A Novel Satin Bowerbird Optimizer based Energy Aware Task Scheduling with Fault Tolerance Model for Green Cloud Computing

[1] M. Getzi [2] Dr. K. Chitra

[1] Research ScholaR,Department of Computer Science, Madurai Kamaraj University, Madurai, Tamil Nadu, India.
[2] Assistant Professor,Department of Computer Science,Government Arts College Melur, Madurai District, Tamil Nadu, India
[1]getzimoses@gmail.com, [2]manikandan.chitra@gmail.com

Abstract

Green cloud computing (GCC) becomes an emergent computing paradigm which intends to handle the energy consumption performance in the cloud data centers. But the minimization of energy usage might delay the service response time that greatly affect the efficiency of the service. Therefore, it becomes essential to schedule the tasks effectively by finding the optimal tradeoff among energy usage and performance. At the same time, the task scheduling process can be considered as an NP hard problem, which can be addressed by utilization of metaheuristic algorithms. With this motivation, this article introduces a novel satin bowerbird optimizer based energy aware task scheduling with fault tolerance (SBO-EATSFT) model in GCC environment. The intention of the SBO-EATSFT technique is to allocate the tasks to resources in an energy efficient way. Besides, the SBO-EATSFT technique derives an objective function with the aim of the minimization of energy consumption and bandwidth. In addition, the SBO-EATSFT technique involves a fault tolerant scheme and thereby improves the overall efficacy of the GCC environment. The design of SBO algorithm for optimal task scheduling process in GCC environment shows the novelty of the work. To make sure the enhanced scheduling performance of the SBO-EATSFT technique, a series of simulations were performed under varying numbers of tasks and the comparative results analysis highlighted the better performance of the SBO-EATSFT technique over the recent approaches.

***Keywords** - Energy consumption, Green cloud computing, Makespan, Metaheuristics, NP hard problem, Objective function.*

1. Introduction

Cloud computing (CC) is a computing technology for cooperative computing, that is powered by high-bandwidth networks, large data centres, distributed computing resources, and other large-scale storage [1, 2]. Hence, there are a massive amount of servers which have efficiently controlled in data centres. Green cloud computing is evolving as an innovative computing model which focuses on managing energy utilization efficacy in cloud data centres. For saving cost as well as reducing carbon emissions, the operator of GCC has given greater consideration to energy utilization efficacy [3]. Energy utilization has become an important concern for cloud service providers because of environmental impact and operating costs [4]. There have been few researches dedicated to the energy-effective technologies that are applicable for large data centres.

The availability of large infrastructure and higher computing power is needed which provide distinct stages of quality of service (QoS) for the CC and grid environment [5]. There are several approaches to solve the problems of scheduling; usually scheduling systems are determined as heuristic workflow scheduling, hybrid metaheuristic, meta heuristic-based scheduling, heuristic scheduling, workflow and task scheduling [6,7]. Metaheuristic scheduling systems could take advantage of Particle Swarm Optimization (PSO),Ant Colony Optimization (ACO) and Stimulated Annealing (SA) approaches for generating optimal scheduling. For metaheuristic scheduling workflow systems, the intended objective is frequently originated in the fitness function. For these reasons, when various objections are executed for scheduling, several co-efficient is assigned to all the goals which could change the effects of all the goals on the entire workflow scheduling.

Similarly, metaheuristic algorithm is widely employed for solving different optimization problems, for example, feature selection [8]. Over the past two decades, the metaheuristic optimization algorithm has gained more popularity. Usually, they are stimulated by animals’ behaviour and physical phenomena. In addition, combining, learning and using this algorithm is easily performed. The only fact in utilizing this algorithm is how to provide input and attain output from the scheme. This algorithm acts better than traditional optimization approaches compared to local optimization provided their stochastic nature. On the other hand, each metaheuristic method has certain shortcomings [9]. Abualigah and Diabat [10] proposed a hybrid antlion optimization method using elite-based differential evolution to solve multiobjective task scheduling challenges from CC environment. In the presented model, the multiobjective nature of problem originates in the requirement for minimalizing makespan when increasing resource consumption.

Yuan et al. [11] presented a spatial task scheduling and resource optimization (STSRO) methodology for minimizing the overall cost by cost-effective scheduling each incoming task of heterogeneous application for meeting task delay-bound constraints. In every time slot, the cost reduction problem for DGCDC is expressed as a limited optimized one and resolved by presented SA-based bat algorithm (SBA). Mohammadzadeh et al. [12] developed IGWO, a better form of the GWO model that employs the hill-climbing technique and chaos method for attaining effective outcomes. Next, a binary version of the presented IGWO method, utilizing different S and V functions, is presented to handle the task scheduling problems in CC data centres aims to minimalize the power consumption, execution cost, and makespan.

Ari et al. [13] designed a biologically inspired scheduling system, that is a depending on an altered form of the ACO method which focuses on minimizing the makespan time when guaranteeing load balancing amongst resources for enabling GCC. Experiments of the presented method in different scenarios have been carried out for elaborating the impacts of presented method. In [14], the Dynamic Voltage Frequency Scaling system was integrated into the optimization process and a collection of non-domination solutions are attained by NSGA-II. Furthermore, the ANN method that is an effective ML method, is utilized for predicting the virtual machine based on the features of the resource and task. The optimal solution attained by the optimization method with and without the help of ANN is discussed and presented.

This article presents an efficient satin bowerbird optimizer based energy aware task scheduling with fault tolerance (SBO-EATSFT) technique for in GCC environment. The SBO-EATSFT technique aims to assign the tasks to the resources with maximum energy efficiency and minimum makespan. Moreover, the SBO-EATSFT technique derives an effective fault tolerant scheme and thereby enhances the overall performance of the GCC environment. The design of SBO algorithm for optimal task scheduling process from GCC environment shows the novelty of the work. For inspecting the better scheduling outcomes of the SBO-EATSFT technique, a wide range of experiments were carried out for different number of tasks.

2. The Proposed Model

The GCC service provider contributes to offering resources to execute the user jobs with the aim of increasing resource utilization and balancing load. Figure 1 illustrates the overall process of SBO-EATSFT technique. For finding the optimal solution to attain proper tradeoff, the proposed model is designed to fulfill the requirements of GCC. Consider the data centers, which are generally composed of several resource sites dispersed in varying geographical places in GCC. Practically, every sub task $t_i(i = 1, 2, \ldots, m)$ is provided by a resource region R_j for satisfying the required criteria where n denotes the resource count fulfilling the sub task t_i. Let the GCC be $GCloud = (D, T, P, G), D = \{D_1, D_2, \ldots, D_d\}$ indicates set of d data centers, T indicates the correspondence matrix among the arbitrary task and computing node, T_{ij} signifies the task t_i is accomplished on node j. P_i refers the power of node i in idle situations, and G_i implies the peak power of node i. In this study, the tasks are allocated to resources by the use of SBO algorithm with two parameters namely energy and makespan. For reducing the energy utilization with the constraints of cost, cost restricted energy optimized was determining the energy utilization cost.

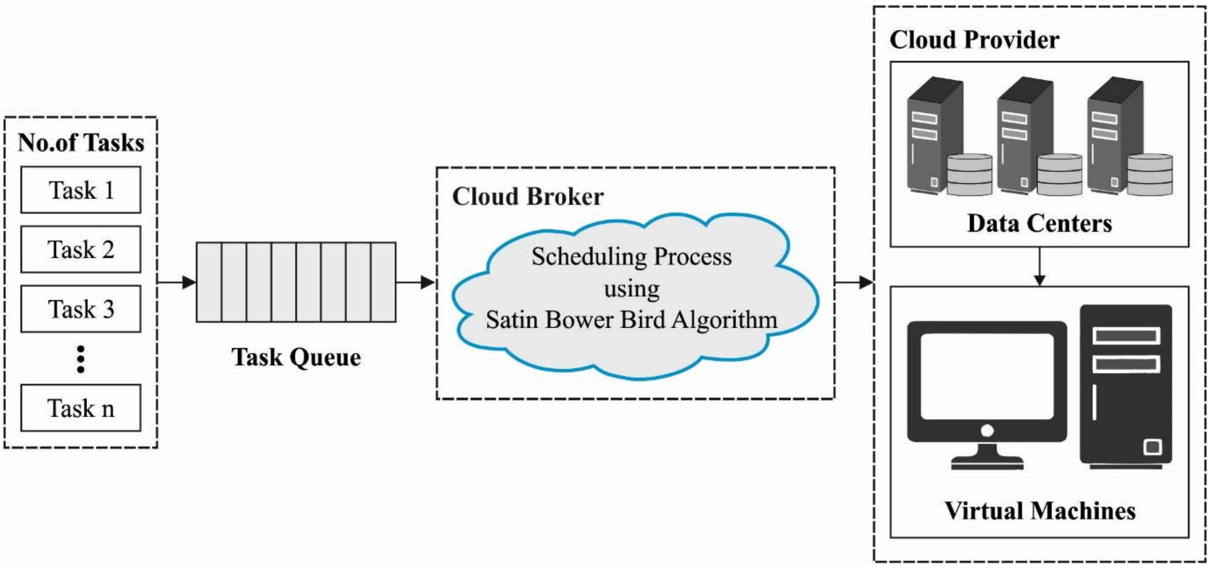

Figure 1. Working Process of SBO-EATSFT Technique

The proposed model involves the design of SBO technique that is dependent upon the performance of satin bower-birds, the male ones are required to construct nests for attracting females and offspring reproduction [15]. The place of the nest identifies the attractive level of the female ones. During the nesting procedure, the male damages the nest constructed by other males. Every male intends to learn from the optimal nest. The number of separate nests gets arbitrarily created among the maximum as well as minimum limits. The dimensionality variable D indicates the identical parameter count needed to resolve the optimization problem. The nest constructed by males has an attractive probability to determine the attractiveness of the females. When the probability value gets increases, it becomes easier for nest to attract the females and it can be defined using Eq. (1)

$$prob_i = \frac{fit_i}{\sum_{i=1}^{NB} fit_i} \tag{1}$$

where fit_i implies the fitness of the ith solution and it can be attained by the use of Eq. (2):

$$fit_i = \begin{cases} \frac{1}{1+f(x_i)} & f(x_i) \geq 0 \\ 1 + |f(x_i)| & f(x_i) < 0 \end{cases} \tag{2}$$

where $f(X)$ implies the cost function value of the ith location which needs to be properly tuned. At every round, the recent modifications at any of the bowers are determined based on Eq. (3).

$$x_{ik}^{new} = x_{ik}^{old} + \lambda_k \left(\left(\frac{x_{jk} + x_{elite,k}}{2} \right) - x_{ik}^{old} \right) \tag{3}$$

where x_i represents the ith bower and x_{ik} indicates the kth member of the vector. x_i can be defined as the target solution amongst every solution in the present round. The variable λ_k (number of steps) computes the attractive nature of the target bower and it can be defined as follows.

$$\lambda_k = \frac{\alpha}{1 + p_j} \tag{4}$$

where α represents the maximum step size and p_j implies the probability attained using Eq. (1) by means of the target bower. During the mutation process, arbitrary variations are employed to x_{ik} with a particular probability. To mutation procedure, a uniform distribution (N) is utilized with a mean of x_{ik}^{old} and variance of σ^2:

$$x_{ik}^{new} = N\left(x_{ik}^{old}, \sigma^2\right) \tag{5}$$

For better comprehension, the probability formulation can be converted as follows.

$$N\left(x_{ik}^{old}, \sigma^2\right) = x_{ik}^{old} + \left(\sigma * N(0,1)\right) \tag{6}$$

where σ indicates a proportion of the space width which can be determined using Eq. (7):

$$\sigma = z * (var_{\max} - var_{\min}) \quad (7)$$

where $var_{\max}$ and $var_{\min}$ are the maximum and minimum limits allocated to the parameters correspondingly. The variable z indicates the percentage difference among the maximum and minimum limits.

In this study, an objective function is derived by the use of energy utilization and makespan C_{max}or makespan and E_{flow} as the energy flow, commonly employed for determining by the end resource whether which one is essential. Therefore, the task scheduling process of the proposed model can be considered as the optimization problem of the minimization of the objective function, as given below.

$$Min\left(\alpha\left(\frac{C_{max}}{c_{sf}}\right) + (1-\alpha)\left(\frac{E_{flow}}{E_{sf}}\right)\right) \quad (8)$$

The objective function includes two major elements namely $E_{flow}(S)$ and $C_{max}(S)$, where $E_{flow}(S)$ is the energy sent S and $C_{max}(S)$ indicates the makespan, involves the highest time taken to the final task for leaving the system. In this study, it is considered that one fault can be tolerated at any time distance. When the host gets failed, every VM and task on the host also fails. Under fault free case, a backup copy t_i^B is deallocated once the primary copy t_i^P gets finished. The overall process of SBO-EATSFT technique is given in Algorithm 1.

Algorithm 1: Pseudocode of SBO-EATSFT technique

Input: Number of Tasks, Available Resources
Output: Scheduling the task in resources
Start Procedure
Allocate Number of Virtual Machine and its Parameters
For Each Task in Virtual Machine
    Calculate the Execution Time on Virtual Machine
        Initializing the primar population of bowers arbitrarily
        Compute the cost of bowers
        Define the optimum bower and consider as elite
        While the final condiiton is not fulfilled
            Compute the probability of bowers by
$Prob = \frac{t1t_i}{\sum_{i=1}^{n} fit_n}$
$Fit = \begin{cases} \frac{1}{1+f(xi)}, f(xi) \geq 0 \\ 1 + \lvert f(xi) \rvert, f(xi) < 0 \end{cases}$
            For all bowers
                For all the elements of bowers
                    Choose the bower utilizing roulette wheel

Compute λ_k by

$$\lambda_k = \frac{\alpha}{1 + p_j}$$

Update the position of bower by

$$x_{ik}^{new} = x_{ik}^{old} + \lambda_k \left(\left(\frac{x_{ik} + x_{elite,k}}{2} \right) - x_{ik}^{old} \right)$$

$$N = \left(x_{ik}^{old}, \sigma^2 \right) = x_{ik}^{old} + \left(\sigma * N(0,1) \right)$$

End for

End for

End while

End For

Final Output of Task Scheduler

End Procedure

The backup overlapping concept is employed here with the following constraints [16].

- Initial and backup copies of identical tasks could not be assigned to the similar host.
- An intitial copy could not overlap with other copies.
- Backup copies could not overlap when the initial copies exist on the identical host.

The repeated portion of the active backup copy could not overlap with some copies.

3. Experimental Validation

The performance validation of the SBO-EATSFT technique takes place using the CloudSim tool and the results are examined under varying numbers of tasks. The parameter settings involved in the experimental analysis of the SBO-EATSFT technique are given in Table 1. The CloudSim tool offers a GCC environment with a job dispatcher, resource planner, cloud, and VM instances.

Table 1. Parameter Settings

Parameters	**Values**
Initialization time	200sec
Shutdown time	4sec
Number of cores	8
Transfer power	30.04W
Support power	12W
Processing Time	60-120 sec
ROM	1000 MB
Hard drive rate	1000 MB

A detailed response time analysis of the SBO-EATSFT technique with other methods [17, 18] is offered in Table 2 and Figure 2. The results depicted that the SBO-EATSFT technique has accomplished effective outcomes with least response time under all tasks. For instance, with 50 tasks, the SBO-EATSFT technique has obtained lower response time of 880ms whereas the Cost-Conscious Scheduling algorithm (CCS), Improved Clonel Seection Algorithm (ICSA), Clonel Selection Resources Scheduling Algorithm (CSRSA) and Best Heuristic Scheduling algorithm (BHS) techniques have attained higher response times of 2428ms, 2354ms, 2944ms and 1986ms respectively. Moreover, with 600 tasks, the SBO-EATSFT technique has resulted to lower response time of 17247ms whereas the CCS, ICSA, CSRSA and BHS techniques have accomplished maximum response times of 26036ms, 22924ms, 21007ms and 19827ms respectively.

Table 2. Response Time Analysis of SBO-EATSFT Technique

No. of Tasks	Response Time (ms)				
	CCS	ICSA	CSRSA	BHS Model	SBO-EATSFT
50	2428	2354	2944	1986	880
100	4419	3313	5377	2428	1470
150	7147	5819	8916	4124	2870
200	9211	7810	10243	5967	4566
250	11423	9727	11128	8400	6999
300	13192	11570	12160	10169	7957
350	15035	13561	12897	11496	9063
400	17173	15625	14888	13339	10169
450	19164	17542	16436	15109	12234
500	21376	19680	17837	16952	14077
550	25799	21597	19680	18721	15699
600	26036	22924	21007	19827	17247

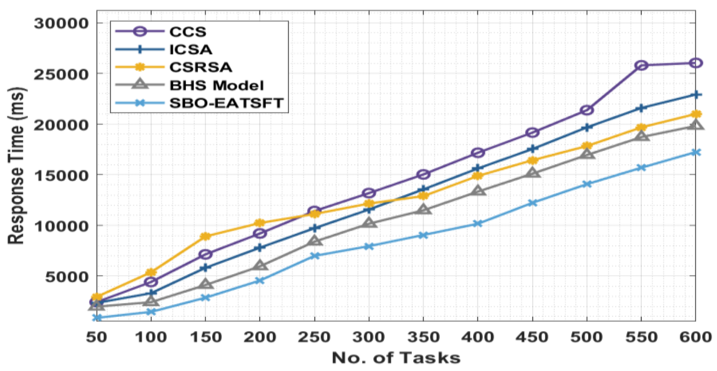

Figure 2. Comparative Response Time Analysis of SBO-EATSFT Technique

A comprehensive execution time analysis of the SBO-EATSFT manner with other approaches is offered in Table 3 and Figure 3. The outcomes demonstrated that the SBO-EATSFT manner has accomplished effective outcomes with wrose execution time under all tasks. With 50 tasks, the SBO-EATSFT technique has reached minimal execution time of 1559ms whereas for 600 tasks execution time is 4954ms.But for other algorithmic techniques values have attained higher execution time .

Table 3. Execution Time Analysis of SBO-EATSFT Technique

No. of Tasks	Execution Time (ms)				
	CCS	ICSA	CSRSA	BHS Model	SBO-EATSFT
50	1907	1771	2013	1665	1559
100	2574	2514	2498	2301	2013
150	2983	3196	3120	2847	2377
200	3787	3711	3574	3332	2862
250	4211	4256	3968	3605	3120
300	4620	4741	4302	3832	3377
350	4878	4817	4650	4120	3726
400	5272	5105	4938	4453	3938
450	5636	5378	5151	4757	4150
500	5817	5651	5393	4999	4408
550	6105	5908	5651	5257	4696
600	6545	6333	5908	5469	4954

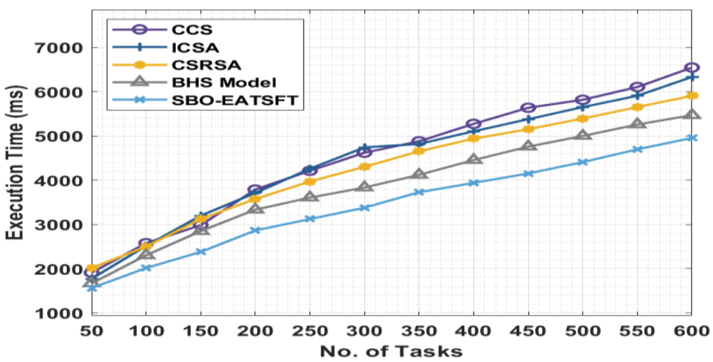

Figure 3. Comparative Execution Time Analysis of SBO-EATSFT Technique

The Average Execution Cost (AEC) analysis of the SBO-EATSFT manner with other algorithms is given in Table 4 and Figure 4. The outcomes depicted that the SBO-EATSFT system has accomplished effective outcomes with least AEC under all tasks. For instance, with 50 tasks, the SBO-EATSFT approach has achieved lower AEC of 89.61$ whereas the CCS, ICSA, CSRSA and BHS manners have reached superior AEC of 302.80$, 309.26$, 270.50$ and 154.21$ respectively. In addition, with 600 tasks, the SBO-EATSFT technique has resulted in decreased AEC of 1433.38$ whereas the CCS, ICSA, CSRSA and BHS algorithms have accomplished higher AEC of 2467.04$, 2299.07$, 1976.05$ and 1640.11$ correspondingly.

Table 4. Average Execution Cost Analysis of SBO-EATSFT Technique

No. of Tasks	Avgerage Execution Cost ($)				
	CCS	ICSA	CSRSA	BHS Model	SBO-EATSFT
50	302.80	309.26	270.50	154.21	89.61
100	367.41	477.24	309.26	173.60	121.91
150	432.01	574.14	302.80	231.74	147.75
200	651.67	735.65	490.16	354.49	186.52
250	923.00	968.23	677.51	509.54	257.58
300	1304.17	1271.87	877.78	722.73	432.01
350	1517.36	1413.99	1045.75	826.10	612.90
400	1749.94	1691.79	1258.94	1032.83	813.18
450	1859.76	1821.00	1446.30	1239.56	1026.37
500	2040.65	1963.13	1659.49	1420.45	1162.04
550	2240.93	2176.32	1788.70	1581.96	1265.40
600	2467.04	2299.07	1976.05	1640.11	1433.38

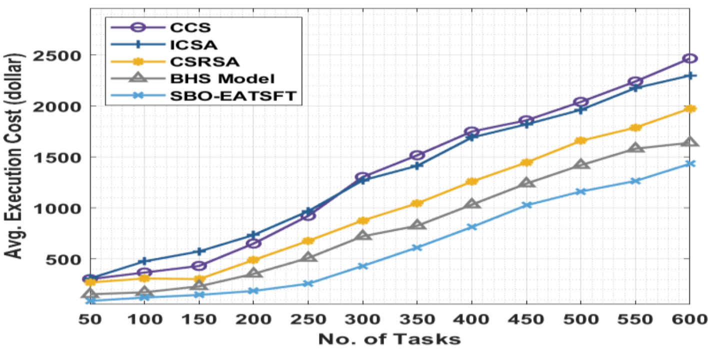

Figure 4. Average Eecution Cost Analysis of SBO-EATSFT Technique

A detailed Average Energy Consumption (AECN) analysis of the SBO-EATSFT method with other system are given in Table 5 and Figure 5. The outcomes outperformed that the SBO-EATSFT approach has accomplished effective outcomes with least AECN under all tasks.

Table 5. Average Energy Consumption Analysis of SBO-EATSFT Technique

No. of Tasks	Average Energy Consumption (kJ)				
	CCS	ICSA	CSRSA	BHS Model	SBO-EATSFT
50	2.01	1.71	1.82	1.57	1.32
100	2.51	2.37	2.28	1.89	1.66
150	2.97	2.87	2.67	2.37	1.98
200	3.56	3.49	3.24	2.90	2.37
250	4.27	4.20	3.88	3.49	2.87
300	5.02	4.91	4.57	4.15	3.26
350	5.96	5.71	4.98	4.54	3.72
400	6.85	6.69	5.62	5.11	4.20
450	7.72	7.51	6.23	5.75	4.57
500	8.84	8.45	7.01	6.51	5.02
550	9.07	8.82	7.81	7.33	5.62
600	9.91	9.45	8.61	7.63	6.51

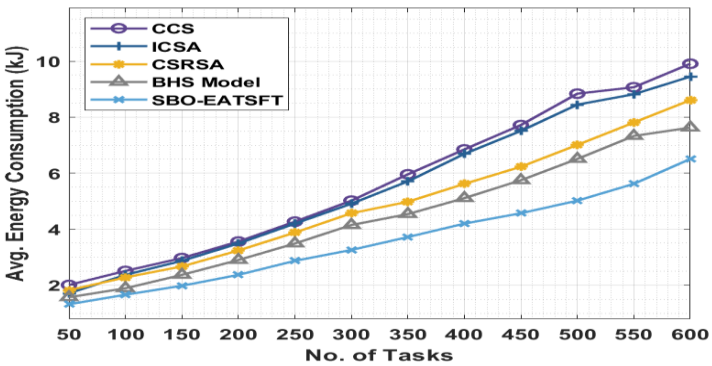

Figure 5. Average Energy Consumption Analysis of SBO-EATSFT Technique

A comprehensive Average execution power AEP analysis of the SBO-EATSFT manner with other methods is offered in Table 6 and Figure 6. The outcomes depicted that the SBO-EATSFT manner has accomplished effectual outcomes with least AEP under all tasks. For instance, with 50 tasks, the SBO-EATSFT algorithm has obtained lower AEP of 1195W whereas the CCS, ICSA, CSRSA and BHS systems have reached maximum AEP of 1507W, 1403W, 1640W and 12997W correspondingly. Besides, with 600 tasks, the SBO-EATSFT technique has resulted in reduced AEP of 4105W whereas the CCS, ICSA, CSRSA and BHS techniques have accomplished superior AEP of 6153W, 5841W, 4728W and 4609W correspondingly.

Table 6. Average Execution Power Analysis of SBO-EATSFT Technique

No. of Tasks	Average Execution Power (W)				
	CCS	ICSA	CSRSA	BHS Model	SBO-EATSFT
50	1507	1403	1640	1299	1195
100	1700	1611	1848	1462	1225
150	1997	1893	1997	1715	1462
200	2368	2323	2264	2012	1655
250	2694	2665	2605	2308	1863
300	3065	2976	2947	2605	2219
350	3451	3437	3333	2976	2472
400	3956	3912	3719	3377	2769
450	4446	4298	3897	3689	3125
500	4995	4773	4164	3971	3466
550	5574	5173	4372	4209	3808
600	6153	5841	4728	4609	4105

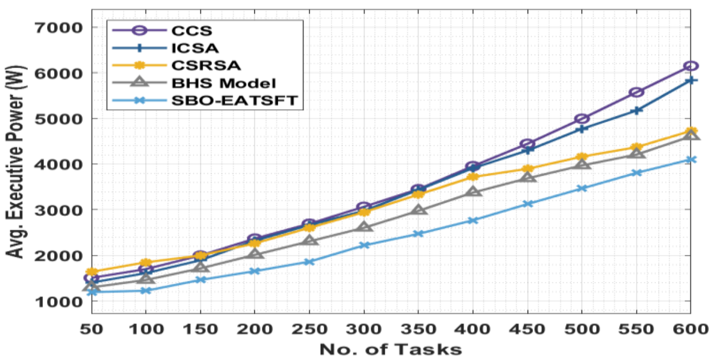

Figure 6. Average execution power analysis of SBO-EATSFT technique

4. Conclusion

This article has presented a new SBO-EATSFT technique to assign the tasks to the resources with maximum energy efficiency and minimum makespan. The tasks are allocated to resources by the use of SBO algorithm with two parameters namely energy and makespan. In addition, the SBO-EATSFT technique derives an effective fault tolerant scheme and thereby enhances the overall performance of the GCC environment. For inspecting the better scheduling outcomes of the SBO-EATSFT technique, a wide range of experiments were carried out under different number of tasks. An extensive set of comparative results analysis highlighted the better performance of the SBO-EATSFT technique over the recent approaches. Therefore, the SBO-EATSFT technique has the ability to achieve better tradeoff between energy consumption and efficiency of the GCC environment. In future, optimal load balancing strategies can be designed for improving the performance of GCC environment.

References

[1] *R. Mandal, M. K. Mondal, S. Banerjee and U. Biswas, "An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing", The Journal of Supercomputing, vol. 76, no. 9,* ***(2020)****, pp. 7374–7393.*

[2] *B. Barzegar, H. Motameni, and A. Movaghar, "EATSDCD: A green energy-aware scheduling algorithm for parallel taskbased application using clustering, duplication and DVFS technique in cloud datacenters", Journal of Intelligent & Fuzzy Systems, vol. 36, no. 6,* ***(2019)****, pp. 5135–5152.*

[3] *Z. Zhou, J. Abawajy, M. Chowdhury et al., "Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms", Future Generation Computer Systems, vol. 86,* ***(2018)****,pp. 836–850.*

[4] *R. Yadav and W. Zhang, "MeReg: managing energy-SLA tradeoff for green mobile cloud computing", Wireless Communications and Mobile Computing, vol. 2017, Article ID 6741972,* ***(2017)****, pp. 1-11.*

[5] *P. A. Sanjay and K. Muthiah, "Green Computing Strategies for Competitive Advantage and Business Sustainability, vol. 16,* ***(2018)****.*

[6] *V. Singh, I. Gupta, and K. Prasanta, "An energy efficient algorithm for workflow scheduling in IaaS cloud", Journal of Grid Computing , vol. 18, no. 3,* ***(2019)****, pp. 357–376.*

[7] *A. Belgacem, K. Beghdad-Bey, H. Nacer, and S. Bouznad, "Efficient dynamic resource allocation method for cloud computing environment", Cluster Computing, vol. 23, no. 4,* ***(2020)****, pp. 2871–2889.*

[8] *R. Yadav, W. Zhang, K. Li et al., "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center", Wireless Networks, vol. 26, no. 3,****(2020)****, pp. 1905–1919.*

[9] *Kalra, M. and Singh, S.,"A review of metaheuristic scheduling techniques in cloud computing", Egyptian Informatics Journal, , vol. 16, no. 3,****(2015)*** *, pp.275-295.*

[10] *Abrol, P., Guupta, S. and Singh, S.,"Nature-Inspired Metaheuristics in Cloud: A review",ICT systems and sustainability,* ***(2020)****, pp.13-34.*

[11] *Abualigah, L. and Diabat, A., "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing",Environments Cluster Computing, , vol. 24, , no. 1,* ***(2021)****, pp.205-223.*

[12] *Yuan, H., Bi, J. and Zhou, M., "Spatial task scheduling for cost minimization in distributed green cloud data centers", IEEE Transactions on Automation Science and Engineering, vol. 16 , no. 2,* ***(2018)****, pp.729-740.*

[13] *Mohammadzadeh, A., Masdari, M., Gharehchopogh, F.S. and Jafarian, A., " Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing", Evolutionary Intelligence,* ***(2020)****, pp.1-30.*

[14] *Ari, A.A.A., Damakoa, I., Titouna, C., Labraoui, N. and Gueroui, A., " Efficient and scalable aco-based task scheduling for green cloud computing environment", IEEE International Conference on Smart Cloud (SmartCloud) , **(2017)**, pp. 66-71.*

[15] *Sofia, A.S. and GaneshKumar, P., "Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II', Journal of Network and Systems Management, vol. 26 , no.2, **(2018)**, pp.463-485.*

[16] *Xu, Z., Sheykhahmad, F.R., Ghadimi, N. and Razmjooy, N., "Computer-aided diagnosis of skin cancer based on soft computing techniques", Open Medicine, vol. 15 , no.1, **(2020)**, pp.860-871.*

[17] *Guo, P., Liu, M. and Xue, Z.," A PSO-based energy-efficient fault-tolerant static scheduling algorithm for real-time tasks in clouds", IEEE 4th International Conference on Computer and Communications (ICCC), **(2018)**, pp. 2537-2541. .*

[18] *Lu, Y. and Sun, N.,"An effective task scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment", Cluster Computing, vol.22, no.1, **(2019)**, pp.513-520.*

[19] *Peng, Z., Barzegar, B., Yarahmadi, M., Motameni, H. and Pirouzmand, P., "Energy-Aware Scheduling of Workflow Using a Heuristic Method on Green Cloud", Scientific Programming, vol. 2020 , Article ID 8898059, **(2020)**.*