Indonesian Journal of Electrical Engineering and Computer Science
Vol. 33, No. 2, February 2024, pp. 1092~1099
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v33.i2.pp1092-1099

Methods for optimizing the assignment of cloud computing resources and the scheduling of related tasks

Zeenath Sultana, Raafiya Gulmeher, Asra Sarwath
Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University (KBNU), Kalaburagi, India

Article Info

Article history:

Received Oct 14, 2023
Revised Nov 16, 2023
Accepted Nov 30, 2023

Keywords:

Cat swarm optimization
Chaos bird swarm
Cloud computing
Deep learning
Machine learning

ABSTRACT

Efficient scheduling algorithms are necessary in the cloud paradigm to optimize service provision to clients while minimizing time duration, energy consumption, and violations of service level agreements (SLAs). Disregarding task appropriateness in resource scheduling can have a detrimental effect on the quality of service provided by cloud providers. Moreover, the utilization of resources in an ineffective manner will necessitate a substantial expenditure of energy to execute activities, leading to prolonged processing duration that adversely affect the temporal duration. Many research projects have focused on employment scheduling problems, and the algorithms used in these studies have offered answers that were deemed nearly flawless. This study presents a chaos bird swarm algorithm (Chaos BSA) approach that use machine learning to consider task priority while allocating tasks to the cloud platform. The method calculates the priorities of task virtual machines and incorporates these values into the scheduler. The scheduler will select tasks that align with the specified priorities and are compatible with the virtual machines. The implementation of the system utilized the openstack cloud platform and the cloudsim tool. The results and comparison with the baseline approach genetic algorithm (GA), ant colony optimization (ACO), and particle swarm optimization (PSO) clearly demonstrate that our Chaos BSA outperforms them by 18% in terms of efficiency.

Corresponding Author:

Zeenath Sultana
Department of Computer Science and Engineering, Faculty of Engineering and Technology
Khaja Bandanawaz University (KBNU)
Kalaburagi, Karnataka, India
Email: profzeenathcse@gmail.com

1. INTRODUCTION

Cloud computing [1] and optimization approaches [2] rapidly gained significant focus in the information technology (IT) industry. With the widespread availability of affordable internet connection and the imminent surge in data produced by applications, a significant amount of data is also being generated. Pay-per-use cloud computing [3] offers on-demand access to computing resources and data. The main objective of cloud computing is to manage the scheduling of activities, processes, and resources, such as central processing unit (CPU), memory, and peripherals. Task scheduling in cloud settings can be performed either statically or dynamically. In a cloud environment, task scheduling [4] considers several elements such as execution time, energy consumption, response time, cost, make-span, and quality of service. Optimization techniques [5], [6] are crucial for the IT sector's growth as they enable the provision of enhanced services and solutions to IT clients, particularly through the use of smart services. The primary concept here is the integration of job scheduling in a cloud environment with optimization methodologies. Optimization techniques [7] are consistently employed to provide intelligent services to end consumers. Supervised or unsupervised external optimization approaches can be employed for the chosen parameters. The request sent to the relevant data centre allows for categorization, enabling the system to determine and classify the most efficient approach for the specific situation [8], [9].

The main objective of this research project is to effectively allocate virtual resources to a diverse group of cloud users, ensuring high-quality service, and simultaneously minimizing energy usage [10]–[13] in the data center for both customers and the provider. In the context of cloud computing, scheduling is a significant challenge due to the fluctuating number of customers who want resources. The cloud provider must employ an efficient scheduling algorithm [14], [15] to meet the customer's expectations and deliver services effectively. Allocating resources in real time based on the specific types of workloads that require cloud services poses significant challenges for cloud providers. In our study, we have meticulously assessed the suitability of occupations by calculating their priorities. These priorities were subsequently included into the scheduler to generate optimal scheduling options.

The ensuing sections outline the contributions of this paper: optimization is employed to develop a task scheduling algorithm [16] that assigns priorities; the technique of job prioritizing is utilized in a scheduling model to allocate tasks to virtual machines (VMs); the simulation algorithm takes a synthetic workload as its input. We propose that an optimal resource allocation model can be identified using optimization approaches [17], specifically by examining the most efficient solution for small-scale data sets. Consequently, we can allocate resources in a way that closely approximates the optimal result. This study aims to transform the complex challenge of allocating cloud resources across several dimensions in an auction setting into a more manageable optimization problem. We achieve this by employing classification or regression techniques. Additionally, we propose two distinct strategies for predicting the distribution of resources based on regression analysis. The core idea is to utilize optimization methods [18] to achieve the most efficient allocation, after the identification of the optimal allocation solution [19], [20] for a user's requirements in small-scale training sets. The final prediction model may choose successful users while ensuring societal welfare, accurate allocation, and optimal resource usage. The remaining manuscript is organized in the following manner: the problem description is outlined in section 1, the literature review is presented in section 2, the proposed methodology is detailed in section 3, simulation results and discussion are provided in section 4, the conclusion, and references are included in the subsequent sections.

2. LITERATURE SURVEY

The resource allocation problem (RAP) [21] in an auction refers to an optimization problem [22] in which customers submit their resource requirements along with the matching value they offer. The objective is to optimize revenue for resource providers and social welfare within the specified resource constraint. The knapsack problem [23], [24] can be likened to the distribution of resources. The definitive, optimal solution to resource allocation problems can be discovered through many means. The problem of allocating resources was resolved by employing integer programming in the study as referenced [25]. To address the same problem, a dynamic programming approach [26] is employed. A randomized resource auction was employed to address a winner decision problem (WDP), and a precise solution was obtained by the utilization of a clique-based approach [27]. Employing a monotone branch-and-bound search approach [28] to obtain the optimal solution. These methods are effective in identifying the optimal solution for small-scale problems. However, as the number of users, resource kinds, and user requirements increase, the computational time required to find the optimum solution increases exponentially. Hence, to achieve practical implementation, a more effective methodology is necessary. The predominant methods employed to solve the problem of approximate resource allocation involve the usage of polynomial runtime approximation scheme (PTAS) [29] algorithms and heuristic algorithms [30]. An n-approximation approach can be used to handle the resource management of several physical devices.

The cloud architecture [31], [32] employed for aviation applications utilizes machine learning algorithms to detect abnormalities, carry out maintenance tasks, predict the onset of component failures, and reduce the overall expenses associated with the lifespan of the aircraft. The purpose of this framework is to emphasize the utilization of optimization techniques for data analytics in cloud-based applications [14], [33]. Mahout machine learning techniques [34] are employed to analyze extensive datasets. It facilitates data analytics for aerospace applications through the use of supervised or unsupervised learning [35].

Odun-Ayo *et al.* [36] presents a comprehensive examination of cloud computing and optimization solutions that are both distributed and based on the software-as-a-service (SaaS) model [37]. The article discusses various cloud-based optimization configurations, including CloudNumbers.com, which utilizes Amazon EC2 to construct computer clusters utilizing scientific computing tools such as Octave, the R system, Mapple, Opani, and CloudStat. The attributes of distributed optimization libraries are also addressed. A presentation is created on a SaaS provider specializing in optimization. The significance of optimization in the context of software as a service provider is discussed.

3. PROPOSED METHOD

The objective of this proposed model is to employ optimized ways to maximize the utilization of cloud computing resources. In the dynamic task scheduling scenario, the task scheduler [38] must allocate the immediate job request to the available resource pool. The scheduler has the option to select from a diverse range of contemporary fundamental and advanced task scheduling techniques, which encompass intricate algorithms such as round robin (RR) [39], [40], shortest job first (SJF), and first come first serve (FCFS). In cloud environments, tasks are planned in a standard dynamic task scheduling manner, where the scheduling is done randomly. The optimization schedule in this proposed framework determines the methods that are deemed most suitable for the specific situation. The selection of the job scheduling approach is determined by the classification of the optimization algorithm [41]. The usefulness and success of cloud computing services rely on the proficiency with which users execute the tasks they submit to the cloud system. Prior to assigning jobs to the task scheduler, it is necessary to employ optimization techniques on the tasks in order to enhance the schedulers' ability to make intelligent decisions. Task scheduling considers various factors, including time span as referred in article [42], energy consumption, waiting time, and response time, in both static and dynamic scheduling situations.

Task scheduling significantly amplifies the source utilization and processing expenses of a cloud-based system. Optimal scheduling is achieved by employing several optimization techniques to enhance the performance of task scheduling. This study implements a chaos bird swarm algorithm (chaos BSA) [43] to optimize job scheduling and improve resource consumption in a cloud setting. Prior studies have yielded many optimization methods for achieving optimal work scheduling. However, there is room for further enhancement in task scheduling algorithms as a result of the initial parameter dependency of existing systems and the issue of local optimum problem. In this study, we have created a chaos BSA algorithm [44]–[46] to create an optimal task scheduling scheme in a cloud environment. The purpose is to combine chaotic behavior with optimization techniques in order to improve the exploration and exploitation capabilities of the solution search space. The evaluation of the result is based on the total cost, which includes the expenses related to sources, storage, and processing, in relation to the number of iterations and tasks.

3.1. Chaos BSA-based optimization approach for task scheduling

3.1.1. Task scheduling model with multiple objectives

Resource utilization is directly embellished by task scheduling strategies for fundamental systems in a cloud environment. Task scheduling have a major concern with assigning the input tasks to the virtual systems (VSs). Initially, the client's jobs will be divided into a group of tasks, which are computed separately from each other. In this paper, the following steps will be performed: i) initially, resources and tasks will be related on the basis of complete information about client tasks and existing virtual systems in harmony amid a definite scheme; ii) then, the optimal task implementation scheme will be obtained by a task scheduler according to the relation to convene the allocation requests; and iii) at last, the optimal scheme is sent to the cloud environment for implementation, and the outcomes will be delivered to the clients.

Multi-objective optimization function [47] is a proposed model that can perform task scheduling in a well-organized manner on the basis of CPU potency and storage. The first objective processing cost function, F_{PC} as in (1), is evaluated by utilizing it to describe the task completion time in terms of CPU potency, and the second objective load cost function, F_{LC} as in (2), is calculated in terms of storage. Similarly, the third objective source cost function, F_{SC} as in (3), is calculated by using power expenditure to utilize the sources. All the parameters P_v, P_c, L_v, L_c, S_v, and S_c as mentioned in Table 1 are on dissimilar scales, so normalization is also applied to objective functions. At last, the absolute multi-objective optimization function FMO is obtained by combining all three objective functions with some weight coefficient factors (W_a) as in Table 1.

$$F_{pc} = \frac{1}{T}\sum_{a=1}^{T}\sum_{b=1}^{N} 0_{ab}\,\frac{P_{c,a}/P_{v,b}}{\max\left\{P_{c,a}/P_{v,b}\right\}} \tag{1}$$

$$F_{LC} = \frac{1}{T}\sum_{a=1}^{T}\sum_{b=1}^{N} 0_{ab}\,\frac{L_{c,a}/L_{v,b}}{max\left\{L_{c,a}/L_{v,b}\right\}} \tag{2}$$

$$F_{SC} = \frac{1}{T}\sum_{a=1}^{T}\sum_{b=1}^{N} 0_{ab}\,\frac{(UP_{c,a}, S_{c,a})/(P_{v,b}S_{c,b})}{\max\{(UP_{c,a}S_{c,a})/(P_{v,b}S_{v,b})\}} \tag{3}$$

here,
$P_{c,a}$ and $P_{v,b}$ = P_c value of a^{th} task and P_v value of b^{th} VS respectively.
$L_{c,a}$ and $L_{v,b}$ = L_c value of a^{th} task and L_v value of b^{th} VS respectively.
$S_{c,a}$ and $S_{v,b}$ = Lc_c value of a^{th} task and S_v value of b^{th} VS respectively.

Table 1. Symbols and their description

Symbol	Description
T	A group of client tasks
N	A group of virtual systems
O_{ab}	Assessment variable to denote that the a^{th} task is executed on the b^{th} VS
P_v (P_c)	Processing strength matrix for VSs (tasks)
L_v (L_c)	Load strength matrix for VSs (tasks)
S_v (S_c)	Source bandwidth matrix for VSs (tasks)
U	Unit cost
W_a	Coefficient weight factor (a=1, 2, 3)

3.1.2. Bird swarm algorithm

A BSA is a nature-enthused approach imitative of bird foraging, observation, and departure activities in the environment. In bird-foraging nature, every bird searches for food based on individual knowledge or group skill. If an arbitrary number consistently dispersed among (0, 1), at that time the bird would scavenge for food. Or else, the bird would remain observant as in (4).

$$X_{e,f}^{t+1} = X_{e,f}^{t} + (R_{e,f} - X_{e,f}) * \alpha * \text{rand } (0.1) + (G_f - X_{e,f}) * \beta * \textit{rand } (0,1) \tag{4}$$

$X_{e,f}$=e^{th} bird location in f^{th} dimension and t^{th} population.
$R_{e,f}$=birds best prior location.
G_f=bird`s best prior location shared by swarm.
Rand (0,1)=arbitrary number between (0,1)
α and β=coefficient of learning
D=number of birds

In departure activities, birds will take off to a new position to look for food because of the hazard of a killer or other cause. Few birds perform as creators, but the others desire to obtain food from creators. Task scheduling is based on chaos BSA. The task scheduling approach is explained on the basis of chaos BSA in Table 2, where birds are denoted as client tasks in a cloud environment. In exploration, a bird has a location according to the task schedule and has a solution. The creator’s bird’s location denotes the present best solution, and the creator’s bird’s fitness value represents the present best value of the absolute multi-objective optimization function FMO. In this manner, a chaos BSA is applied to find the optimal solution for client task scheduling in a cloud environment. The entire birds are modified in their locations in every iteration, and the location data is transferred to a solution for existing client tasks. With the help of FMO, the fitness value of birds is evaluated. The whole procedure will be continued until the last iteration. The location data of the final creator bird will be filled out in the solution, which is utilized for obtaining the optimal task execution strategy in a cloud environment.

Table 2. Terms used in BSA and task scheduling

BSA terms	Task scheduling
Individual bird	Client`s tasks in cloud
Bird foraging nature	Best solution exploration
Bird location	A solution O tn for FMO
Creator bird	Best solution O tn for FMO
Bird`s fitness	FMO value

The following steps are performed on the proposed chaos BSA for task scheduling:
- Step 1: firstly, the mapping between client tasks and birds is to be performed.
- Step 2: the location of birds, dimension of exploration, number of iterations, and constant values are initialized. The bird’s population is initialized by using chaos BSA.
- Step 3: the fitness values of each bird are evaluated on the basis of the bird’s location data in the best solution exploration procedure. The minimum fitness-valued bird is selected as the present optimal solution.

– Step 4: all the birds have modified their locations.
– Step 5: steps 3 to 4 is performed for all iterations.
– Step 6: the location data of the final creator bird will be filled out in the solution, which is utilized for obtaining the optimal task execution strategy in a cloud environment.

4. EXPERIMENTS AND RESULTS

The chaotic BSA algorithm is implemented using the MATLAB 2019 a tool specified in Table 3, and the resulting outputs are compared to those of other algorithms, namely genetic algorithm (GA), ant colony optimization (ACO), and particle swarm optimization (PSO). The overall costs are assessed for 100 to 2,000 tasks, with the number of iterations ranging from 10 to 100, as shown in Tables 4 and 5, respectively. The total expenses are also computed based on the range of tasks, which varies from 200 to 2,000.

Table 3. Experimental parameters

Parameters	Values
Tool	MATLAB 2019a
Operating system	Windows
Number of birds	[100, 1.000]
Number of iterations	[10, 100]
Number of tasks	[200, 2.000]
Weight coefficient factors	0.25, 0.25, 0.50

Table 4. Parameters of simulation

Parameter	Value range (VS)	Value range (tasks)
Storage	[150, 600]	[60, 110]
Source	[150, 300]	[25, 60]
CPU	[250, 600]	[15, 60]

Table 5. The value of FMO for 100 tasks

Number of iterations	GA	ACO	PSO	Chaos BSA
10	0.352	0.3125	0.289	0.2135
20	0.369	0.3145	0.2789	0.2536
30	0.3112	0.3126	0.3385	0.2136
40	0.3251	0.3256	0.2853	0.2365
50	0.3564	0.3874	0.3624	0.2145
60	0.3526	0.3641	0.3211	0.2911
70	0.3965	0.3548	0.3125	0.296
80	0.3265	0.3621	0.2987	0.2154
90	0.36985	0.3251	0.2941	0.2541
100	0.3652	0.3254	0.3114	0.2874

The results in Figure 1 illustrate that ACO provides 10% better efficiency against GA, PSO obtains 14% better outcomes against ACO, and 22% better outcomes against GA. Chaos BSA generates 15% superior efficiency against PSO, 26% superior efficiency against ACO, and 33% superior efficiency against GA in terms of overall cost for a small number of tasks (100 tasks) on the basis of number of iterations. The results in Figure 2 illustrate that the ACO provides 9% better efficiency against GA; PSO obtains 15% better outcomes against ACO, 23% better outcomes against GA; chaos BSA generates 15% superior efficiency against PSO, 25% superior efficiency against ACO, 34% superior efficiency against GA in terms of overall cost for small number of tasks (300 tasks) on the basis of number of iterations. The results in Figure 3 illustrate that ACO provides 9% better efficiency against GA, PSO obtains 13% better outcomes against ACO, and 21% better outcomes against GA. Chaos BSA generates 18% superior efficiency against PSO, 28% superior efficiency against ACO, and 34% superior efficiency against GA in terms of overall cost for a small number of tasks (500 tasks) on the basis of number of iterations. The results in Figure 4 explain that the ACO provides 11% superior efficiency against GA, the PSO obtains 13% superior outcomes against ACO, and the chaos BSA generates 16% superior performance against PSO, 26% superior performance against ACO, and 35% superior performance against GA in terms of overall cost for a large number of tasks (700 tasks) on the basis of number of iterations.

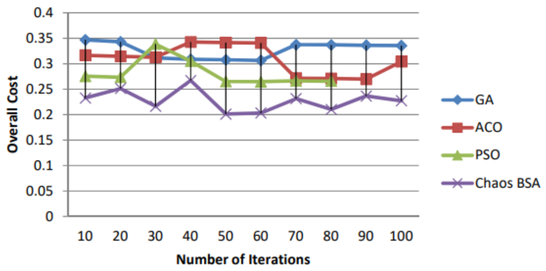

Figure 1. The value of FMO for 100 tasks

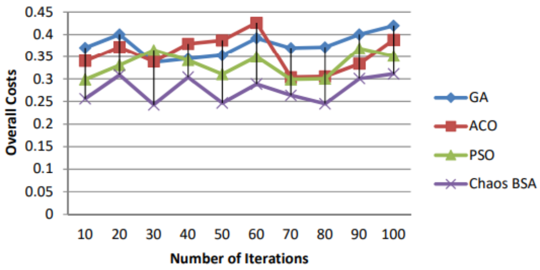

Figure 2. The value of FMO for 300 tasks

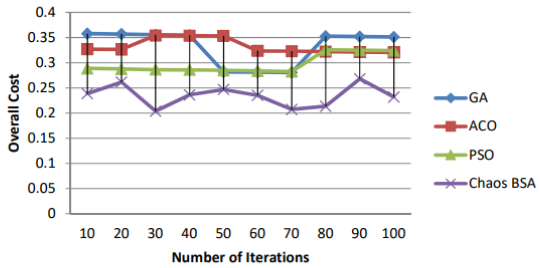

Figure 3. The value of FMO for 500 tasks

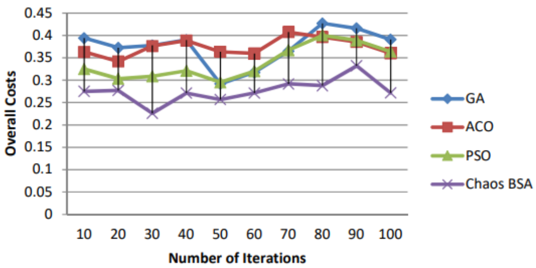

Figure 4. The value of FMO for 700 tasks

The results describe that ACO provides 8% superior performance against GA, PSO obtains 12% superior outputs against ACO and 19% superior outputs against GA, and chaos BSA generates 17% better performance against PSO, 26% better performance against ACO, and 32% better performance against GA in terms of overall cost for 100 iterations on the basis of number of tasks. It means that the proposed chaos BSA generates better results than GA, ACO, and PSO on the basis of the number of tasks and number of iterations. The graphs illustrate that the overall cost of all algorithms decreases with an increase in the number of iterations. It means that the cost decreases throughout the best solution exploration procedure.

5. CONCLUSION

The utilization of resources and the expenses associated with implementing a cloud-based system are significantly influenced by job scheduling. Multiple academics have devised and applied numerous optimization methods to achieve optimal job scheduling in a cloud setting. In this study, we implement a chaos BSA to achieve efficient resource use for optimal task scheduling by incorporating chaotic behavior. The simulation is conducted using the MATLAB 2019a software. The resultsdemonstrate that chaos BSA outperforms GA, ACO, and PSO in terms of overallcost, for both small and high job numbers and iterations.

REFERENCES

[1] A. Gautam, "Cloud computing: a review paper," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 6, pp. 3233–3236, 2022, doi: 10.22214/ijraset.2022.44588.

[2] L. Kang, R. S. Chen, Y. C. Chen, C. C. Wang, X. Li, and T. Y. Wu, "Using cache optimization method to reduce network traffic in communication systems based on cloud computing," *IEEE Access*, vol. 7, pp. 124397–124409, 2019, doi: 10.1109/ACCESS.2019.2938044.

[3] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "NIST cloud computing reference architecture," in *Proceedings - 2011 IEEE World Congress on Services, SERVICES 2011*, Jul. 2011, pp. 594–596, doi: 10.1109/SERVICES.2011.105.

[4] N. Bansal and M. Dutta, "Performance evaluation of task scheduling with priority and non-priority in cloud computing," in *2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2014*, Dec. 2015, pp. 547–553, doi: 10.1109/ICCIC.2014.7238289.

[5] N. Bansal and A. K. Singh, "Effective task scheduling algorithm in cloud computing with quality of service alert bees and grey wolf optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, pp. 550–560, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp550-560.

[6] N. Bansal and A. K. Singh, "Grey wolf optimized task scheduling algorithm in cloud computing," in *Advances in Intelligent Systems and Computing*, vol. 1013, 2020, pp. 137–145.

[7] Y. Xu, K. Li, L. He, L. Zhang, and K. Li, "A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3208–3222, Dec. 2015, doi: 10.1109/TPDS.2014.2385698.

[8] A. Ullah, N. M. Nawi, E. Sutoyo, A. Shazad, S. N. Khan, and M. Aamir, “Search engine optimization algorithms for page ranking: comparative study,” *International Journal of Integrated Engineering*, vol. 10, no. 6, pp. 19–25, 2018, doi: 10.30880/ijie.2018.10.06.003.

[9] L. S. Vailshery, “Challenges to enterprise cloud computing usage worldwide in 2019 to 2023,” 2023, https://www.statista.com/statistics/511283/worldwide-survey-cloud-computing-risks/ (accessed May 24, 2019).

[10] J. F. Kovar, “Data center power consumption grow less than expected: report,” *Crn*, 2011. http://www.crn.com/news/data-center/231400014/data-center-power-consumption-grows-less-than-expected-report.htm (accessed Aug. 10, 2011).

[11] S. Behere, “Data center energy efficiency,” *31st West Coast Energy Management Congress, EMC 2013*, 2013.

[12] A. S. Abdalkafor and K. M. A. Alheeti, “A hybrid approach for scheduling applications in cloud computing environment,” *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 1387–1397, 2020, doi: 10.11591/ijece.v10i2.pp1387-1397.

[13] K. Venkatachalapathy and D. Sundaranarayana, “A min-max scheduling load balanced approach to enhance energy efficiency and performance of mobile ADHOC networks,” *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3416460.

[14] A. S. Abdalkafor, A. A. Jihad, and E. T. Allawi, “A cloud computing scheduling and its evolutionary approaches,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 21, no. 1, pp. 489–496, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp489-496.

[15] N. Bansal and A. K. Singh, “Trust for task scheduling in cloud computing unfolds it through fruit congenial,” in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 4, 2018, pp. 41–48.

[16] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, “Resource allocation in a network-based cloud computing environment: design challenges,” *IEEE Communications Magazine*, vol. 51, no. 11, pp. 46–52, Nov. 2013, doi: 10.1109/MCOM.2013.6658651.

[17] C. Aroef, R. P. Yuda, Z. Rustam, and J. Pandelaki, “Multinomial logistic regression and support vector machine for osteoarthritis classification,” *Journal of Physics: Conference Series*, vol. 1417, no. 1, p. 012012, Dec. 2019, doi: 10.1088/1742-6596/1417/1/012012.

[18] B. Muthulakshmi and K. Somasundaram, “A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment,” *Cluster Computing*, vol. 22, no. S5, pp. 10769–10777, Sep. 2019, doi: 10.1007/s10586-017-1174-z.

[19] G. Sreenivasulu and I. Paramasivam, “Hybrid optimization algorithm for task scheduling and virtual machine allocation in cloud computing,” *Evolutionary Intelligence*, vol. 14, no. 2, pp. 1015–1022, Jun. 2021, doi: 10.1007/s12065-020-00517-2.

[20] P. Xu, G. He, Z. Li, and Z. Zhang, “An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 12, p. 155014771879379, Dec. 2018, doi: 10.1177/1550147718793799.

[21] A. Kaur and B. Kaur, “Load balancing optimization based on hybrid heuristic-metaheuristic techniques in cloud environment,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 813–824, Mar. 2022, doi: 10.1016/j.jksuci.2019.02.010.

[22] L. Abualigah and A. Diabat, “A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments,” *Cluster Computing*, vol. 24, no. 1, pp. 205–223, Mar. 2021, doi: 10.1007/s10586-020-03075-5.

[23] F. Aisopos, K. Tserpes, and T. Varvarigou, “Resource management in software as a service using the knapsack problem model,” *International Journal of Production Economics*, vol. 141, no. 2, pp. 465–477, Feb. 2013, doi: 10.1016/j.ijpe.2011.12.011.

[24] S. Laabadi, M. Naimi, H. E. Amri, and B. Achchab, “The 0/1 multidimensional knapsack problem and its variants: a survey of practical models and heuristic approaches,” *American Journal of Operations Research*, vol. 08, no. 05, pp. 395–439, 2018, doi: 10.4236/ajor.2018.85023.

[25] P. Zhang, M. C. Zhou, and X. Wang, “An intelligent optimization method for optimal virtual machine allocation in cloud data centers,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1725–1735, Oct. 2020, doi: 10.1109/TASE.2020.2975225.

[26] S. C. Nayak, S. Parida, C. Tripathy, and P. K. Pattnaik, “An enhanced deadline constraint based task scheduling mechanism for cloud environment,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 282–294, Feb. 2022, doi: 10.1016/j.jksuci.2018.10.009.

[27] L. Wang, L. Zhou, J. Lu, and J. Yip, “An order-clique-based approach for mining maximal co-locations,” *Information Sciences*, vol. 179, no. 19, pp. 3370–3382, Sep. 2009, doi: 10.1016/j.ins.2009.05.023.

[28] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, “Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning,” *Discrete Optimization*, vol. 19, pp. 79–102, Feb. 2016, doi: 10.1016/j.disopt.2016.01.005.

[29] I. Trummer, B. Faltings, and W. Binder, “Multi-objective quality-driven service selection-a fully polynomial time approximation scheme,” *IEEE Transactions on Software Engineering*, vol. 40, no. 2, pp. 167–191, Feb. 2014, doi: 10.1109/TSE.2013.61.

[30] D. Comes, H. Baraki, R. Reichle, M. Zapf, and K. Geihs, “Heuristic approaches for QoS-based service selection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6470 LNCS, 2010, pp. 441–455.

[31] F. Jauro, H. Chiroma, A. Y. Gital, M. Almutairi, S. M. Abdulhamid, and J. H. Abawajy, “Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend,” *Applied Soft Computing Journal*, vol. 96, p. 106582, Nov. 2020, doi: 10.1016/j.asoc.2020.106582.

[32] Z. Sultana and D. R. Gulmeher, “A survey on artificial intelligence and machine learning for resources allocation in cloud data centers,” *JETIR*, vol. 10, no. 3, 2023.

[33] W. A. R. W. M. Isa, A. I. H. Suhaimi, N. Noordin, A. F. Harun, J. Ismail, R. A. Teh, “Cloud computing adoption reference model,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 16, no. 1, pp. 395–400, 2019, doi: 10.11591/ijeecs.v16.i1.pp395-400.

[34] R. Anil *et al.*, “Apache Mahout: machine learning on distributed dataflow systems,” *ACM Digital Library, The Journal Of Machine Learning Research*, vol. 21, no. 127, pp. 1−6, 2020.

[35] A. Abdelhadi, S. Zainudin, and N. S. Sani, “A regression model to predict key performance indicators in higher education enrollments,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 454–460, 2022, doi: 10.14569/IJACSA.2022.0130156.

[36] I. Odun-Ayo, T. A. Williams, and J. Yahaya, “Cloud management and monitoring: a systematic mapping study,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 21, no. 3, pp. 1648–1662, 2021, doi: 10.11591/ijeecs.v21.i3.pp1648-1662.

[37] J. K. R. Sastry and M. T. Basu, “Securing SAAS service under cloud computing based multi-tenancy systems,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 13, no. 1, pp. 65–71, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp65-71.

[38] X. Wu, M. Deng, R. Zhang, B. Zeng, and S. Zhou, "A task scheduling algorithm based on QoS-driven in cloud computing," *Procedia Computer Science*, vol. 17, pp. 1162–1169, 2013, doi: 10.1016/j.procs.2013.05.148.
[39] H. F. Ugurdag and O. Baskirt, "Fast parallel prefix logic circuits for n2n round-robin arbitration," *Microelectronics Journal*, vol. 43, no. 8, pp. 573–581, Aug. 2012, doi: 10.1016/j.mejo.2012.04.005.
[40] M. Oveis-Gharan and G. N. Khan, "Index-based round-robin arbiter for NoC routers," in *2015 IEEE Computer Society Annual Symposium on VLSI*, Jul. 2015, vol. 07-10-July, pp. 62–67, doi: 10.1109/ISVLSI.2015.27.
[41] P. Arora and A. Dixit, "An optimized load balancing algorithm in cloud computing," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 5, pp. 683–688, Jun. 2020, doi: 10.35940/ijeat.e9242.069520.
[42] V. K. Patel and M. H. Pandya, "Learning of scheduling algorithm with maximum compatible activity or minimum time span," *International Journal of Engineering Development and Research (IJEDR)*, vol. 1, no. 2, 2014.
[43] X. B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, "A new bio-inspired optimisation algorithm: bird swarm algorithm," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 673–687, Jul. 2016, doi: 10.1080/0952813X.2015.1042530.
[44] M. Ahmad, N. Javaid, I. A. Niaz, S. Shafiq, O. U. Rehman, and H. M. Hussain, "Application of bird swarm algorithm for solution of optimal power flow problems," in *Advances in Intelligent Systems and Computing*, vol. 772, 2019, pp. 280–291.
[45] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons and Fractals*, vol. 40, no. 4, pp. 1715–1734, May 2009, doi: 10.1016/j.chaos.2007.09.063.
[46] M. M. Akawee, M. A. Ahmed, and R. A. Hasan, "Using resource allocation for seamless service provisioning in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 26, no. 2, pp. 854–858, May 2022, doi: 10.11591/ijeecs.v26.i2.pp854-858.
[47] F. Zou, G. G. Yen, L. Tang, and C. Wang, "A reinforcement learning approach for dynamic multi-objective optimization," *Information Sciences*, vol. 546, pp. 815–834, Feb. 2021, doi: 10.1016/j.ins.2020.08.101.

BIOGRAPHIES OF AUTHORS

Zeenath Sultana perceived B.E., M.Tech. from Visvesvaraya Technological University in 2006 and 2010 respectively. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University, Kalaburagi, Karnataka, India. She is the research scholar at Khaja Bandanawaz University, Kalaburagi, Karnataka, India. Her area of interest in research work is deep learning, machine learning, cloud computing, cyber security. She has published many articles in national and international journals and is an active reviewer for numerous international journals. She can be contacted at email: profzeenathcse@gmail.com.

Dr. Raafiya Gulmeher is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University, Kalaburagi, Karnataka, India. She received Ph.D. degree in computer science from JJTU, Rajasthan, India. She has more than 20 years of academic experience and have published many articles in international and national journals. She can be contacted at email: profraafiya.cse@gmail.com.

Asra Sarwath perceived B.E., M.Tech. from Visvesvaraya Technological University in 2007 and 2014 respectively. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University, Kalaburagi, Karnataka, India. She is the research scholar at Khaja Bandanawaz University, Kalaburagi, Karnataka, India. Her area of interest in research work is deep learning, machine learning, cyber security, IoT. She has published many articles in national and international journals. She can be contacted at email: asra.sarwath2003@gmail.com.