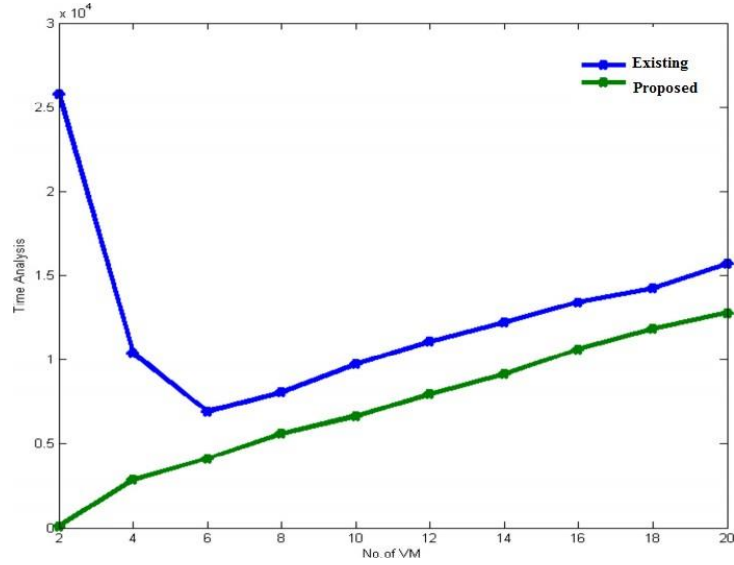


Fig. 3: Average Cost (Rs) Analysis

Fig. 3 shows comparison of existing and proposed method average time analysis while running the Workflow of Cybershake in the framework of HDD-PLB. When 2 VMs are employed as resource for the existing method achieved the time analysis of 2.5 and decreased shortly when deployed by 4 VMs and subsequently decreased in 14 number of VMs, finally it increased a little bit near 1.5 when the number of VMs are 20. While 2 VMs are employed as resource for the proposed method, the time analysis of 80 and increases when deployed by 4 VMs and subsequently increased till 14 number of VMs, finally it increases drastically near 1 when the number of VMs are 20. From the time analysis results computed during execution of three scientific workflows on the proposed are better than existing approach. The results of proposed method show consistent pattern of low cost incurred with small number of VMs and then rises slowly with increased VMs.

## 5 Conclusion

Load balancing techniques are utilized to ensure that each machine in the cloud data centre performs around the equivalent number of assignments anytime of time. In cloud computing, the requests of client are exceedingly unique in nature and multi-tenure requires confining diverse clients from one another and from the cloud foundation. The resources are allotted and used by VMs facilitating on physical machine. The proposed framework depends on binary ant colony optimization algorithm. From the time analysis results computed during execution of three scientific workflows on the proposed are better than existing approach. The results of proposed method show consistent pattern of low cost incurred with small number of VMs and then rises slowly with increased VMs.



## References

1. Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D., 2011, “Cloud Task Scheduling based on Load Balancing Ant Colony Optimization”, in IEEE 2011 Sixth Annual Chinagrid Conference (ChinaGrid), pp. 3-9.
2. Chen, H., Wang, F., Helian, N., Akanmu, G. “User-priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing”, 2013 *National Conference on Parallel Computing Technologies*, PARCOMPTECH 2013, 2013, pp. 1-8, IEEE.
3. Tawfeek, M. A., El-Sisi, A., Keshk, A. E., Torkey, F. A., 2013, Cloud Task Scheduling based on Ant Colony Optimization. In: Proceedings of 8th *International Conference on Computer Engineering & Systems (ICCES)*, pp. 64–69.
4. Xue, S., Li, M., Xu, X., Chen, J., 2014. An ACO-LB algorithm for task scheduling in the cloud environment. *J. Software* 9 (2), 466–473.
5. Panwar, R., Mallick, B., 2015, Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm. In: Proceedings of *IEEE International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 773–778.
6. Guo, Q., 2017. Task Scheduling Based on Ant Colony Optimization in Cloud Environment”, Proceedings of AIP Conference. *AIP Publishing*.
7. Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D., 2011, “Cloud Task Scheduling based on Load Balancing Ant Colony Optimization”, in IEEE 2011 Sixth Annual Chinagrid Conference (ChinaGrid), pp. 3-9.
8. Keskinturk, T., Yildirim, M.B., Barut, M., 2012. An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times. *Comput. Operat. Res.* 39 (6), 1225–1235.
9. Elhoseny, M., Shankar, K., Lakshmanaprabu, S.K., Maseleno, A. and Arunkumar, N., 2018. Hybrid optimization with cryptography encryption for medical image security in Internet of Things. *Neural Computing and Applications*, pp.1-15.
10. Karthikeyan, K., Sunder, R., Shankar, K., Lakshmanaprabu, S.K., Vijayakumar, V., Elhoseny, M. and Manogaran, G., 2018. Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA). *The Journal of Supercomputing*, pp.1-17.
11. Shankar, K., Elhoseny, M., Chelvi, E.D., Lakshmanaprabu, S.K. and Wu, W., 2018. An Efficient Optimal Key Based Chaos Function for Medical Image Security. *IEEE Access*, 6, pp.77145-77154.