CAMES, **29**(1–2): 87–103, 2022, doi: 10.24423/cames.405

This article belongs to the *Special Issue on Advanced Optimization Methods for Uncertainties in Intelligent Industrial Systems* edited by Dr. I.J. Jacob, Dr. B.H. bin Ahmad and Dr. Z.F. Khan

A Hybrid Optimized Resource Allocation Model for Multi-Cloud Environment Using Bat and Particle Swarm Optimization Algorithms

D. SELVAPANDIAN*, R. SANTHOSH

Department of Computer Science and Engineering
Faculty of Engineering
Karpagam Academy of Higher Education
Coimbatore, India
*Corresponding Author e-mail: selvapandian79@gmail.com

In cloud computing, scheduling and resource allocation are the major factors that define the overall quality of services. An efficient resource allocation module is required in cloud computing since resource allocation in a single cloud environment is a complex process. Whereas resource allocation in a multi-cloud environment further increases the complexity of allocation procedures. Earlier, resources from the multi-cloud environment were allocated based on task requirements. However, it is essential to analyze the present resource availability status and resource capability before allocating to the requested tasks. So, in this research work, a hybrid optimized resource allocation model is presented using bat optimization algorithm and particle swarm optimization algorithm to allocate the resource considering the resource status, distance, bandwidth, and task requirements. Proposed model performance is evaluated through simulation and compared with conventional optimization algorithms. For a set of 500 tasks, the proposed approach allocates resources in 47 s, with a minimum energy consumption of 200 kWh. Compared to conventional approaches, the performance of the proposed model is much better in terms of deadline missed tasks, resource requirement, energy consumption, and allocation time.

Keywords: multi-cloud computing, resource allocation, hybrid optimization, bat algorithm, particle swarm optimization, quality of services (QoS).

1. Introduction

Cloud computing is an unavoidable paradigm in the present era. Due to the numerous benefits, ease of access, low cost, and other benefits, clouds are adopted by small-scale to large-scale organizations. People with smartphones access the cloud services through applications every day, and the number of users increases rapidly. Nevertheless, resource utilization in cloud computing is a major factor.

Since most of the resources are virtual and utilized by millions of people, proper resource allocation management is required to maintain the quality of services between users and cloud service providers. Nowadays, due to diverse applications and processing requirements, users move towards multi-cloud architecture. Instead of using single cloud for all the services, multi-cloud offers the users access to the services over a multi-cloud environment depending on their requirements. Figure 1 illustrates a multi-cloud system in which users can access the services from different cloud service providers based on their requirements. The cloud management broker takes the responsibility of scheduling, resource allocation, and security factors [1, 2] in the multi-cloud system.

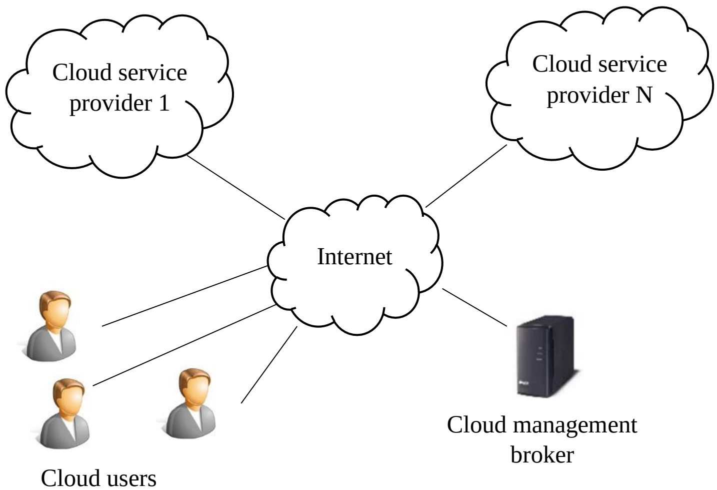

Fig. 1. Multi-cloud system.

Resource allocation in conventional cloud computing is a complex process as each task and user has different requirements. Requests in the cloud services are considered tasks and scheduled to execute through scheduling algorithms. Cloud resources are required to execute the scheduled tasks, so an efficient cloud resource allocation procedure must be established to avoid service level agreement (SLA) violations. In the case of a multi-cloud environment, resource allocation from multiple clouds is more complex than the conventional cloud resource allocation procedure. It requires an optimal resource for the given request and the resource must be free to accommodate the new request. The current strategy followed in multi-cloud resource allocation is identifying the current load of resources and allocating them to the respective task. However, it is essential to look into other factors such as energy consumption, allocation time, resource utilization, etc. Figure 2 illustrates a resource allocation procedure in a cloud environment.

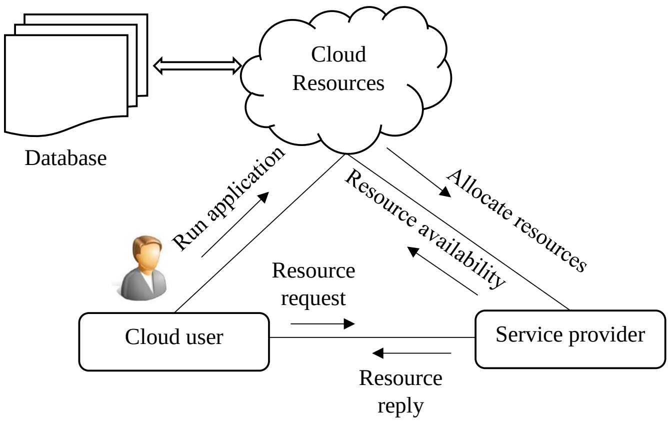

Fig. 2. Resource allocation in a cloud environment.

This research analyzes the issues in multi-cloud resource allocation and provides a solution through a hybrid optimization algorithm. Nature-inspired bat optimization algorithm (BOA) and particle swarm optimization (PSO) algorithm are combined as a hybrid approach to reduce the issues in multi-cloud resource allocation. Among all other optimization algorithms, the selected optimization algorithms are well known for their global optimal solution and quick convergence characteristics. With the optimization algorithm benefits, an efficient resource allocation model is presented in this research work. The major contributions of this research are summarized as follows:

- A hybrid nature-inspired optimization algorithm for multi-cloud resource allocation using bat optimization and PSO algorithms is presented.
- The optimal resource selection and allocation model performance for the given resource request through intense simulation are presented.
- Comparative analysis of proposed hybrid optimization models and conventional optimization models is presented to validate the superior performance.

The rest of the research work is structured as follows. Various cloud resource allocation methodologies and their features are presented as a literature review in Sec. 2. The proposed hybrid optimization algorithm for multi-cloud resource allocation is described in Sec. 3. Simulation analysis and its discussion are presented in Sec. 4. At last, the observations are concluded in Sec. 5.

2. Related works

A brief survey on resource allocation in cloud computing is presented in this section. Recent methodologies and their features are analyzed in detail to frame the research motivation. A priority-aware virtual machine allocation process reported in [3] avoids network congestion by placing the high-priority application processing virtual machine near the host. This process ensures the user gets guaranteed bandwidth and high-quality service in a multi-tenant cloud data center. A similar priority-based resource allocation model reported in [4] discusses the challenges in multimedia content processing in vehicular clouds. Response time, quality of services, and resource cost are the major parameters considered for analysis to provide an efficient resource allocation in multimedia cloud computing.

Resource underutilization and overutilization occur in cloud computing due to inefficient resource allocation. To avoid a skewed resource utilization allocation procedure reported in [5] prevents resource starvation and resource waste in the cloud environment. The presented resource allocation algorithm verifies the workload status and allocation correctness to avoid resource underutilization. Fair resource allocation in cloud computing reported in [6, 7] presents hierarchical long-term resource fairness and an extended long-term resource fairness approach. The fairness issues in resource allocation policies are reduced by the presented approach and perform better than the existing fair scheduling methods.

Service requests and resource demands will keep on changing in cloud computing. Service provided to these dynamic requests and resource allocation needs an adaptive allocation procedure to satisfy the quality of services. The shared cloud computing resource introduces interference and leads to request failures affecting the overall service. An adaptive control strategy reported in [8] allocates resources for dynamic requests considering the interferences in other hosted services to maintain a better quality of services. A dynamic power-saving resource allocation model reported in [9] improves energy efficiency and power-optimized resource allocation using PSO algorithm. The dynamic model considers the energy consumption of virtual machines, physical machines and the air conditioner energy efficiency ratio in the forecasting process. Due to this, virtual machine migrations are eliminated and required essential resources are allocated to the virtual machine.

The multi-cloud model reported in [10] provides optimal resource allocation in a multi-tenant distributed environment using an improved multi-objective genetic algorithm with a k-means approach. Initially, a radial basis function neural network is introduced to convert simulation tasks into specific resource requirements for efficiency improvement. The genetic algorithm is combined with multi-agent optimization in [11] as a hierarchical multi-agent optimization model that enhances resource utilization by reducing the bandwidth cost in cloud computing. The requested tasks are processed through the service nodes obtained by a genetic algorithm and bandwidth cost is minimized by a decentralized multi-agent optimization algorithm to attain better resource management in the cloud model. A similar multi-objective genetic algorithm reported in [12] forecasts the energy consumption and resource utilization to formulate the issues in resource allocation. Based on the prediction results, resource requirements are forecasted and VMs are placed for the next slot, which improves memory, CPU utilization, and also reduces data center energy consumption.

Intercloud resource allocation is a crucial task in cloud computing as all the resources are controlled by different cloud service providers. An agent-based resource allocation procedure reported in [13] interacts with other agents and takes independent decisions in the resource allocation process. Though the decisions can be made, the final allocation process still needs authentication from the cloud service provider, which is the limitation of the presented approach. A cooperative resource allocation procedure reported in [14] discussed the issues in market ownership and operation in cloud computing. To secure the resource allocation protocols, cryptographic techniques are used in the presented approach. The analysis demonstrates the better performance of cooperative infrastructure in terms of allocation performance, overhead, and cost analysis.

Machine learning-based cloud resource allocation identifies the suitable resources through its learning characteristics. However, it requires an additional feature extraction module before classification. An adaptive resource allocation using reinforcement learning algorithm reported in [15, 16] monitors the system changes through a virtual machine hiring policy to satisfy quality of service (QoS) requirements. The adaptive model monitors the changes in system capacity, service cost, and service demand to enhance the cloud provider profit and avoid SLA violations. A simple fuzzy logic-based resource allocation model reported in [17] reduces the SLA violations and improves the resource allocation performances through a fuzzy management system. The adaptive model measures the resource status through sensors and processes them in the fuzzy controller to make decisions. The dynamic update of the membership function satisfies the QoS requirements and enhances the resource allocation performances.

Various optimization models were introduced to select optimal resources for the tasks. Modified PSO and modified cuckoo search optimization algorithm-based resource management and task scheduling are reported in [18]. The conventional optimization parameters are changed based on the cloud computing environment requirements and allocate the resources in an efficient manner. Improved reliability and reduced average response time are the observed merits of the modified optimization model. The task allocation model for the multi-cloud environment presented in [19] reduces the make-span and energy consumption. The presented energy-aware allocation procedure overlaps the makespan for different tasks not producing any deviations in the allocation process. Compared to existing fuzzy-based resource allocation, cloud z-score normalization of the presented energy-aware algorithm performs better in terms of energy consumption and makespan.

A multi-objective scheduling algorithm presented in [20] schedules the workflow based on PSO and allocates resources based on fuzzy logic. The presented approach considers the reliability and resource utilization in the allocation process. Minimized computation cost and makespan are the observed merits of the presented research model. The multi-cloud resource allocation procedure presented in [21] introduces a hybrid genetic algorithm model considering the response time, latency and execution time for optimum resource allocation. By introducing new features such as domain-tailored service clustering, solution representation, repair algorithm and basic genetic operators, the presented approach performs better than conventional optimization models.

The joint optimization between offloading decisions and resource allocation is considered in the research model presented in [22] to minimize the energy consumption in user terminal devices. The optimization problem is formulated as a nonlinear programming model considering the offloading decisions, power allocation, channel selection, and resource allocation [24]. The iterative procedure produces a stable solution for the optimization problems compared to traditional methods. From the analysis, the importance of resource allocation in cloud computing is observed. However, existing methods select resource based on task requirements and do not consider the other parameters [25]. Specifically, multi-cloud resource allocation is reported in a few works, but the allocation performances can be improved further if the above-mentioned parameters are considered in resource selection [26]. Considering these limitations, this research work presents a hybrid optimized resource allocation for a multi-cloud environment to improve the allocation performance and reduce energy consumption.

3. Proposed work

The proposed optimized resource allocation in multi-cloud computing is developed by combing the optimization characteristics of the bat algorithm and PSO. Compared to other optimization algorithms, the bat algorithm provides quick convergence for the given objective function. It switches from exploration to exploitation quickly in the initial stage itself. Bat algorithm and PSO are selected for the proposed approach because of the distinct characteristics in obtaining an optimal solution for optimization problems. These characteristics are used to identify the optimal resource from the multi-cloud environment.

Further, the fitness function of the bat algorithm is improved using PSO in the optimal resource allocation process. The resources are selected based on resource status, distance, bandwidth, and task requirements. Allocating resources based on these parameters will greatly reduce the allocation time and deadline missed tasks. The novelty of this research work is present in the resource allocation procedure and the way of using the optimization model for the given problem. Moreover, the conventional optimization model traps in local optima solution, which reduces the performance of the entire system. The presented hybrid optimization provides an optimization solution for the multi-objective cloud environment, which differs the presented model from conventional optimization models. These parameters are related to the characteristics of the optimization model in the resource identification process.

Figure 3 depicts a simple overview of the proposed resource allocation model. The process starts with resource requests for the scheduled tasks. Based on the task resource requirements, suitable resources are selected as optimal resources using a hybrid optimization algorithm. Since it is a multi-cloud environment, resources for the task will be available in multiple cloudlets. So, all the available resources that satisfy the task request are selected from multiple clouds. From the selected resources, the most optimal resource is allocated based on the solution obtained from the optimization model that considers the distance, bandwidth, and energy parameters.

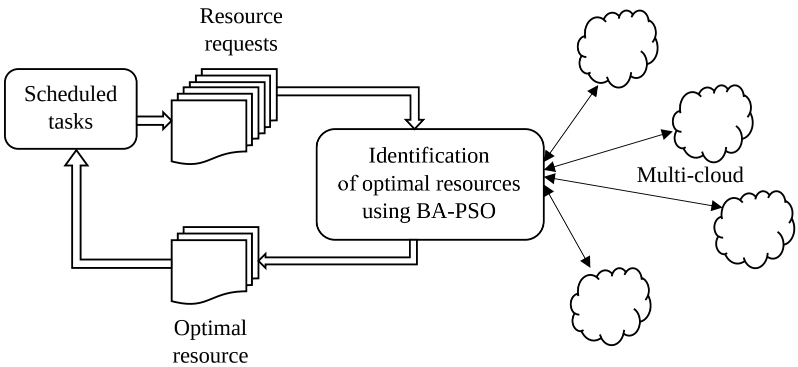

Fig. 3. Overview of the proposed resource allocation model.

For the system model, a simple multi-cloud architecture is considered, which has N number of clouds. The selected multi-cloud environment has public and private clouds. Consider that each cloud is allocated with M number of different resources. The diversity of resources is defined as M_x where x represents the different resources that vary from 1 to k in all the clouds. The necessary objective function is to identify optimal M_x for the given resource request. To identify the optimal resource, a hybrid bat algorithm particle swarm optimization (BAPSO) algorithm is employed.

Bat algorithm is a nature-inspired optimization model based on the echolocation characteristics of bats. Bats typically use a type of sonar called echolocation to detect prey, locate the obstacles in the path and roosting crevices. The echoes produced by the bats are loud and the variation in pulses is related to the hunting strategies. Mostly, frequency modulated signals are produced, which return back to the bat in octave shape. The frequency range is about 25 kHz to 150 kHz, and the typical pulse count is 10 to 20 per second and it can increase up to 200 pulses per second. The hunting nature of bats using echolocation is related to the optimal resource identification from a multi-cloud environment. The general rules followed while formulating the bat algorithm are (1) bats know the echo difference between prey and other background obstacles, (2) bats automatically adjust the emitted pulse frequency and pulse emission rate based on the target, and (3) the loudness varies from large positive to a constant minimum. These common rules are related to the optimal feature selection process in which the difference between essential resource and other resources are well known for the allocation model. Next, the resource identification selects the optimal features by adjusting the system parameters, and finally, the maximum optimal value to minimum optimal value is considered for the further classification process.

The bat motion is mathematically formulated to define the optimization process. Initially, the bat velocity (v_x^t), location (M_x^t), frequency (f_x) and solution space (d) for iteration t are considered. The best solution present among the bats is given as M_* based on the defined functions. The above general rules are formulated into mathematical functions as follows:

$$f_x = f_{\min} + (f_{\max} - f_{\min})\,\varphi, \tag{1}$$

$$v_x^t = v_x^{t-1} + \left(M_x^{t-1} - M_*\right) f_x, \tag{2}$$

$$M_x^t = M_x^{t-1} + v_x^t, \tag{3}$$

where φ represents the random vector function obtained from the uniform distribution. The range of φ is defined as $[0, 1]$. Initially, all the bats are assigned with a random frequency, in the range $[f_{\max}, f_{\min}]$. Due to this reason, the bat algorithm is defined as a frequency tuning algorithm that attains better exploration and exploitation features. Further, the loud variations and pulse rates are formulated to switch the system from exploration to exploitation stage. In this stage, the variations in the loudness (l_x) and the pulse emission rate (r_x) are varied in the iteration process. The loudness will generally decrease if the bat identifies the prey while the pulse emission rate is increased. The loudness can be selected between $l_{\max}$ and $l_{\min}$. Generally, the minimum loudness is assumed to be $l_{\min} = 0$. In the scenario where the bat identifies the prey and stops the pulse emission, and it is expressed mathematically as:

$$l_x^{t+1} = \rho l_x^t, \tag{4}$$

$$r_x^{t+1} = r_x^0(1 - e^{-\sigma t}), \tag{5}$$

where ρ and σ are constants and r_x^0 represents the initial pulse rate. For any $0 < \rho < 1$ and $\sigma > 0$ the above functions are changed into

$$l_x^t = 0, \tag{6}$$

$$r_x^t = r_x^0. \tag{7}$$

In this stage, the ρ is selected in the range 0.9 to 0.98. Now to improve the detection performance of the optimization model, a PSO is included. The PSO is a stochastic technique that is formulated based on the birds flocking characteristics in the food searching process. The random population is initiated as particles carry information about the search space. The other particles in the model perform the same so the information is exchanged with other particles. The best solution in the PSO is termed as global best and all the remaining particles need to move from the current position to the optimal position. Based on the best solution, the trajectory movement will be decided and the process repeats to obtain the best solution. Velocity values from current and previous are held by the particles in the population so that the next best solution can be obtained [27]. The velocity, each particle position vector is mathematically expressed as:

$$M_{xy}^{t+1} = M_{xy}^t + v_{xy}^{t+1}, \tag{8}$$

where M_{xy}^t is the position vector of an x-th particle at iteration t and v_{xy}^{t+1} is the velocity vector of an i-th particle at iteration $t+1$. The velocity is given as:

$$v_{xy}^{t+1} = \omega v_{xy}^t + c_1 r_1 \left(pbest_{xy}^t - M_{xy}^t\right) + \left(c_2 r_2 gbest_{xy}^t - M_{xy}^t\right), \tag{9}$$

where v_{xy}^{t+1} is the position vector of an x-th particle at iteration $t+1$, c_1 and c_2 represent the coefficients, and r_1 and r_2 represent the random numbers in the range $[0,1]$ for the uniformly distributed system. The particle fitness value is defined as *pbest* and *gbest* which is based on the fitness function of each particle. Based on the fitness function, each particle position is evaluated and it is expressed as:

$$f = \left(M_{xy}^t - v_{\max}\right)^2 + \left(M_{xy}^t - x_{\max}\right)^2. \tag{10}$$

Each particle fitness function is compared to update the *pbest* and *gbest* positions, and if the present position is comparatively better than the previous position, then the present position is considered as the best value and the overall fitness function is also updated based on that. Mathematically, it is formulated as:

$$pbest_{xy}^t = \begin{cases} M_{xy}^t & \text{if} \quad f(M_{xy}^t) < pbest_{xy}^t, \\ pbest_{xy}^t & \text{otherwise,} \end{cases}, \tag{11}$$

$$gbest_{xy}^t = \min\left(pbest_x^t, pbest_{x+1}^t, ..., pbest_s^t\right). \tag{12}$$

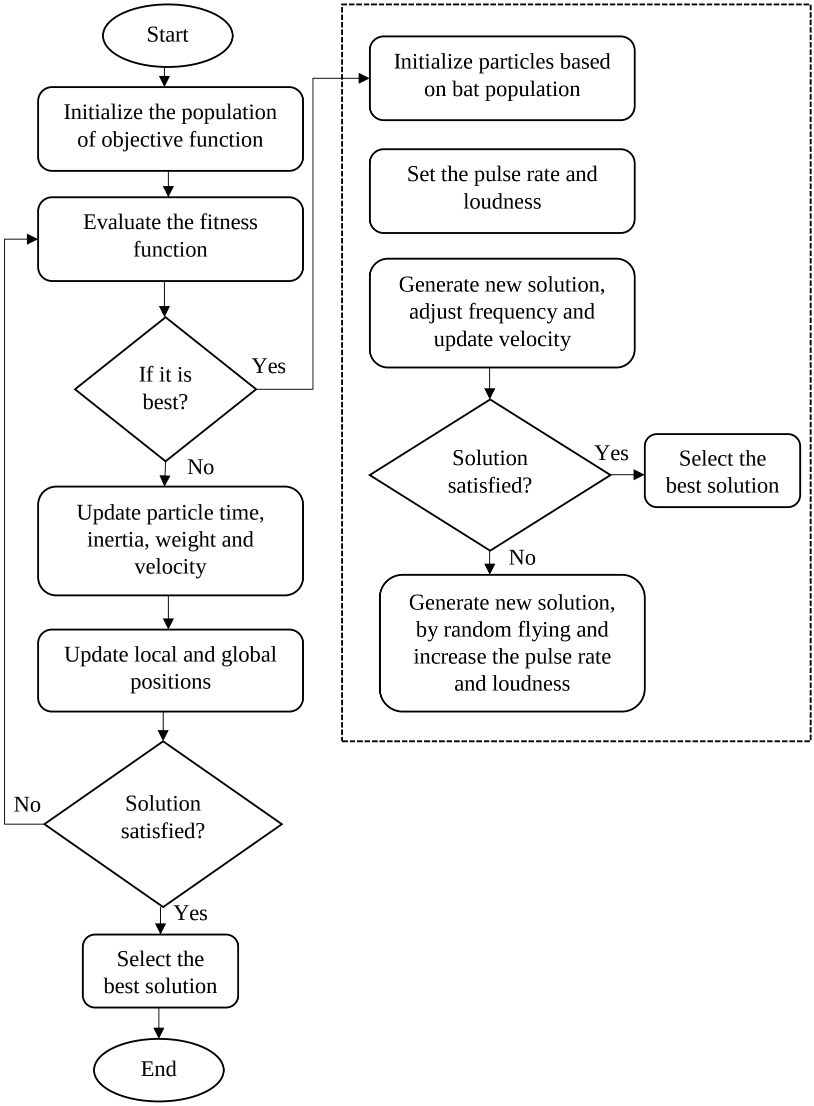

FIG. 4. Process flow of the proposed hybrid optimization model.

To update the position and velocity of all the particles above equations are used and the same procedure is implemented to tune the fitness function of the BOA. The final velocity of bats using PSO is expressed as:

$$v_{xy}^{t+1} = \omega v_{xy}^{t} + \left(c_1 r_1 \left(pbest_{xy}^{t} - M_{xy}^{t}\right) + \left(c_2 r_2 gbest_{xy}^{t} - M_{xy}^{t}\right)\right) f_x. \tag{13}$$

The summarized pseudocode for the proposed hybrid optimization-based resource allocation is given as follows.

Pseudocode for the proposed hybrid optimization-based resource allocation

Initialize bats (b_i), *velocity* v_x^t *location* (M_x^t), *frequency* (f_x) *and solution space* (d) *and particles* (p_i)
Input: b_i, p_i *for* $i = 1, 2, 3, ..., k$
Output: $gbest, v_{xy}^{t+1}$
Begin
Assign f_x *for* b_i
Initiate loudness (l_x), *pulse emission rate* (r_x)
If $(l_{\min} = 0)$

$$l_x^{t+1} = \rho l_x^t$$

$$r_x^{t+1} = r_x^0 \left(1 - e^{-\sigma t}\right)$$

Else
Initiate particle parameters and obtain solution for all the candidates
$pbest = gbest$
While $(i < i_{\max})$
Update the frequency randomly as $f_x = f_{\min} + (f_{\max} - f_{\min}) \varphi$
Update the velocity with new frequency

$$v_{xy}^{t+1} = \omega v_{xy}^t + \left(c_1 r_1 \left(pbest_{xy}^t - M_{xy}^t\right) + \left(c_2 r_2 gbest_{xy}^t - M_{xy}^t\right)\right) f_x$$

Obtain the new position $M_x^t = M_x^{t-1} + v_x^t$
If $(r_1 > r_x)$
Select the solution
Else if $(r_1 < l_x)$ *and* $f(M_x) < f(M_*)$
Update the new solution
Increase r_x *and* l_x
End if
End
Calculate the best sources
End while
End

4. Result and discussion

The proposed hybrid optimization model for the multi-cloud resource allocation scheme is implemented in the CloudSim simulator. Table 1 depicts the simulation parameter settings used in the proposed work analysis. The analysis is performed with a single data center having 16 clouds. The total number of hosts is 40 and a maximum of 5 hosts to a minimum 1 host is placed in the cloud. The resource capacity of each cloud is considered as different and the physical host is also different for each cloud to replicate the actual multi-cloud environment. For the proposed hybrid optimization algorithm, the simulation parameters are depicted in Table 2. To validate the performance of the proposed multi-cloud re-

Table 1. Simulation parameters.

Parameter	Value
Total number of clouds	16
Total number of hosts	40
VM speed	200–2000 MIPS
Memory	500–1 TB
Input task length	3000–5000
Hypervisor	Xen

Table 2. Optimization model parameters.

Parameter	Value
Bat size	15
Acceleration constants	1.04
Maximum number of iterations	150
Frequency, loudness, pulse rate (minimum)	0
Frequency, loudness, pulse rate (maximum)	3, 1, 2
Loudness constant	0.96
Pulse rate constant	0.9

source allocation model, conventional BOA and PSO model are compared with the proposed BAPSO model.

Figure 5 depicts the comparative analysis of the proposed BAPSO model and conventional PSO and BOA model performances in terms of the number

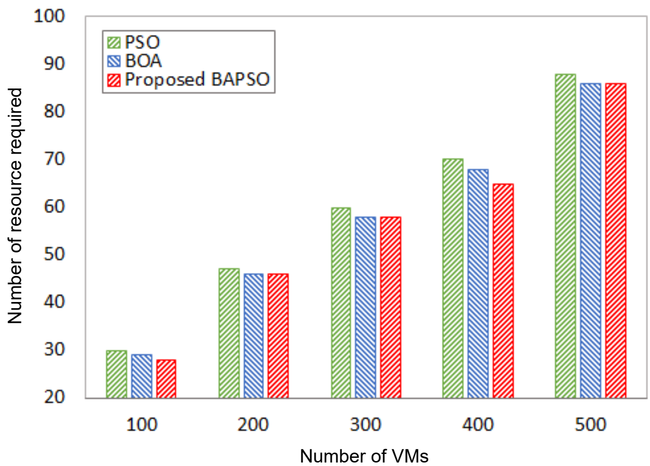

Fig. 5. Resource requirement for VMs.

of resources required to handle the request. It can be observed from the results that the proposed model uses a minimum number of resources to handle the request, whereas the PSO has maximum resource requirement for the request. The analysis is initially started with 100 virtual machines and it is gradually increased to 500. Throughout the process, the proposed model displays the best performance by selecting minimum resources for the given request. For a set of 500 tasks, the resource requirements by the proposed model are 86, whereas for the conventional PSO and BOA models, the resource requirements are 88 and 90, which indicates the significant improvement of the proposed model. The moderate performance of the conventional BOA model is enhanced by the proposed hybrid optimization model.

The total energy consumption of proposed optimization model and conventional optimization models is analyzed and depicted in Fig. 6. Results demonstrate that the minimum energy consumption of the proposed model is smaller compared to conventional models. As the proposed system uses fewer resources to handle the request, the energy consumption reduces greatly, whereas conventional model consumes more resources, which increases the energy consumption. The optimal resource selection procedure in the proposed hybrid optimization model selects the resources considering the distance, bandwidth, energy and task requirements, which introduces a huge impact in energy analysis and resource requirement analysis. It is observed for a set of 500 tasks that the energy consumed by the proposed hybrid optimization is 200 kWh, whereas the conventional optimization models PSO and BOA consume 240 kWh and 230 kWh energy, respectively.

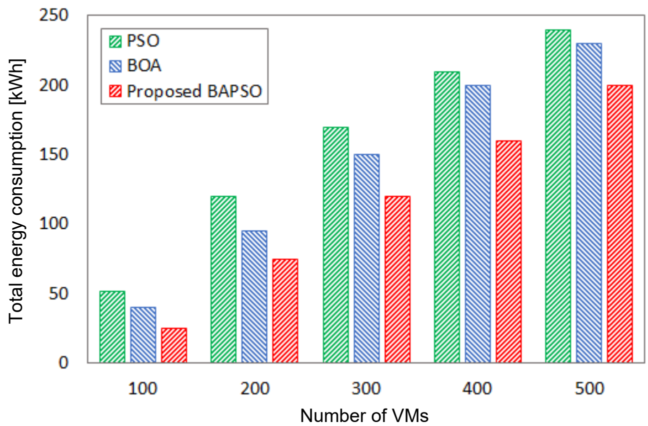

Fig. 6. Total energy consumption.

Figure 7 depicts the comparative analysis of optimization model in terms of deadline missed tasks. Deadline missed tasks is one of the important SLA

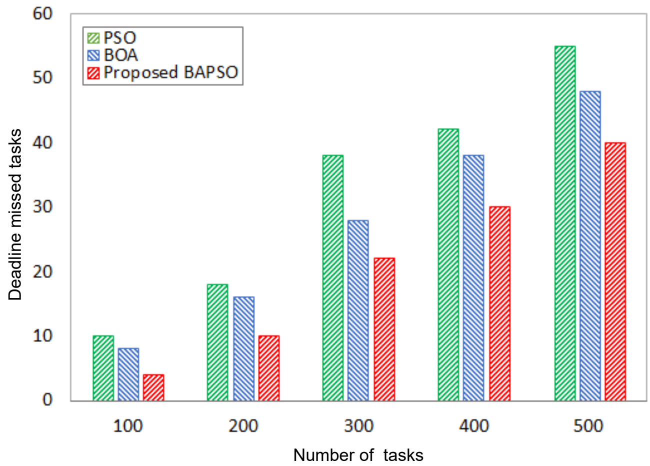

Fig. 7. Deadline missed tasks.

violations in cloud services. The analysis clearly depicts that maximum missed tasks are observed in the PSO-based resource allocation. Whereas the BOA performs better with fewer deadline missed tasks compared to the PSO. However, its performance is still not as good as the proposed hybrid optimization algorithm. Even for 500 tasks, the deadline missed in the proposed model is 40 tasks, whereas in BOA 48 tasks and 55 tasks in the PSO model.

Resource allocation time comparison of the proposed hybrid optimization model and conventional optimization models is depicted in Fig. 8. It is evident from the results that the minimum resource allocation time is attained by the proposed approach due to the optimal feature selection procedure. For a set of 500 tasks, the allocation time attained by the proposed model is 47 s, whereas for the conventional optimization models of PSO and BOA it is 74 s and 64 s,

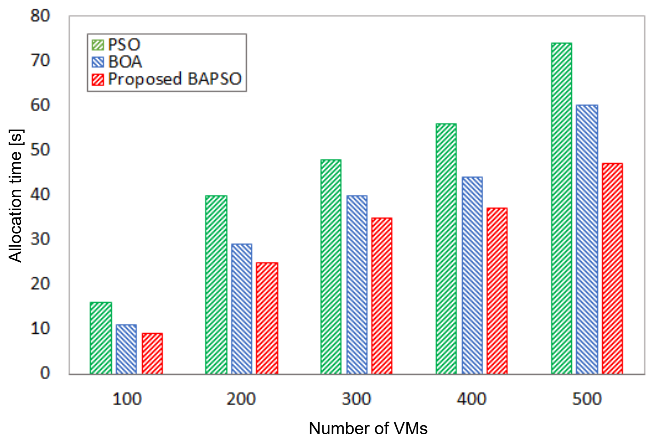

Fig. 8. Resource allocation time.

respectively, which is much higher than the proposed approach. To validate the performance of the proposed hybrid optimization model, a comparative analysis is presented in Table 3. Optimization models such as genetic algorithm (GA), ant colony optimization (ACO), and genetic algorithm with random forest (GA-RF) are compared with the proposed hybrid optimization model. The results are presented for a set of 500 VMs and its resource requests. The parameters considered for comparative analysis are energy consumption and allocation time.

Table 3. Performance comparative analysis.

Algorithm	Energy consumption [kWh]	Execution time [sec]
Genetic algorithm [23]	380	110
ACO [23]	460	135
GA-RF [23]	310	90
Proposed BAPSO	200	47.22

The hybrid bat optimization and PSO attain improved feature selection characteristics due to the global solution and fitness functions. At the same time, the conventional model relies either on fitness or global solution, which increases the resource allocation time. From the analysis, it can be confirmed that resource allocation in a multi-cloud environment through a hybrid optimization algorithm will provide enhanced performance compared to the conventional cloud computing environment.

5. Conclusion

A hybrid optimized resource allocation model for a multi-cloud environment was presented in this research work. Combining the optimization characteristics of the BOA and PSO, a hybrid optimization was introduced for efficient resource allocation. The optimal resources in a multi-cloud environment were identified by the hybrid model considering the distance, bandwidth, energy, and task requirements before allocating the resources to the task. The performance of the proposed model was evaluated in terms of resource requirements, energy consumption, SLA violations, and allocation time. The conventional PSO and bat optimization were compared with the proposed model. Results demonstrate that the proposed hybrid optimization model outperforms for all the parameters compared to conventional optimization models. Despite the fact that the performance of the proposed model is better, the number of deadline missed tasks must be reduced further to attain better QoS. Further, this research work can be extended by introducing hybrid deep learning approaches for improved performances.

References

1. J.S. Manoharan, A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits, *Journal of Innovative Image Processing (JIIP)*, **3**(1): 36–51, 2021, doi: 10.36548/JIIP.2021.1.004.
2. S. Shakya, An efficient security framework for data migration in a cloud computing environment, *Journal of Artificial Intelligence*, **1**(1): 45–53, 2019, doi: 10.36548/jaicn.2019.1.006.
3. J. Son, R. Buyya, Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds, *IEEE Transactions on Sustainable Computing*, **4**(1): 17–28, 2019, doi: 10.1109/TSUSC.2018.2842074.
4. M.H. Siddiqi, M. Alruwaili, A. Ali, S.F. Haider, F. Ali, M. Iqbal, Dynamic priority-based efficient resource allocation and computing framework for vehicular multimedia cloud computing, *IEEE Access*, **8**: 81080–81089, 2020, doi: 10.1109/ACCESS.2020.2990915.
5. L. Wei, C.H. Foh, B. He, J. Cai, Towards efficient resource allocation for heterogeneous workloads in IaaS clouds, *IEEE Transactions on Cloud Computing*, **6**(1): 264–275, 2018, doi: 10.1109/TCC.2015.2481400.
6. S. Tang, B.-S. Lee, B. He, Fair resource allocation for data-intensive computing in the cloud, *IEEE Transactions on Services Computing*, **11**(1): 20–33, 2018, doi: 10.1109/TSC.2016.2531698.
7. P. Poullie, T. Bocek, B. Stiller, A survey of the state-of-the-art in fair multi-resource allocations for data centers, *IEEE Transactions on Network and Service Management*, **15**(1): 169–183, 2018, doi: 10.1109/TNSM.2017.2743066.
8. S. Gong, B. Yin, Z. Zheng, K.-Y. Cai, Adaptive multivariable control for multiple resource allocation of service-based systems in cloud computing, *IEEE Access*, **7**: 13817–13831, 2019, doi: 10.1109/ACCESS.2019.2894188.
9. L.-D. Chou, H.-F. Chen, F.-H. Tseng, H.-C. Chao, Y.-J. Chang, DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization, *IEEE Systems Journal*, **12**(2): 1554–1565, 2018, doi: 10.1109/JSYST.2016.2596299.
10. G. Peng, H. Wang, J. Dong, H. Zhang, Knowledge-based resource allocation for collaborative simulation development in a multi-tenant cloud computing environment, *IEEE Transactions on Services Computing*, **11**(2): 306–317, 2018, doi: 10.1109/TSC.2016.2518161.
11. X. Gao, R. Liu, A. Kaushik, Hierarchical multi-agent optimization for resource allocation in cloud computing, *IEEE Transactions on Parallel and Distributed Systems*, **32**(3): 692–707, 2021, doi: 10.1109/TPDS.2020.3030920.
12. F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, V.C.M. Leung, Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm, *IEEE Systems Journal*, **12**(2): 1688–1699, 2018, doi: 10.1109/JSYST.2017.2722476.
13. K.M. Sim, Agent-based approaches for intelligent intercloud resource allocation, *IEEE Transactions on Cloud Computing*, **7**(2): 442–455, 2019, doi: 10.1109/TCC.2016.2628375.
14. K. Chard, K. Bubendorfer, Co-operative resource allocation: building an open cloud market using shared infrastructure, *IEEE Transactions on Cloud Computing*, **7**(1): 183–195, 2019, doi: 10.1109/TCC.2016.2594174.

15. A. Alsarhan, A. Itradat, A.Y. Al-Dubai, A.Y. Zomaya, G. Min, Adaptive resource allocation and provisioning in multi-service cloud environments, *IEEE Transactions on Parallel and Distributed Systems*, **29**(1): 31–42, 2018, doi: 10.1109/TPDS.2017.2748578.

16. T. Thein, M.M. Myo, S. Parvin, A. Gawanmeh, Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers, *Journal of King Saud University – Computer and Information Sciences*, **32**(10): 1127–1139, 2020, doi: 10.1016/j.jksuci.2018.11.005.

17. P. Haratian, F. Safi-Esfahani, L. Salimian, A. Nabiollahi, An adaptive and fuzzy resource management approach in cloud computing, *IEEE Transactions on Cloud Computing*, **7**(4): 907–920, doi: 10.1109/TCC.2017.2735406.

18. S.G. Domanal, R.M.R. Guddeti, R. Buyya, A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment, *IEEE Transactions on Services Computing*, **13**(1): 3–15, 2020, doi: 10.1109/TSC.2017.2679738.

19. S.K. Mishra *et al.*, Energy-aware task allocation for multi-cloud networks, *IEEE Access*, **8**: 178825–178834, 2020, doi: 10.1109/ACCESS.2020.3026875.

20. M. Farid, R. Latip, M. Hussin, N.A.W. Abdul Hamid, Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment, *IEEE Access*, **8**: 24309–24322, 2020, doi: 10.1109/ACCESS.2020.2970475.

21. T. Shi, H. Ma, G. Chen, S. Hartmann, Location-aware and budget-constrained service deployment for composite applications in multi-cloud environment, *IEEE Transactions on Parallel and Distributed Systems*, **31**(8): 1954–1969, 2020.

22. H. Li, H. Xu, C. Zhou, X. Lü, Z. Han, Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment, *IEEE Transactions on Vehicular Technology*, **69**(9): 10214–10226, 2020, doi: 10.1109/TVT.2020.3003898.

23. Madhusudhan H.S., Satish Kumar T., S.M.F.D. Syed Mustapha, P. Gupta, R.P. Tripathi, Hybrid approach for resource allocation in cloud infrastructure using random forest and genetic algorithm, *Scientific Programming*, **2021**: Article ID 4924708, 10 pages, 2021, doi: 10.1155/2021/4924708.

24. K. Kottursamy, A review on finding efficient approach to detect customer emotion analysis using deep learning analysis, *Journal of Trends in Computer Science and Smart Technology*, **3**(2): 95–113, 2021.

25. H.K. Andi, Analysis of serverless computing techniques in cloud software framework, *Journal of IoT in Social, Mobile, Analytics, and Cloud*, **3**(3): 221–234, doi: 10.36548/jismac.2021.3.004.

26. Y.B. Hamdan, A. Sathesh, Construction of efficient smart voting machine with liveness detection module, *Journal of Innovative Image Processing*, **3**: 255–268, 2021, doi: 10.36548/jiip.2021.3.007.

27. S. Senthilkumar, V. Mohan, S.P. Mangaiyarkarasi, M. Karthikeyan, Analysis of dingle-diode PV model and optimized MPPT model for different environmental conditions, *International Transactions on Electrical Energy Systems*, **2022**: Article ID 4980843, 17 pages, 2022, doi: 10.1155/2022/4980843.

Received October 5, 2021; revised version January 11, 2022; accepted February 3, 2022.