# Improvised Bat Algorithm for Load Balancing-Based Task Scheduling

**Bibhav Raj, Pratyush Ranjan, Naela Rizvi, Prashant Pranav and Sanchita Paul**

**Abstract** The development of computing system has always focused on performance improvements driven by the demand of applications by customers, scientific and business domain. Cloud computing has emanated as a new trend as well as required domain for the efficient usage of computing systems. As the applications operating in cloud environments are becoming popular, the load is also rising on the servers and the traffic is increasing rapidly. In this paper, a new metaheuristic algorithm has been discussed known as improvised Bat algorithm and the case study of it is explained with proper example. The improvised Bat algorithm works on Min-Min, Max-Min and Alpha-Beta pruning algorithm for population generation and then uses the Bat algorithm for determining the sequence of execution of tasks to keep it minimum.

**Keywords** Algorithm · Cloud · Computing · Task · Load

B. Raj (✉) · P. Ranjan · N. Rizvi · P. Pranav · S. Paul
Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi 835215, Jharkhand, India
e-mail: raj.bibhav@gmail.com

P. Ranjan
e-mail: pratyushranjan@live.com

N. Rizvi
e-mail: naelarizvi92@gmail.com

P. Pranav
e-mail: prashantpranav19@gmail.com

S. Paul
e-mail: sanchita07@gmail.com



P.K. Sa et al. (eds.), *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*, Advances in Intelligent Systems and Computing 518,
DOI 10.1007/978-981-10-3373-5_52

# 1 Introduction

Cloud computing focuses on maximizing the capabilities of the shared resources. Cloud resources are not only shared among multiple users but are also dynamically reallocated as per demand. Load balancing distributes the workload among all the computing resources. In cloud computing, task scheduling determines proper sequencing of tasks to virtual machines for maximum utilization of the resources and increasing throughput. It focuses on keeping the available resources equally busy and avoids overloading any one of the machine or resource with many tasks. Load balancing is done so that every virtual machine in cloud system does the same amount of work, increase the throughput and minimize the response time. The algorithm schedules the task in such a way that the load is balanced effectively. The improvised Bat algorithm works using the Min-Min and Max-Min optimization technique to allocate and reduce the execution time of the tasks requested to the VMs. After the population is optimized, the Bat algorithm executes and determines the sequence of execution of the allotted tasks keeping the execution time minimum, thus reducing the load. Section 2.1 explains the proposed algorithm, and Sect. 3 shows the case study of the algorithm with Sect. 4 detailing the result obtained through case study. Section 5 gives the conclusion and limitations of the proposed work and also suggests some future work.

# 2 Proposed Methodology

## *2.1 Improvised Bat Algorithm*

a. The Proposed system is working on the new metaheuristic method called Bat Algorithm.
b. The improvisation is also done with the method by utilizing the Min-Min and Max-Min algorithm for generating an enhanced and optimized population of the virtual bats.
c. Min-Min algorithm: It enumerates the minimum finishing time of the tasks and then selects the minimum from them. It then schedules the resource taking minimum execution time, and the available time is added to all other tasks.
d. Max-Min algorithm: It computes the maximum finishing time of the tasks and then selects the minimum from them. It then schedules the resource having minimum execution time, and the available time is added to all.
e. Population generation: It generates the population on the basis of scheduling
f. List keeping the processing time and availability time of each node.
g. By applying the Min-Min and Max-Min, a more preferable population is generated for a better optimized result. Now, pulse frequency (f) for each node (Xi) is initialized; f lies in [0, fmax].

h. Also, pulse rates (r) and loudness (Ai) are also initialized;

   I. 'r' lies in [0,1]: 0—no pulse and 1—max pulse.
   II. 'Ai' lies in A0—1 and Amin—0.

i. Now, the Bat algorithm is applied till it reaches an optimum result. The Bat algorithm works on the property of echolocation and the frequency of the ultrasonic waves which they produce.
j. It is the nature of bat that they compute their target as well as any obstacles in the flying path with the rate of emission and waves produced by them for guidance. They decrease the loudness or frequency of sound produced if they reach their prey and stops temporarily the emission of the waves.

## *2.2 Pseudocode*

Objective function:-f(x)= [x1,x2,……, xd]t
Initialized population Xi; I = 1,2,……,n and velocity Vi
Apply Min-Min and Max-Min algorithm
Reconsider the population for further process
The pulse rate ri and loudness Ai is initialized
Initialize pulse frequency fi for each node Xi
While(t<maximum no. of generation)
Generate the solution by making adjustments in frequency
Update the velocity, location/solution [fromequ–2&4]
if(rand>ri)
opt for a solution from the best solution
muster a local solution from the selected solution
end if
if(rad.<Ai & f(xi)<f(x*)) Acquire
the new solution raiser
and reduce Ai.
End if
Order the bats and acquire the current best x*
End while
Post process result ,visualization
Virtual Bats movement:-
$F_i = f_{min} + (f_{max}-f_{min})*b$ -------2]
$V_i(t) = V_i(t-1) + (X_i(t)–X^*)*fi$-------[3]
$X_i(t) = X_i(t-1) + V_i(t)$--------[4]
b= [0,1]
New solution:-
$X_{new} = X_{old} + eA(t)$
e = [-1,1]
A(t) = <A(t)>----------avg. loudness at time 't'

## 3 Case Study

Phase I: The case study is drawn taking '10' virtual machines and '5' tasks having different makespan on different VMs. The initial population for the algorithm was taken from this matrix (ref. Table 1.), and then, the Min-Min, Max-Min and Alpha-Beta pruning algorithm was applied which gave different result for different conditions of allotment to the VMs.

Phase II: Now, for allotment the optimization algorithms are used. The results thus we get for the allotment of the tasks for Min-Min and Max-Min were different and are shown below:

In Min-Min and Max-Min scheduling technique, makespan means the largest operating time of any machine on which the tasks are requested. In Fig. 1 and Fig. 2, Y-axis denotes the operating times of the machines and X-axis denotes the different machines. For Min-Min—'240' and Max-Min—'230', for '5' tasks/cloudlets allotted to '10' VMs. One technique may outperform the other, and assignments of

**Table 1** Execution times

| | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| VM1 | 160 | 110 | 200 | 110 | 230 |
| VM2 | 180 | 130 | 210 | 190 | 290 |
| VM3 | 200 | 150 | 190 | 120 | 240 |
| VM4 | 210 | 170 | 220 | 180 | 280 |
| VM5 | 250 | 190 | 180 | 130 | 250 |
| VM6 | 280 | 210 | 230 | 170 | 270 |
| VM7 | 170 | 230 | 170 | 140 | 260 |
| VM8 | 150 | 250 | 240 | 160 | 210 |
| VM9 | 220 | 270 | 160 | 150 | 200 |
| VM10 | 240 | 290 | 250 | 100 | 190 |

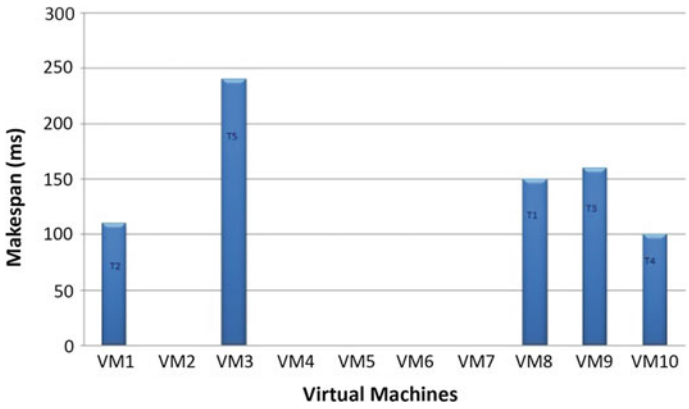


**Fig. 1** Min-Min algorithm assigning tasks

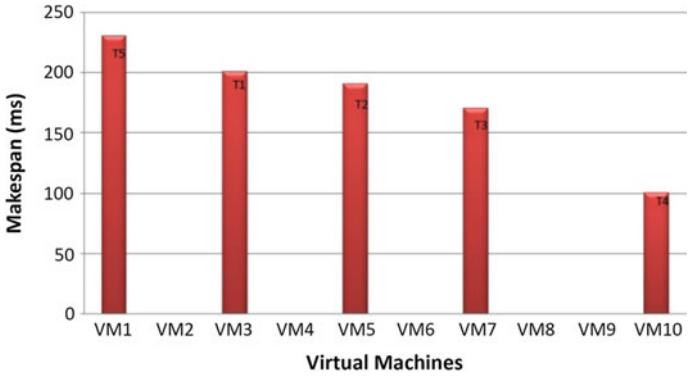


**Fig. 2** Max-Min algorithm assigning tasks

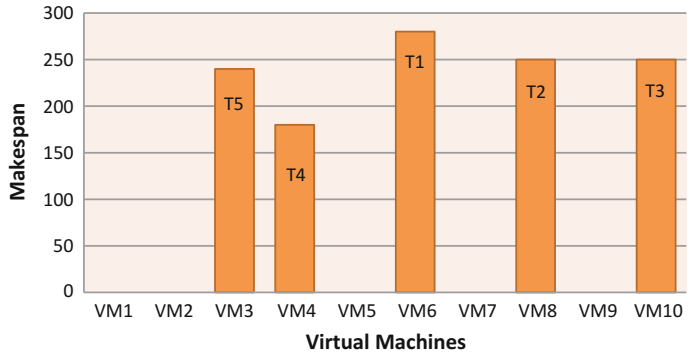


**Fig. 3** Tasks assignment using Alpha-Beta pruning

tasks on VMs may change. Population generated for each method is compared keeping the VMs fixed and varying the tasks.

**Alternatively**, we can do the combined assignment in a more computation reducing method known as Alpha-Beta pruning. In this method, the search is done in a DFS mode and the current best of the node traversed is taken either as alpha (if encountered a maximizing node) and beta (if encountered a minimizing node). This method generates an optimal maximized result. This method is used to do the assignment of the tasks at each stage. In alpha-beta pruning, the maximum makespan calculated was '250' ms (Fig. 3).

Phase III: **Execution of Bat Algorithm for each Allotment**:

In phase III, the Bat algorithm is executed for each of the allotments done by the Min-Min, Max-Min and Alpha-Beta to the VMs. In this phase, the execution times or the processing time of each of the tasks in each allotment is compared. The following tables discuss the processing times, and the comparative studies of the tasks are displayed (Tables 2, 3 and 4).

**Table 2** Bat algorithm with Min-Min allotment

| Iteration 1 | | Iteration 2 | | Iteration 3 | | Iteration 4 | | Iteration 5 | |
|---|---|---|---|---|---|---|---|---|---|
| Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time |
| V1 = 45 | T1 = 45 | V1 = 56.25 | T1 = 101.25 | V1 = 83.43 | T1 = 185.18 | V1 = 139.36 | T1 = 324.05 | V1 = 251.08 | T1 = 575.13 |
| V2 = 33 | T2 = 33 | V2 = 35.25 | T2 = 68.25 | V2 = 41.80 | T2 = 110.05 | V2 = 55.47 | T2 = 165.52 | V2 = 82.97 | T2 = 248.49 |
| V3 = 48 | T3 = 48 | V3 = 61.50 | T3 = 109.50 | V3 = 93.83 | T3 = 203.33 | V3 = 159.97 | T3 = 363.30 | V3 = 292.99 | T3 = 655.84 |
| V4 = 30 | T4 = 30 | V4 = 27.75 | T4 = 57.75 | V4 = 27.99 | T4 = 85.74 | V4 = 27.99 | T4 = 113.73 | V4 = 27.99 | T4 = 141.72 |
| V5 = 72 | T5 = 72 | V5 = 103.50 | T5 = 175.50 | V5 = 177.33 | T5 = 352.83 | V5 = 327.56 | T5 = 680.39 | V5 = 628.55 | T5 = 1308.94 |
| Best | 30 | Best | 57.75 | Best | 85.74 | Best | 113.73 | Best | 141.72 |

**Table 3** Bat algorithm with Max-Min allotment

| Iteration 1 | | Iteration 2 | | Iteration 3 | | Iteration 4 | | Iteration 5 | |
|---|---|---|---|---|---|---|---|---|---|
| Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time |
| V1 = 40 | T1 = 40 | V1 = 49.86 | T1 = 89.86 | V1 = 79.73 | T1 = 175.58 | V1 = 119.66 | T1 = 298.75 | V1 = 229.84 | T1 = 545.12 |
| V2 = 34 | T2 = 34 | V2 = 36.05 | T2 = 70.05 | V2 = 43.47 | T2 = 117.15 | V2 = 59.77 | T2 = 171.31 | V2 = 89.35 | T2 = 261.70 |
| V3 = 46 | T3 = 46 | V3 = 60.05 | T3 = 106.05 | V3 = 90.56 | T3 = 200.03 | V3 = 148.37 | T3 = 352.40 | V3 = 278.68 | T3 = 636.80 |
| V4 = 27 | T4 = 27 | V4 = 24.75 | T4 = 51.75 | V4 = 25.80 | T4 = 77.55 | V4 = 25.80 | T4 = 103.35 | V4 = 25.80 | T4 = 129.15 |
| V5 = 68 | T5 = 68 | V5 = 98.6 | T5 = 166.6 | V5 = 166.66 | T5 = 329.33 | V5 = 302.49 | T5 = 651.47 | V5 = 582.95 | T5 = 1098.90 |
| Best | 27 | Best | 51.75 | Best | 77.55 | Best | 103.35 | Best | 129.15 |

**Table 4** Bat algorithm with Alpha-Beta allotment

| Iteration 1 | | Iteration 2 | | Iteration 3 | | Iteration 4 | | Iteration 5 | |
|---|---|---|---|---|---|---|---|---|---|
| Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time | Velocity | Execution time |
| V1 = 84 | T1 = 84 | V1 = 106.5 | T1 = 190.5 | V1 = 159.78 | T1 = 301.59 | V1 = 235.54 | T1 = 498.32 | V1 = 315.68 | T1 = 602.68 |
| V2 = 75 | T2 = 75 | V2 = 90.75 | T2 = 165.75 | V2 = 175.78 | T2 = 316.54 | V2 = 255.74 | T2 = 418.64 | V2 = 335.32 | T2 = 536.75 |
| V3 = 75 | T3 = 75 | V3 = 90.75 | T3 = 165.75 | V3 = 175.78 | T3 = 316.54 | V3 = 255.74 | T3 = 418.64 | V3 = 335.32 | T3 = 536.75 |
| V4 = 54 | T4 = 54 | V4 = 51.25 | T4 = 105.25 | V4 = 51.25 | T4 = 156.50 | V4 = 51.25 | T4 = 207.75 | V4 = 51.25 | T4 = 248.68 |
| V5 = 72 | T5 = 72 | V5 = 74.25 | T5 = 146.25 | V5 = 82.78 | T5 = 389.53 | V5 = 107.65 | T5 = 502.33 | V5 = 149.36 | T5 = 723.52 |
| Best | 54 | Best | 105.25 | Best | 156.50 | Best | 207.75 | Best | 248.68 |

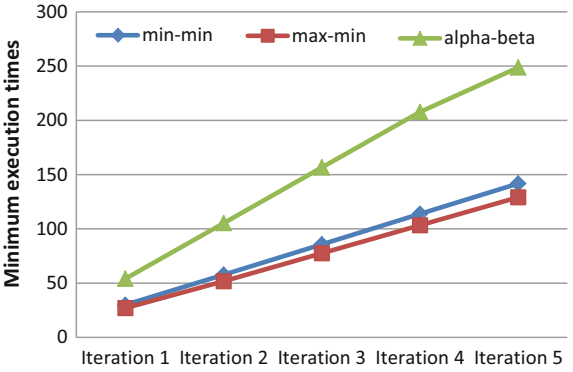


**Fig. 4** Comparative chart of the processing times

The above shows all the processing times of the tasks which are allotted to the VMs with different methods of optimization and with the application of Bat algorithm the variation in their makespans.

## 4 Result Comparison and Discussion

The Bat algorithm is also a heuristic method of optimizing the problems, and it also functions on multiple solution paths. It is a method which is used for generation of an optimal solution.

Figure 4 shows that the red graph line which is Max-Min allotted tasks process is the better one for the execution by the algorithm. The graph chart infers that for the consideration of all the tasks allotment we have to consider the minimum makespan algorithm to be executed by the Bat algorithm which can be seen in the figure.

## 5 Conclusion, Limitations and Future Direction

In this paper, we discussed the allotment of different tasks on the virtual machines, and their execution times are compared. In the result, we observed that the allotment of tasks and minimum execution time is gained from Min-Min and Max-Min algorithm, while in Alpha-Beta pruning method, the tasks execution is optimistically distributed, but the overall execution/processing time is greater than any other. As part of future work, we would also like to explore some benefits from the optimization methods used and other algorithms such as ACO, PSO.

## References


1. Peenaz Pathak, Er. Kamna Mahajan – "A Review on Load Balancing in Cloud Computing"- International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 4 Issue6 June 2015, PageNo.12333–12339
2. Akshat Dhingra, SanchitaPaul - "A Survey of Energy Efficient Data Centres in a Cloud Computing Environment"-International Journal of Advanced Research in Computer and Communication Engineering Vol.2, Issue10, October 2013
3. Kalyani Ghuge, Prof. Minaxi Doorwar – "A Survey of Various Load Balancing Techniques and Enhanced Load Balancing Approach in Cloud Computing" - International Journal of Emerging Technology and Advanced Engineering (ISSN2250-2459, ISO9001:2008 Certified Journal, Volume 4, Issue 10, October 2014)
4. Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang– "Cloud Task scheduling based on Load Balancing Ant Colony Optimization"- 2011Sixth Annual China Grid Conference
5. Mayanka Katyal, Atul Mishra – "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment"- http://www.publishingindia.com
6. Akshat Dhingra and Sanchita Paul– "Green Cloud:Heuristic based BFO Technique to Optimize Resource Allocation"- Indian Journal of Science and Technology, Vol 7(5), 685–691, May 2014
7. Pardeep Kumar, Amandeep Verma – "Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm"- International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 5, May 2012, ISSN:2277128X
8. Nada M. Al Sallami – "Load Balancing in Green Cloud Computation" - Proceedings of the World Congress on Engineering 2013 Vol II, WCE2013, July 3–5, 2013, London, U.K.
9. Velagapudi Sreenivas, Prathap. M, Mohammed Kemae – "Load BalancingTechniques: Major Challenge In Cloud Computing– A Systematic Review"
10. Paulin Florence and V. Shanthi – "A Load Balancing Model Using Firefly Algorithm In Cloud Computing"-Journal of Computer Science 10(7):1156–1165, 2014 ISSN:1549-3636
11. Raja Manish Singh Abhishek Kumar Priyanka Karn, Dr. Sanchita Paul– "Task Scheduling in Cloud Computing using ATM Approach"- International Journal of Engineering Research & Technology (IJERT), ISSN:2278-0181, Vol. 4 Issue 04, April-2015
12. Xin-She Yang– "A New Metaheuristic Bat-Inspired Algorithm"
13. Devipriya, S., and C. Ramesh. "Improved Max-min heuristic model for task scheduling in cloud", 2013 International Conference on Green Computing Communication and Conservation of Energy (ICGCE), 2013
14. Zhan, Zhi-Hui, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, 8 and Yun Li. "Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches", ACM Computing Surveys, 2015