ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter164892020.pdf
https://doi.org/10.30534/ijeter/2020/164892020

A Robust Resource Allocation Framework Using Hybrid Bat Algorithm (HBA)

A. Jemshia Miriam[1], Dr. R. Saminathan[2], Dr. S. Chakaravarthi[3]
[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Tamilnadu, India, jemshiamiriamajs@gmail.com
[2]Associate Professor, Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Tamilnadu, India, samiaucse@yahoo.com
[3]Professor & Head, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamilnadu, India, chakra2603@gmail.com

ABSTRACT

In the era of cloud computing and IT paradigm the resource allocation and over provisioning has introduced lot of automation in terms of manual and automated process. However, most of the cloud resource providers focus more on the effective resource allocation with better pricing. Recent solution adapted for cloud resource allocation and management leads to task level sub optimal performance due to on-demand varying job requirements. To address this issue in the existing cloud resource scheduling and allocation, a novel fine-grained scheduling solution is proposed. The key idea of the proposed solution is to utilize Hybrid Bat algorithm (HBA). Further, the HBA is a modified BAT algorithm. HBA algorithm is privileged to execute in the middleware and tasks allocation subjected to individual physical and virtual machines are completely handled for time critical jobs with fixed and varying deadline. The proposed resource allocation framework has multi-tendency support for all forms of cloud (public, private and hybrid). The entire framework is deployed and validated using Amazon EC2 instance with spark cluster 20-node. The experimental results proven that the proposed HBA based framework is robust and offers a maximum range of high resource utilization.

Key words: Social Bat Algorithm, Bat Algorithm, Cloud resource scheduling

1.INTRODUCTION

Cloud computing refers to the servers that can be accessed online over the internet. Cloud computing in short is delivering of hosted services over the internet like data storage, accessing data [1]. Before Cloud computing, we had many issues like buying lot of servers in order to host a website, maintaining and monitoring a server is always a tough job. Buying more servers is always a costly option as buying more servers means maintaining more servers and for maintains more servers; we need to hire a greater number of people, which is again expensive [2-3]. The traffic is never constant, you might have to buy more number of servers for the varying traffic on the website but it is not always constant, the server is idle most of the time, which means, we are buying more amounts of servers for the traffic, which is idle most of the time [4]. Before cloud, to create an application or a website we needed a team of experts to install, configure, test, run, secure and update them which means we need to hire a team to do that, so individuals were not able to create applications and websites, where as it is a completely different scenario now. It used to be hard for a big company to create and maintain hundreds of applications, there will not be a chance for small and medium companies to build that large amount of applications or websites. With Cloud Computing, even an individual can build an application as there will not be any problem of managing software and hardware as it will be maintained by the service providers. Cloud Computing is on-demand, which means you can have the control over turning resources on or off, which ensures that there is no lack of resources [5-6]. Cloud Computing is cost effective- We only pay for what we use. Cloud Computing is accessible from anywhere, which makes it easier to use data [7].

In this cloud era, the resource management plays a huge role in both the client and service provider site in terms of cost cutting and reduction. The resource management system mechanism helps coordinate IT resources in response to management actions performed by both cloud consumers and cloud providers. Present day heterogeneous resources are located in various geographical locations requiring security-aware resource management to handle security threats. However, existing techniques are unable to protect systems from security attacks. To provide a secure cloud service, a security-based resource management technique is required that manages cloud resources automatically and delivers secure cloud services [8]. Core to this system is the Virtual Infrastructure Manager (VIM) that coordinates the server hardware so that virtual server instances can be created from the most expedient underlying physical server. To address this, this research work addresses a secured framework, which involves context aware resource allocation [9].

Contributions in this paper:

- A novel algorithm is proposed by modifying the existing state of the art BAT Algorithm.
- The proposed Social Bat Algorithm is modified and the experimentation is proven that the proposed HBA is better than the particle swarm optimization and parallel genetic algorithm.
- HBA uses an optimal best solution obtained when processing the local solutions. Further, the proposed framework is tested in all real time networks and the results confirm that the proposed HBA is optimal.
- The proposed HBA utilizes the effective handling of optimization with best solutions, which avoids overfitting and under fitting problems.
- The proposed HBA is a hybrid bat algorithm and completely deployed and validated in Hadoop Cluster.

The organization of this paper is as follows: Section II deals with the literature survey. Section III proposes the cloud resource management framework. This section explains the HBA algorithm. Section IV deals with the experimental setup. The penultimate section deals with the result and performance analysis. Finally, the paper is concluded in the Section VI.

2.LITERATURE SURVEY

From the literature survey, it is observed that enormous amount of algorithms for cloud resource scheduling was developed in recent days [13][14][15]. Out of those algorithms, evolutionary algorithm based solutions found to give promising results for cloud scheduling. In this section, a detailed survey is given based on the evolutionary algorithms.

Bat algorithm is one of the best evolutionary algorithms in the decade. The algorithm works based on the principle of echo localization. This echo localization property is used by BATs for hunting and obstacle finding. The other common evolutionary algorithm widely used for resource scheduling is particle swarm optimization. Maria Alejandra Rodriguez and RajkumarBuyya [1] proposed an approach on deadline based resource provisioning and scheduling. The key idea of their work is to utilize particle swarm optimization technique for efficient resource provisioning. They validated their approach in Cloudsim and presented an outstanding result.

Qi Zhang et al presented a novel model called prism, which is a Fine-Grained Resource-Aware Scheduling scheme for MapReduce [2]. In their work, the authors validated various case scenarios in which tasks with high varying resource requirements were perfectly handled and achieved betterment in execution of such process in terms of scheduling, computation and fast processing. To achieve this they introduced a novel Fine-grained resource aware scheduling scheme, which divides the tasks into multiple phases, and each phases are then handled appropriately. To the point, each fine-segregated phase holds the information such as resource usage profile, which will help the admin to know the total resource utilization and consumption [10].

In addition to the above contribution, Xiaomin Zhu proposed Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds. The key of their work is to perform guaranteed system schedulability. They proposed task-oriented energy consumption model. From the literature survey, it is observed that the existing methodologies for cloud resource provisioning is not efficient and requires greater computation [11-12]. Most of the algorithmic cases reported the efficiency and execution time of the algorithm. Hence, there is a pressing need for an efficient algorithm for cloud resource allocation and scheduling. To address this issue, this paper proposed a novel framework utilizing the Social Bat Algorithm, which is a hybrid bat algorithm for cloud resource allocation. The novelty of the proposed Social Bat Algorithm is that the algorithm utilizes the effective handling of optimization with best solutions, which avoids overfitting and under fitting problems. The proposed algorithm always sticks to the best optimal solution which avoids overfitting and under fitting problems.

3.PROPOSED FRAMEWORK

The proposed framework for cloud resource management as shown in the Fig. 1 is implemented using Hybrid Bat Algorithm (HBA). HBA is a novel and hybrid algorithm, which relies on BAT algorithm. HBA also work based on the echo localization, a technique used in BAT algorithm. The key idea is to use sonar waves to detect food and obstacles with the boundary range. The motto of this work is to utilize this hybrid BAT algorithm for efficient resource scheduling. The rule set governed for HBA is stated as follows.

There is no path restriction for bats to fly, where the velocity is defined by V_i at position X_i.

The frequency is fixed in order to communicate within the population whereas the wavelength can be adjusted automatically based on the target locality.

The loudness with the echo can be varied from the higher pitch to the lower pitch. The detailed illustration of the parameters considered for the algorithm is stated in the section 'Algorithm' and the pictorial representation is given in the Fig. 2.

The proposed HBA is designed for optimizing the resource allocation solutions in order to avoid resource wastage and to increase the cost of income.

The proposed HBA is applied for resource scheduling for efficient optimization in resource allocation.

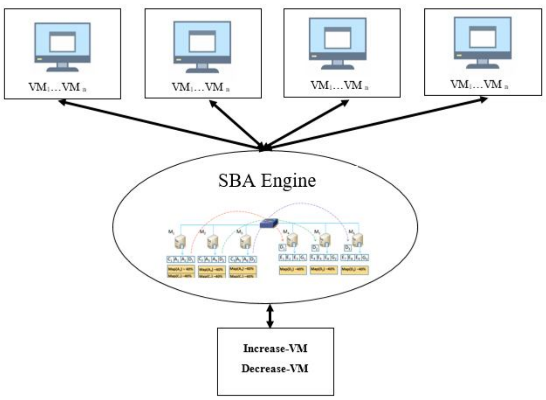

Figure 1: Architecture of the Proposed Framework

Fig. 1 shows the architecture of the proposed framework. The total population i.e., the bats utilizes echo localization technique to find the distance between food and obstacles.

Algorithm:

Algorithmic part of this section contains two main components namely i) features ii) algorithm

Features considered for experimentation are listed as follows

- Population space or dimension
- Population generation
- Loudness
- Pulse rate
- Frequency min
- Frequency max
- Lower bound
- Upper bound

Algorithm

Fig. 2 shows the algorithmic part of the proposed HBA.

```
Begin move_bat():
    S = [[0.0 for i in range(.D)] for j in range(NP)]
    init_bat()
    for t in range(N_Gen):
            for i in range(NP):
                rnd = np.random.uniform(0, 1)
                Q[i] = Qmin + (Qmax - Qmin) * rnd
                for j in range(D):
                    v[i][j] = v[i][j] + (Sol[i][j] - best[j]) * Q[i]
                    S[i][j] = Sol[i][j] + v[i][j]
                    S[i][j] = simplebounds(S[i][j], Lb[j],Ub[j])
                rnd = np.random.random_sample()
                if rnd > r:
                    for j in range(D):
                        S[i][j] = best[j] + 0.001 * random.gauss(0, 1)
                        S[i][j] = simplebounds(S[i][j], Lb[j],Ub[j])
                Fnew = Fun(D, S[i])
                rnd = np.random.random_sample()
                if (Fnew <= Fitness[i]) and (rnd < A):
                    for j in range(D):
                        Sol[i][j] = S[i][j]
                    Fitness[i] = Fnew
                if Fnew <= f_min:
                    for j in range(D):
                        best[j] = S[i][j]
                    f_min = Fnew
        return (f_min)
End
```

Figure 2: Algorithm of Social BAT Movement

4.EXPERIMENTAL SETUP

The proposed framework is experimented and validated in the simulated test bed. The test bed is executed in a machine with 64 GB RAM, Intel Xeon processor running in Ubuntu Server Operating system. The entire simulation is made using python and the simulation executed for more than 20 hours. The entire results were plotted in the three dimensional plot with the considered parameters such as resource wastage, file unavailability and power consumption. The results were obtained for the allocation in VM in the individual physical machine. Table 1 - Table 4 shows the resource available plot and allocation-scheduling plot. Multi VMs are allowed to run inside physical machine. Each VM running in the PM has several VMs execution on it. In addition to the above setup, we configured with different variants of VMs. VM pricing is also estimated in order to estimate the total cost incurred during the allocation. This will help us to estimate the loss incurred by the cloud service provider during the idle wait happening while allocation and scheduling.

The workflow of the propose framework is also validated based on the processing time of different VMs. Each VMs configuration is listed in the Tables 1 – 4. Further, the validation is calculated based on the fixed deadline for individual physical machine. The deadline period is restricted not to get relaxed in order to estimate the exact loss incurred during the allocation or idle wait during scheduling.

Table 1: Resource Availability on Physical Machine 1

Physical Machine	VM1	VM2
PM1	VM(25)	VM(25)
	VM(50)	VM(50)
	VM(75)	VM(75)

Table 2: Resource Availability on Physical Machine 2

Physical Machine	VM1	VM2	VM3
PM2	VM(100)	VM(100)	VM(100)
	VM(125)	VM(125)	VM(125)
	VM(150)	VM(150)	VM(150)

Table 3: Resource Availability on Physical Machine 3

Physical Machine	VM1	VM2	VM3
PM3	VM(150)	VM(150)	VM(150)
	VM(175)	VM(175)	VM(175)
	VM(200)	VM(200)	VM(200)

Table 4: Resource Availability on Physical Machine 4

Physical Machine	VM1	VM2	VM3
PM4	VM(200)	VM(200)	VM(200)
	VM(225)	VM(225)	VM(225)
	VM(250)	VM(250)	VM(250)

5.RESULT ANALYSIS

The experimental results of the proposed framework are given in the tables from Table 5 to Table 14. The effectiveness of the proposed HBA is validated with the various benchmarks.

The benchmarked strategies used in the HBA are levy, sphere, rosenbrock, schwefel, rastrigin, griewank, ridge, salomon, whitley, ackley. Fig. 3 shows the results of benchmarked strategies for BA and HBA for the min parameter. Fig. 4 shows the results of benchmarked strategies for BA and HBA for the max parameter.

Fig. 5 shows the results of benchmarked strategies for BA and HBA for the mean parameter. Fig. 6 shows the results of benchmarked strategies for BA and HBA for the median parameter.

Fig. 7 shows the results of benchmarked strategies for BA and HBA for the standard deviation parameter.

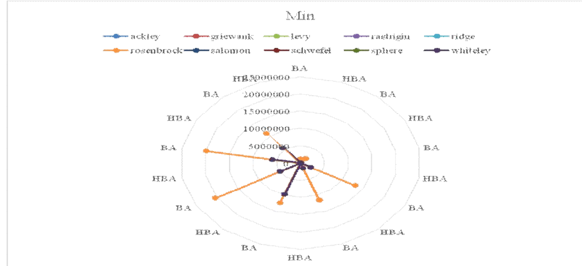

Figure 3: BA vs HBA Results – Min Parameter

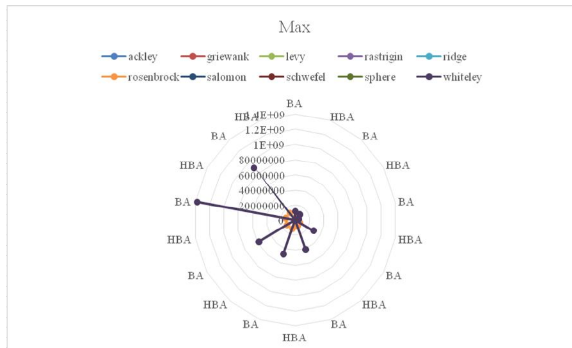

Figure 4: BA vs HBA Results – Max Parameter

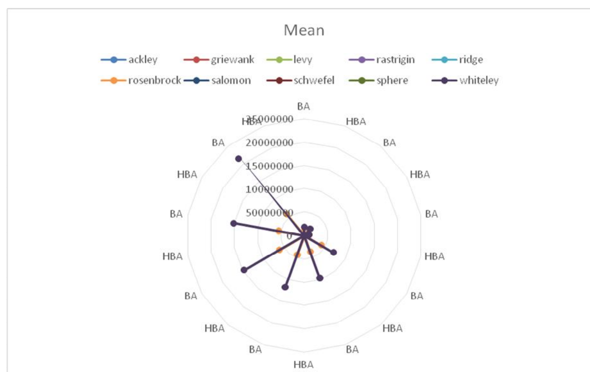

Figure 5: BA vs HBA Results – Mean Parameter

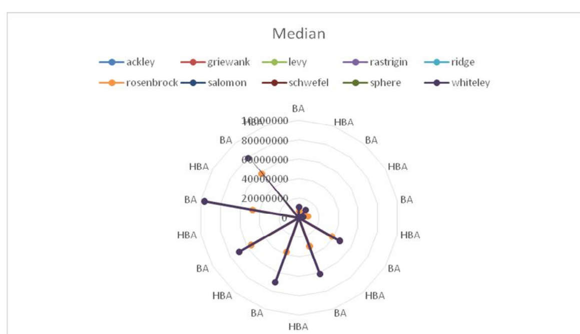

Figure 6: BA vs HBA Results – Median Parameter

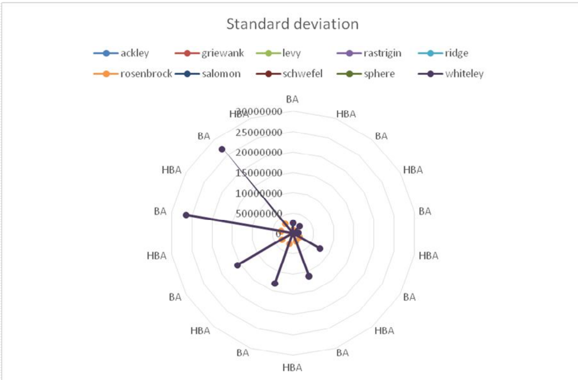

Figure 7: BA vs HBA Results – Standard Deviation Parameter

Table 5: Statistical based Comparative Analysis of Ackley Benchmark Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	ackley	D=10 Np=20	12.95681395	19.03018126	16.43930662	16.74286	1.570757
HBA	ackley	D=10 Np=20	7.42338E-07	2.814349981	0.65320626	2.31E-06	0.819045
BA	ackley	D=10 Np=30	13.41228052	18.24848154	16.47125976	17.04906	1.340698
HBA	ackley	D=10 Np=30	4.53012E-07	3.574239616	1.083724707	1.155149	1.121569
BA	ackley	D=10 Np=50	14.04885709	19.50722109	16.87788528	17.0948	1.128057
HBA	ackley	D=10 Np=50	5.94777E-07	2.814349981	1.032712165	1.155149	0.975228
BA	ackley	D=20 Np=20	16.30478118	19.4487542	17.90237446	17.85586	0.688578
HBA	ackley	D=20 Np=20	1.839993476	6.31580377	3.604944489	3.489881	1.276403
BA	ackley	D=20 Np=30	17.214451	19.51799151	17.98783775	17.86543	0.58904
HBA	ackley	D=20 Np=30	1.155148503	7.294016103	2.985236878	2.922803	1.334953
BA	ackley	D=20 Np=50	15.87159318	19.46855449	17.98837529	17.98534	0.724352
HBA	ackley	D=20 Np=50	1.646223633	5.977244539	3.158546452	2.920986	1.197545

Table 6: Statistical based Comparative Analysis of Griewank Benchmark Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	griewank	D=10 Np=20	1.543100219	4.571646101	2.995083965	2.821429	0.732493
HBA	griewank	D=10 Np=20	0.032013428	0.437065819	0.13124492	0.110651	0.085948
BA	griewank	D=10 Np=30	1.852639427	4.95855177	3.21632097	3.248924	0.803675
HBA	griewank	D=10 Np=30	0.012320989	0.757460717	0.15550507	0.125512	0.147123
BA	griewank	D=20 Np=20	4.495300461	9.5402401	7.171027262	7.125763	1.43381
HBA	griewank	D=20 Np=20	2.22045E-16	0.098055013	0.025859073	0.02459	0.020506
BA	griewank	D=20 Np=30	2.997100035	10.04286618	6.30077734	5.838527	1.781593
HBA	griewank	D=20 Np=30	7.21645E-15	0.120074052	0.032317282	0.029481	0.029357

Table 7: Statistical based Comparative Analysis of Levy Benchmark Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	levy	D=10 Np=20	6.68271E-07	1.032367083	0.347613475	0.090845	0.426536
HBA	levy	D=10 Np=20	8.6269E-17	0.850676406	0.059463531	1.87E-15	0.173929
BA	levy	D=10 Np=30	4.79581E-07	0.991519002	0.162691623	4.02E-06	0.333619
HBA	levy	D=10 Np=30	8.25863E-17	0.181689108	0.039971604	6.32E-15	0.052971
BA	levy	D=20 Np=20	2.34072E-06	2.872842893	0.968379018	0.893494	0.909044
HBA	levy	D=20 Np=20	7.2699E-19	1.792197365	0.350836641	0.090845	0.491524
BA	levy	D=20 Np=30	3.29245E-06	1.927657309	0.563721111	0.272573	0.621769
HBA	levy	D=20 Np=30	3.96103E-18	1.701352811	0.211094634	0.181689	0.387035

Table 8: Statistical based Comparative Analysis of Rastrigin Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	rastrigin	D=10 Np=20	13.9298977	101.485199	44.01707144	36.81367	21.81012
HBA	rastrigin	D=10 Np=20	3.979836434	22.88399826	11.82233731	11.9395	4.781466
BA	rastrigin	D=10 Np=30	12.93462382	69.64698799	38.12693548	36.81358	16.53619
BA	rastrigin	D=20 Np=20	36.81490846	142.2796348	88.67176142	93.5275	28.43502
HBA	rastrigin	D=20 Np=20	21.8890791	69.64681497	46.44457022	41.78822	12.78918
BA	rastrigin	D=20 Np=30	41.78951117	166.1586778	88.87070429	74.62325	36.19036

Table 9: Statistical based Comparative Analysis of Ridge Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	ridge	D=10 Np=20	1207.791395	7924.437666	4134.581048	3928.559	1812.34
HBA	ridge	D=10 Np=20	2.93919E-07	6.89773E-05	7.30567E-06	1.98E-06	1.4E-05
BA	ridge	D=10 Np=30	2013.497515	8748.925565	4767.739665	4596.844	1607.933
BA	ridge	D=20 Np=20	6078.217716	27218.41793	14134.79567	13916.41	4572.348
HBA	ridge	D=20 Np=20	0.002534585	0.067033432	0.020681288	0.013499	0.01811
BA	ridge	D=20 Np=30	3516.182348	25494.74263	12474.47509	11510.14	4908.982

Table 10: Statistical based Comparative Analysis of Rosenbrock Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	rosenbrock	D=10 Np=20	1179657.884	33527106.07	9724989.174	7050440	8181975
HBA	rosenbrock	D=10 Np=20	0.008163632	71.15021665	5.607646668	2.479962	13.78989
BA	rosenbrock	D=10 Np=30	1656245.723	27527151.43	9745564.166	6682500	7823574
BA	rosenbrock	D=20 Np=20	13087223	77426325.36	40923505.07	38091583	18213488
HBA	rosenbrock	D=20 Np=20	0.017135306	111.2791496	15.9941995	10.4267	23.99042
BA	rosenbrock	D=20 Np=30	11539319.1	89858820.08	37260634.7	30673234	20346744

Table 11: Statistical based Comparative Analysis of Salomon Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	salomon	D=10 Np=20	380.9147749	1452.423593	927.8252177	909.0687	260.549
HBA	salomon	D=10 Np=20	0.099495906	1.591924382	0.4457406	0.397983	0.326369
BA	salomon	D=10 Np=30	377.0942963	1810.693347	904.2148048	961.2369	370.6271
BA	salomon	D=20 Np=20	1126.797068	3671.3748	2180.988738	2077.56	556.607
HBA	salomon	D=20 Np=20	1.591924379	12.03837243	4.652300226	3.581799	2.954069
BA	salomon	D=20 Np=30	1392.05862	3375.818632	2273.522364	2247.024	503.3671

Table 12: Statistical based Comparative Analysis of Schwefel Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	schwefel	D=10 Np=20	1959.602247	3289.229564	2725.59624	2727.627	279.3531
HBA	schwefel	D=10 Np=20	713.664254	1761.422227	1137.196604	1129.726	302.4351
BA	schwefel	D=10 Np=30	2256.802075	3108.460391	2685.240477	2684.772	208.3854
BA	schwefel	D=20 Np=20	5465.207747	6767.540433	6186.573044	6163.477	298.6477
HBA	schwefel	D=20 Np=20	1764.423571	3815.612049	2839.429654	2916.921	527.6105
BA	schwefel	D=20 Np=30	5414.22199	6736.635909	6233.538811	6275.883	311.5135

Table 13: Statistical based Comparative Analysis of Sphere Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	sphere	D=10 Np=20	2.835117732	21.16130085	10.67777626	8.936861	5.673571
HBA	sphere	D=10 Np=20	2.82199E-15	9.43268E-14	2.20982E-14	9.85E-15	2.22E-14
BA	sphere	D=10 Np=30	1.728915327	19.1461846	9.608488422	9.373496	4.9299
BA	sphere	D=20 Np=20	4.70684935	39.20025809	15.64143313	13.00739	9.109928
HBA	sphere	D=20 Np=20	1.32526E-17	9.20161E-15	1.45096E-15	2.92E-16	2.56E-15
BA	sphere	D=20 Np=30	1.420018171	42.61122137	17.32465154	15.06807	10.64779

Table 14: Statistical based Comparative Analysis of Whitley Function

Alg.	Bench.	Meas.	Min	Max	Mean	Median	Std
BA	whitley	D=10 Np=20	93390.94528	114047414.7	18348589.16	11068826	25972973
HBA	whitley	D=10 Np=20	18.07292823	60.03289276	41.13330325	43.39945	10.67432
BA	whitley	D=10 Np=30	2567.358656	93314625.34	19248524.17	10458668	24097765
BA	whitley	D=20 Np=20	2461923.754	290671465.9	71254935.68	47193820	74979817
HBA	whitley	D=20 Np=20	150.2285584	350.5333379	243.5080122	245.4359	40.69101
BA	whitley	D=20 Np=30	1591392.813	417382556.6	96891183.93	61383259	1.12E+08

Further, the Table 5 – 14 presents the exclusive statistical analysis of benchmark function used for optimization. All the benchmark functions are tested and validated with the same set of machines as displayed in the Table 1 – 4. During the experimentation the statistical parameters such as min, max, mean, standard deviation and median are calculated.

6.CONCLUSION

This paper is concluded by proposing a novel cloud resource management using Social Bat Algorithm. The experimental results shows that the demand in the resource leads to increased amount of traffic overload. To address this issue, various optimization algorithms are proposed. The purpose of optimization algorithm is to find the best and optimal solution out of the other solutions. In this regard, Particle Swarm Optimization is considerable producing promising results as of now. However, the problem with the existing PSO based algorithms suffers due to convergence problem and it is not suitable for multi-objectives. To address this issue, this paper proposed a novel hybrid algorithm called Social Bat Algorithm.

The HBA used in this paper considers the resource allocation and achieves multi-objective functionality without any convergence. The proposed HBA solved the major problem of resource provisioning and cost reduction. The proposed HBA also proves that the cost estimation for the HBA is comparatively less than the existing state of the art models. Finally, when compared with the other work such as parallel genetic algorithm, NSGA algorithms, the proposed HBA is best in terms of processing, execution etc.

REFERENCES

[1] M. A. Rodriguez and R. Buyya, "**Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds**," In IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222-235, 1 April-June 2014.

[2] Q. Zhang, M. F. Zhani, Y. Yang, R. Boutaba and B. Wong, "**PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce**," In IEEE Transactions on Cloud Computing, vol. 3, no. 2, pp. 182-194, 1 April-June 2015.

[3] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin and X. Liu, "**Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds**," In IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 168-180, April-June 2014.

[4] T. Tran, B. Vo, T. T. N. Le, N. T. Nguyen, "**Text clustering using frequent weighted utility itemsets**", Cybern. Syst., vol. 48, no. 3, pp. 193-209, Mar. 2017.

[5] M. Quadrana, A. Bifet, R. Gavaldà, "**An efficient closed frequent itemset miner for the MOA stream mining system**", AI Commun., vol. 28, no. 1, pp. 143-158, Jan. 2015.

[6] H. Huang, "**Fusion of Modified Bat Algorithm Soft Computing and Dynamic Model Hard Computing to Online Self-Adaptive Fuzzy Control of Autonomous Mobile Robots**", In IEEE Transactions on Industrial Informatics, vol. 12, no. 3, pp. 972-979, June 2016.

[7] W. Dong, M. Zhou, "**Gaussian classifier-based evolutionary strategy for multimodal optimization**", IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 6, pp. 1200-1216, Jun. 2014.

[8] R. Boutaba, L. Cheng, Q. Zhang, "**On cloud computational models and the heterogeneity challenge**", J. Internet Serv. Appl., pp. 1-10, 2012.

[9] D. Xie, N. Ding, Y. Hu, R. Kompella, "**The only constant is change: Incorporating time-varying network reservations in data centers**", Proc. ACM SIGCOMM, pp. 199-210, 2012.

[10] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, K. Vahi, "**Characterizing and profiling scientific workflows**", Future Generation Comput. Syst., vol. 29, no. 3, pp. 682-692, 2012.

[11] S. Pandey, L. Wu, S. M. Guru, R. Buyya, "**A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments**", Proc. IEEE Int. Conf. Adv. Inform. Netw. Appl., pp. 400-407, 2010.

[12] D. Sharma, A. Chatterjee, A. Rakshit, "**A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization**", IEEE Trans. Fuzzy Syst., vol. 17, no. 2, pp. 329-342, Apr. 2009.

[13] Chandra Sekhar Reddy, N. Purna Chandra Rao Vemuri, Govardhan, A. **An Emperical Study on Support Vector Machines for Intrusion Detection,** *International Journal of Emerging Trends in Engineering Research,* Vol. 7, No. 10, pp. 383-387, October 2019.

[14] Rufo I. Marasigan Jr, Alvin Sarraga Alon, Mon Arjay F. Malbog, Joshua S. Gulmatico, **Copra Meat Classification using Convolutional Neural Network,** *International Journal of Emerging Trends in Engineering Research,* Vol. 8. No. 2, February 2020

[15] Subramaniyam, S, **Performance Analysis on Diesel Engine using Neem and Soya Bean Oil,** *International Journal of Emerging Technologies in Engineering Research (IJETER),* Vol. 5, Issue 8, August 2017.