

Enhancement of Task Scheduling Performance with Dependency Aware Clustering Model Using Bioinspired Optimizations in Multicloud

Bharti Vijay Nikose

Department of Computer Engineering,
Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, Maharashtra, India.
E-mail: bharti.nikose@rait.ac.in
ORCID: <https://orcid.org/0009-0002-6263-3316>

Gautam Murlidhar Borkar

Department of Information Technology,
Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, Maharashtra, India.
Corresponding author: gautam.borkar@rait.ac.in
ORCID: <https://orcid.org/0000-0002-6339-5295>

(Received on March 5, 2026; Revised on June 22, 2026 & July 15, 2026; Accepted on August 9, 2026)

Abstract

Task scheduling in clouds is a multimodal task that involves integration of task-requirement analysis, Virtual Machine (VM) capacity analysis, dependency analysis, deadline analysis, and compatibility evaluation between task & VMs. Efficient task-scheduling models aim at maximizing cloud utilization, while maintaining higher deadline hit ratio, lower cloud effort, and higher task diversity levels. Most of the existing task scheduling models either, do not consider inter-task dependency while scheduling, or are highly complex which limits their scalability performance when applied to large-scale task sets. To overcome these issues, this study proposes design of a dependency-aware clustering model for improving task-scheduling performance via bioinspired optimizations. The proposed model initially scans input tasks and uses a Service Level Agreement for dependency resolutions. These dependency-resolved tasks are clustered on the basis of a score metric that combines make-span and deadline requirements. Clustering is facilitated via a combination of K-Means, Fuzzy C-Means, and Hierarchical Clustering techniques. The clusters along with VM configurations are processed by a Crow Search Optimization Model, which assists in mapping VMs to relevant tasks. The mapping process is facilitated via a deadline-aware & task-capacity-aware fitness function, which assists in improving its scheduling performance under heterogeneous task types. It is observed that the proposed model is showing improvement in achieving less scheduling delay, higher deadline hit ratio, increase in scheduling efficiency and reduction in energy required for scheduling tasks.

Keywords- Task scheduling, Crow search optimization, Service level agreement, Multicloud.

1. Introduction

A network-based platform that provides instantaneous access to IT and computing resources on demand is known as "the cloud" (Xu and Palanisamy, 2021). Both cloud service providers and their clients consider cloud computing as a viable and effective strategy at the present time. Customers are drawn to solutions that provide cheap service costs, great scalability, and high availability so that they may deploy computationally and power-intensive applications without breaking the bank (Hosseinalipour and Dai, 2021). Companies providing cloud computing services would do well to consider major profit opportunities for Cloud Service Providers (CSP). Google Apps Engine, Amazon Web Services, and Windows Azure are well-known commercial cloud computing services that schedule tasks using the Multi-Objectives Symbiotic Organism Search algorithm. It is estimated that by 2020, American data centers would use 139 billion kilowatt hours of power at a cost of \$13.7 billion per year up from 91 billion kilowatt hours in 2013 (Belgacem et al., 2022).

Managing resources on a cloud platform helps service providers to save costs by taking advantage of economy of scale. However, modern cloud service providers have increasingly adopted Reinforcement Learning (RL) and related optimization techniques to address the adverse effects of low server resource utilization on the power efficiency of large-scale data centers (Mireslami et al., 2021). When fewer than half of a computer's processing power and memory are being utilized, power efficiency suffers. Similar work in paper Middya et al. (2022) highlights two primary features of data center power use: (i) Most servers should be used at a power efficiency level of between 70% and 80%, (ii) servers use a lot of electricity even when they are not actively processing data. As a result, the Resource Utilization Rate (RUtR) of data center must be taken into consideration by resource-efficient cluster management to guarantee continuous service without sacrificing quality. Quantity of Service is defined in this study Fan et al. (2021) by three metrics: (i) The percentage of rejected tasks, (ii) the percentage of missed deadlines, and (iii) the percentage of missed incentives which gauges the frequency with which the user-specified task priority criteria is met. Large-scale data centers require efficient cloud resource management and task scheduling to ensure optimal utilization of computing resources. By improving resource efficiency and reducing operational overhead, CSP can achieve higher profit margins. A variety of resource management and scheduling difficulties have surfaced in unison with the rapid growth of the cloud computing sector (Tan et al., 2022).

Scalability is now a primary concern when providing support for large-scale data centers that have been developed worldwide. Self-learning techniques are therefore essential for handling the ambiguity and volatility of user expectations. Demands for resources, such as CPU and memory, are likely to fluctuate over time and exhibit a range of trends (Kim et al., 2022a). A new paradigm in cloud computing known as intelligent cloud resource management has emerged as a primary means of addressing these issues. Research into cloud resource management is on the rise, with several methods having been presented to achieve different goals (Chen et al., 2022a). Increasing the resource efficiency of large-scale data center may improve their capabilities and cost-effectiveness, as suggested by the authors of paper Alam et al. (2021). Various solutions proposed which involves a Quality of service (QoS)-aware, resource-efficient cluster management system for centralized resource allocation and assignment. The model explained here in this paper Du et al. (2022), where the users of a cloud can indicate complementarities and substitutability among those resources by submitting their demands for virtual resources as bids using the given bidding language.

The resource pricing and demand allocation strategy aimed at revenue maximization also meets critical features such as budget feasibility, incentive compatibility, and fair divisions. A technique for supplying cloud resources that utilizes both on-demand and reserved resources is described by researchers Gao et al. (2021a). Resource allocation is described as an infinite knapsack problem, whereas resource scaling is modelled as a one-step forward resource allocation issue. To reduce the likelihood of Service Level Agreement (SLA) violations and performance deterioration from under loaded or overloaded servers, an online resource prediction system is created and implemented at each Virtual Machine (VM) (Zhang et al., 2022). A study on market-based resource allocation of distributed cloud computing services found that the proposed methodology can increase cloud computing service revenue and reduce energy consumption expenses through case studies on a 55-bus electrical system. To improve energy cost-effectiveness, the authors of a particular investigation used a realistic electric price model; nevertheless, their approach has scalability problems, and their offline algorithms are unable to handle the large number of inputs (Chen et al., 2022b). Deep Reinforcement Learning (DRL) techniques were used in a study to partially address the resource allocation problem, which may facilitate scaling. Solution techniques that consider the maximum power, resource block allocation, fronthaul capacity refers to the maximum amount of data that can be transmitted between the Centralized Processing Unit (CPU) and Remote Radio Units (RRUs). The quality

of service constraints of macro cell customers is constructed using the Dinkelbach algorithm which solves fractional optimization problem, Lagrange multiplier, and Alternate Direction Method of Multipliers (ADMM) methodologies. In order to guarantee proper execution of concurrent processes in the cloud, their approach does not plan data-dependent activities (Najm et al., 2021).

He et al. (2022) proposed a deep reinforcement learning QoS aware work scheduling system in which a predicted job response time is supplied as the QoS requirement, with the goal of decreasing the average task response time. While executing their job, however, they pay little attention to energy costs. In order to save energy costs in real life, these studies do not employ fine-grained online scheduling, which involves dynamically dividing each activity into a precise number of minutes (Li et al., 2021). In this research, the author examines the dynamic resource allocation problem for the deployment of Virtual Network Functions (VNFs) in satellite edge clouds. They begin by developing system models to examine the delay and energy consumption caused by various computing modes, and then formalize the combined problem as a mixed-integer optimum choice problem (Gamsiz and Özer, 2021). Virtual Machine cannot handle heavy workloads due to shortcomings in its high computational complexity, energy consumption, and memory requirements. The main component of this research is service level agreement. The CSP shall assess the services provided in a timely manner to determine the probability of any violations to support the proactive technique. Categorizing SLA violations enhances the reliability of QoS estimation. In the event of a predicted SLA violation, QoS expectations are dynamically revised during the SLA renegotiation cycle. In paper Gao et al. (2021b), the authors introduced a distributed scheduler that takes admission control and scheduling quality or performance goals into account via Genetic Algorithm (GA) based Multi Agent Optimization (MAO) process. In addition, its security regulations are also thought to be combined with encrypted texts to guarantee system security and avoid violating the privacy of data owners when task is transferred between multiclouds (Lattuada et al., 2022). Machine learning in parallel presents a number of challenges and opportunities, including data separation, communication, and scalability. A major problem in parallel machine learning is partitioning the data into smaller subsets, or "partitions," that may be processed independently by different processors or computers. It is challenging to partition the data so that each processor may work independently and still be able to combine their findings to get a global answer. A cloud architecture with faster processing speed and a greater learning convergence rate can somewhat increase the accuracy of intelligent fault classification (Chhetri et al., 2021).

This research presents a dependency-aware clustering methodology to address these problems and enhance the efficiency with which tasks are scheduled by drawing inspiration from biological systems. The efficiency of the suggested strategy was compared to traditional scheduling techniques in order to assess its efficacy. The paper concludes with some observational thoughts on the suggested paradigm and suggestions for improving its functionality.

2. Literature Review

Saxena et al. (2022) proposed the OP-MLB framework for efficient cloud resource management using online resource prediction at each VM to minimize SLA violations and performance issues. Their multi-objective VM placement and migration algorithms reduce power usage and network traffic. Tested on real-world datasets, the framework achieves up to 85.3% power savings over Best-Fit, improving load balancing, reducing active servers, and enhancing overall data center efficiency.

Tao et al. (2022) presented a market-based approach for computing resource allocation in distributed cloud services. It introduces Batch Workload Scheduling (BWS)-based and Thermal Inertia (TI)-based Virtual Energy Storage System (VESS) models to enhance resource aggregation and energy flexibility in virtual IDCs. The method also models energy trading in a transactive market. Case studies on a 55-bus electricity

system show reduced energy costs and increased revenue from cloud service delivery using the proposed technique.

Liu et al. (2022a) proposed a pricing method for optimal resource allocation in MEC systems based on Smart Mobile Device (SMD) budgets. It introduces a MEC resource market model enabling resource exchange between providers (APs and edge clouds) and consumers (SMDs). Using microeconomic theory, it determines optimal budget allocation for maximizing SMD utility. An Equilibrium Price Finding (EPF) technique is also presented to identify system equilibrium prices, enhancing efficiency and overall utility in resource allocation.

Zafari et al. (2021) models resource sharing as a multi-objective optimization problem and introduces a Cooperative Game Theory-based framework (CGT). Service providers prioritize native applications and share surplus resources with others. The game's convexity and monotonic utility ensure a non-empty core and stable grand coalition. Two core-based allocation methods Game-theoretic Pareto Optimal Allocation (GPOA) and Polyandrous-Polygamous Matching Pareto Optimal Allocation (PPMPOA) are proposed, ensuring Pareto-optimal outcomes and stability among cooperating service providers.

Misra et al. (2022) studied the topic of assuring profitability for numerous sensor-owners in sensor-cloud while meeting end-user service needs is investigated in this study. Existing studies offered a few node allocation strategies that do not take into account the cost spent by sensor owners or the Quality of Information (QoI) of sensed-data in sensor-cloud. To overcome this issue, RACE, a strategic resource allocation method, is presented, which involves sensor owners in the node allocation process.

Ferdouse et al. (2021) studied an underlaid strategy for resource allocation for small and macro cell users in this research to increase Energy Efficiency (EE) in H-CRANs. A two-step energy efficient underlaid cellular IoT (UC-IoT) supported H-CRAN technique is presented and tested in comparison to overlay cellular IoT (OC-IoT) supported H-CRAN and underlaid H-CRAN without cellular IoT devices. The suggested technique is assessed in terms of energy efficiency and the Jain's fairness index, taking into account the influence of the number of cellular IoT density in each H-CRAN small cell. The simulation results show that the suggested strategy outperforms prior approaches.

Fang et al. (2022) presented a resource allocation system based on Deep Reinforcement Learning (DRL) to optimise content distribution in a layered Fog Radio Access Network (FRAN). The optimum resource allocation problem is formulated as a minimal delay model, with in-network caching deployed and the identical content requests from mobile users pooled in the queue of each base station. In order to deal with increasing user requests and overcome FRAN capacity constraints, our model employs a cloud-edge cooperation offloading scheme in which the integrated allocation of caching, computing, and communication resources, as well as joint optimization between in-network caching and routing, are considered to promote resource utilization and content delivery.

Kim et al. (2022a) introduced Cloud Insight, an innovative cloud workload prediction platform using an ensemble of multiple predictors. It dynamically assigns predictor weights via multi-class regression based on current workload accuracy. Tested on real-world workload traces, Cloud Insight outperforms existing methods by 13–27%. Simulations show it reduces under/over-provisioning periods by 15–20%, enhancing cost-efficiency and minimizing SLA violations, making it highly effective for accurate, adaptive cloud resource management.

Gao et al. (2022a) studied the dynamic resource allocation problem for Virtual Network Function (VNF) deployment in satellite edge clouds in this research. The goal is to reduce both the network bandwidth cost and the service end-to-end latency. To solve the VNF placement problem, we define it as an integer non-linear programming problem and present a distributed VNF placement (D-VNFP) technique. The tests are carried out to assess the performance of the proposed D-VNFP algorithm, with Viterbi and Game theory serving as the baseline algorithms. The findings indicate that the suggested D-VNFP method is effective and efficient for resolving the VNF placement problem in satellite edge clouds.

Liu et al. (2022b) presented an online task offloading and resource allocation strategy for edge-cloud orchestrated computing in this research, with the goal of minimizing task average latency over time. We begin by developing system models to examine the delay and energy consumption caused by various computing modes, and then formalize the combined problem as a mixed-integer optimum choice problem. The issue is then decomposed using Lyapunov optimization and duality theory into a series of subproblems that may be addressed semi-decentralized. We also formalize our approach's ability to reach near-optimal performance. Numerous simulations are being run to validate our approach's advantages.

Jawade et al. (2022) addressed task scheduling challenges in cloud computing, where VMs struggle with high energy use, computational complexity, and memory demands. Minimizing active host usage while maximizing resource utilization can also introduce security risks. To overcome these issues, a Greywolf Optimization technique with Deadline Constraints is proposed. It efficiently assigns tasks and resources within specified time limits, reducing execution time, improving resource utilization, and enhancing overall scheduling performance in cloud environments.

Kumar and Kumar (2021) introduced a conceptual Service Level Agreement (SLA) mechanism incorporating an SLA negotiation process to minimize violations in cloud computing environments. The methodology allows dynamic agreement adjustments between providers and clients, leading to improved service reliability and client satisfaction. However, the negotiation process adds overhead, potentially increasing response time and system complexity. The approach suits applications requiring flexible SLA terms, though scalability under high load remains a challenge.

Mayyahi and Seno (2022) proposed a security and privacy-aware computing approach for data sharing in cloud environments. Their method focuses on encrypting sensitive data and enforcing strict access control using secure computing protocols. The main advantage is enhanced data confidentiality and protection against unauthorized access. However, the increased security comes at the cost of higher computational load and processing delay. The model is effective but may impact performance in large-scale systems.

Salman et al. (2023) explored the use of parallel machine learning algorithms to enhance data processing in big data environments. By distributing tasks across multiple processors, they achieved faster computation and better scalability. The parallel approach is advantageous for handling large datasets efficiently. However, the methodology faces limitations like synchronization issues, load imbalance, and dependency on high-performance computing infrastructure. Despite these challenges, the model significantly reduces training time for complex learning tasks.

The existing techniques and their advantages and limitations are summarized in **Table 1**. The analysis of these studies highlight the need for dependency aware clustering approach that can improve task scheduling performance using crow search optimization.

Table 1. Literature review based on various techniques.

Sr. No.	Related work	Technique	Advantages	Issues
1.	Saxena et al. (2022)	Multi-objective load balancing with online prediction.	Reduces SLA violations while increasing power efficiency and resource utilization.	Does not emphasize trust and reliability in VM allocation.
2.	Tao et al. (2022)	Virtual Energy Storage Systems (VESS) for resource allocation.	Reduces energy costs and increases service revenues.	Security and privacy concerns not addressed.
3.	Liu et al. (2022a)	Equilibrium Price Finding (EPF) algorithm for MEC.	Maximizes system utility with optimal resource allocation.	Considers budget and resource limitations, but lacks broader analysis.
4.	Zafari et al. (2021)	Resource sharing in edge clouds based on cooperative game theory.	Enables efficient resource leasing among service providers.	Highlights resource flexibility but not overhead or coordination cost.
5.	Misra et al. (2022)	RACE: Strategic resource allocation in sensor cloud.	Enhances profits and service availability.	Needs analysis of service delays and security vulnerabilities.
6.	Ferdouse et al. (2021)	Allocating resources in H-CRANs with energy efficiency.	High energy efficiency and fairness in IoT-dense environments.	Ignores trade-off between throughput and energy efficiency.
7.	Fang et al. (2022)	DRL-based scheme for Fog Radio Access Networks (FRAN).	Fast convergence under heterogeneous networks.	Low latency is difficult to guarantee in complex networks.
8.	Kim et al. (2022b)	Cloud Insight – an online workload prediction framework.	Enhances SLA adherence and cost-efficiency.	Entirely online; lacks offline robustness.
9.	Gao et al. (2022b)	Algorithms for VNF placement using Viterbi and Path Selection.	Decreases end-to-end latency and bandwidth costs.	Risk of resource conflicts due to decentralized agent strategies.
10.	Liu et al. (2022b)	Lyapunov and duality-based semi-decentralized problem solving.	Minimizes task latency via cloud-edge collaboration.	Impact of edge computing in multi-cloud is not explored.
11.	Jawade et al. (2022)	Grey Wolf Optimizer with deadline restrictions.	Low encryption time, consistent throughput, & secure task scheduling.	Faces challenges in complexity, scalability, and resource overhead.
12.	Kumar and Kumar (2021)	SLA negotiation-based mechanism for minimizing SLA violations.	Enhances trust between users and providers; reduces SLA violations; improves service reliability.	Complex negotiation mechanisms may increase overhead and delay in service provisioning.
13.	Mayyahi and Seno (2022)	Security-aware and privacy-preserving model for secure data sharing.	Protects sensitive data; ensures secure sharing among multiple cloud parties.	May increase computational load; potential scalability issues with large datasets.
14.	Salman et al. (2023)	Use of parallelization techniques in machine learning algorithms to enhance performance.	Reduces processing time; improves performance for big data analytics.	High resource consumption; dependency on hardware and data distribution efficiency.

The above literature survey has been done to design the dependency aware clustering model which improves the performance of task through Crow Search Optimization which is explained more clearly in Section 3.

3. Design of a Dependency-Aware Clustering Model for improving Task-Scheduling Performance via Bioinspired Optimizations (DCMTSB)

A review of the existing literature indicates that most task scheduling techniques either neglect inter-task dependencies or involve computationally intensive procedures, thereby constraining their scalability and performance when processing extensive task workloads. To overcome these issues, this section discusses design of a dependency-aware clustering model for improving task-scheduling performance via bioinspired optimizations. Detailed algorithm of the model is explained below, where it can be observed that the proposed model initially scans input tasks, and uses a SLA for dependency resolutions. These dependency-resolved tasks are clustered based on a score metric that combines make-span and deadline requirements. Clustering is facilitated via either of K-Means, Fuzzy C-Means, and Hierarchical Clustering techniques. The clusters along with VM configurations are processed by a Crow Search Optimization (CSO) Model, which assists in mapping VMs to relevant tasks. The mapping process is facilitated via a deadline-aware &

task-capacity-aware fitness function, which helps in enhancing its scheduling performance under heterogeneous task types. The proposed dependency-aware clustering model addresses several challenges in cloud task scheduling:

- **Inter-Task Dependency Consideration:** Traditional task scheduling models often overlook inter-task dependencies, leading to suboptimal scheduling decisions. By incorporating a dependency resolution mechanism based on SLA, the proposed model ensures that task dependencies are adequately addressed, thereby improving the overall scheduling accuracy.
- **Scalability:** Many existing task scheduling models become impractical when dealing with large sets of tasks due to their complexity. The proposed model aims to mitigate this challenge by employing bioinspired optimizations, specifically the Crow Search Optimization model, which is known for its scalability and efficiency in handling large-scale optimization problems.
- **Deadline and Make-Span Optimization:** Efficient task scheduling requires balancing various objectives, such as minimizing make-span while meeting deadlines. The proposed model employs a score metric that combines make-span and deadline requirements to guide the clustering process, ensuring that tasks are grouped optimally based on their scheduling constraints.
- **Heterogeneous Task Types:** Cloud environments often host diverse workloads with varying resource requirements and execution characteristics. The proposed model addresses this challenge by incorporating a deadline-aware and task-capacity-aware fitness function, which considers the specific requirements of different task types during the VM-to-task mapping process. This approach enhances scheduling performance by effectively matching tasks to suitable VM configurations, thereby optimizing resource utilization and meeting diverse task requirements.

To tackle this, the method employs a dependency-aware clustering approach inspired by biological systems. This step involves analysing the dependencies between tasks, ensuring that related tasks are grouped together intelligently. By organizing tasks into clusters based on their dependencies, the method lays the groundwork for more efficient resource allocation and scheduling decisions. Once tasks are clustered, the method leverages advanced optimization techniques to maximize scheduling efficiency while meeting diverse performance objectives. Specifically, it utilizes bioinspired optimization algorithms, such as the Crow Search Optimization model, to dynamically adjust to changing resource demands and user expectations. This adaptive approach enables the method to scale effectively, even in large-scale data centers comprising thousands of servers, addressing the scalability challenge highlighted in the introduction. Furthermore, the method places a strong emphasis on SLA management, recognizing the importance of maintaining high service quality and customer satisfaction.

To achieve this, it implements proactive and continuous SLA renegotiation techniques, allowing the cloud service provider to assess service performance and anticipate potential breaches. By continuously monitoring and adjusting SLA parameters, the method reduces the risk of SLA violations and enhances overall service reliability. In summary, the proposed method for cloud task scheduling offers a comprehensive and innovative approach to addressing the challenges outlined in the introduction. By integrating advanced techniques such as dependency-aware clustering, bioinspired optimization, SLA management, and real-time usage considerations, the method demonstrates its effectiveness in optimizing resource allocation, enhancing scheduling efficiency, and meeting diverse performance objectives in cloud environments. The overall working of the model is presented with the help of **Figure 1**.

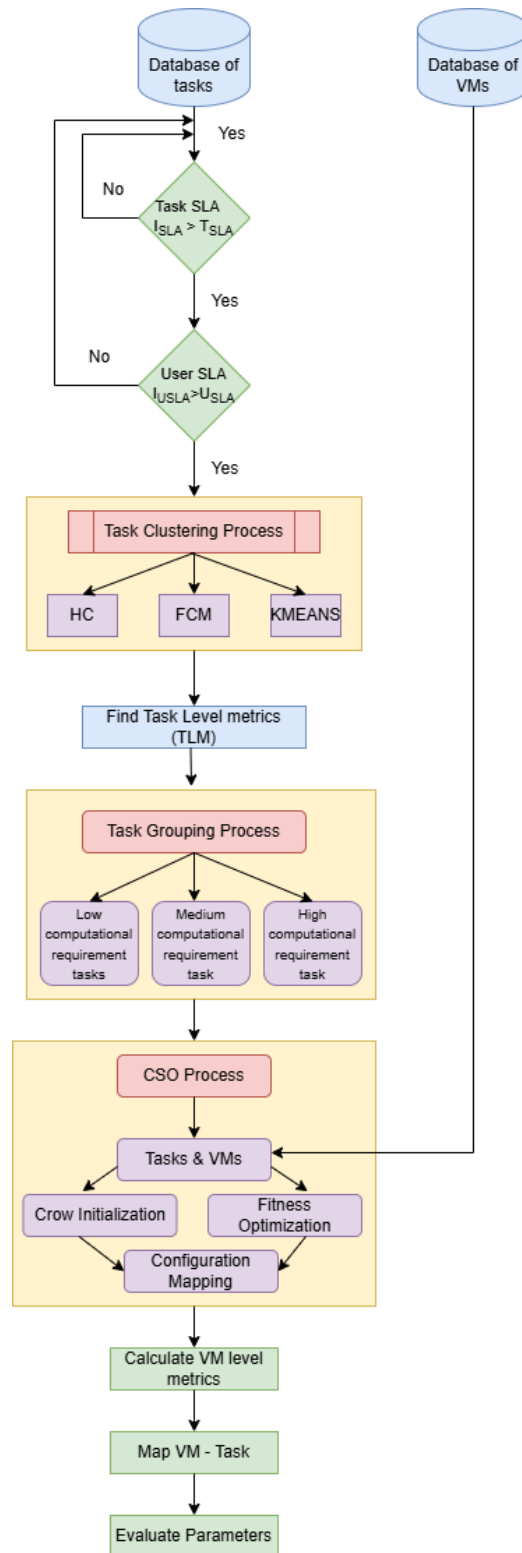


Figure 1. SLA-based task scheduling workflow with task level metrics and VM mapping.

The reason for selection of Crow Search Optimization algorithm is as follows;

- CSO's improved classification accuracy and higher dimensional compression capabilities demonstrate that it can handle feature selection problems with intense competition and adequate superiority (Rai and Kamble, 2023).
- Crow search algorithm helps to strike a balance between exploration and exploitation during the search phase, thereby enhancing the program's capacity to identify linked relationship (Zhang et al., 2023).
- The current approaches frequently do not consistently yield the best results on a variety of multi-class classification challenges. To identify the most important characteristics that help with precise task classification, the crow search technique is employed (Jayalatchumy et al., 2024).

As per the flow of the proposed model, it can be observed that database of tasks is initially processed by a set of SLA operations. These operations assist in resolving inter-task dependencies, which allows the scheduling layers to effectively map different tasks to suitable VMs. The following SLA-based method is utilized to complete this task, which assists in rearranging task order as per their make-span and deadline requirements,

All input tasks are properly organized as per their arrival timestamps, and client IDs are stored to verify SLA. A set of constants,

T_{SLA} = Task Timestamp SLA.

U_{SLA} = User Timestamp SLA are defined to enforce SLAs.

To perform enforcing operations, the following 2 conditions are checked, and tasks are rearranged as per these conditions.

Condition 1. Calculate differential timestamp between two tasks via Equation (1),

$$I_{SLA} = t(i + 1) - t(i) \quad (1)$$

This equation calculates the time difference between the arrival of two consecutive tasks where,

$t(i)$: Timestamp when the i^{th} task arrives.

$t(i + 1)$: Timestamp when the next task arrives.

This differential timestamp helps in understanding the interval between task arrivals, which is crucial for enforcing SLA related to task scheduling.

Condition 2. If $I_{SLA} > T_{SLA}$, then the task is rescheduled at the end of current queue sets.

Once all tasks are rescheduled, then a User-level SLA is estimated via Equation (2),

$$IU_{SLA} = t(i + 1)|_U - t(i)|_U \quad (2)$$

This equation calculates the time difference between tasks submitted by the same user where,

$t(i)|_U$: Timestamp when the i^{th} task from user U arrives.

$t(i + 1)|_U$: Timestamp when the next task from the same user arrives.

If $IU_{SLA} > U_{SLA}$, then current task is shifted at the end of scheduling queue sets.

Due to these enforcements, the following advantages are observed,

- Underlying cloud deployment is not overloaded with tasks from same set of users.
- Dependency between tasks is maintained, and independent tasks are executed concurrently on different VMs.

The aggregated set of tasks are clustered via a series of Hierarchical Clustering, K-Means, and Fuzzy C-Means (FCM), which assists in identification of requirement-based task clusters. As Fuzzy C refers to clustering used to group data even in the presence of noise or outliers. It can also be extended to semi-supervised clustering, which uses information from both labeled and unlabeled data, and multi view clustering, which considers multiple data perspectives (Li and Wang, 2024). K-Means algorithm divides data points into K clusters based on how far apart they are from each other's centers. Determining the initial ideal centroids of clusters is one of the primary roles of the K-Means algorithm. They group the task into different clusters to interpret various task computational requirements (Dubey and Rajavat, 2023). Hierarchical clustering helps to find similar kind of tasks into a single group. The end points are the set of clusters (Zhou et al., 2023). Clustering is applied so that all the incoming tasks are divided into various clusters and then scheduling is applied.

To perform these clustering operations, a task-level metric is calculated via Equation (3),

$$TLM = \frac{MS}{Max(MS)} + \frac{DL}{Max(DL)} \quad (3)$$

where, MS & DL represents make-span and deadline of individual tasks. Using this metric, all tasks are clustered into 3 groups, each of which contains tasks that have low-computational requirements, medium-computational requirements, and high-computational requirements. Through the use of a CSO procedure, these clustered tasks are mapped with relevant VMs. Three times, once for each cluster, this procedure is iterated to help with effective task-to-VM mapping depending on task needs and VM computing capabilities. A VM-level capacity $C(VM)$ metric is assessed prior to iterating over the tasks using Equation (4), which combines the VM's bandwidth (BW), processing capacity in Millions of Instructions Per Second (MIPS), and RAM availability in real-time.

$$C(VM) = \sum_{i=1}^{N_{PE}} \frac{BW_i}{Max(BW)} + \frac{MIPS}{Max(MIPS)} + \frac{RAM}{Max(RAM)} \quad (4)$$

where, N_{PE} = Represents number of processing elements (or cores) that are available with individual VMs. Once these values $C(VM)$ & TLM are estimated then the CSO Model is iterated for each cluster via the following process,

- To start the optimization process, a set of iterative parameters are initialized as follows,
- Count of the number of crows (N_c).
- Internal iterations for which each set of crows will be reconfigured (N_i).
- Crow-level learning rate (L_r).
- Maximum Number of VMs that will be used during these optimizations $C(VM)$.
- A set of N_c crows are initially setup via the following operations,

VMs are stochastically mapped with tasks, and their fitness levels are evaluated via Equation (5),

$$fS_i = \frac{\sum S_{task}}{\sum C(VM)} + \frac{\sum BW_{task}}{\sum B} + \frac{\sum RAM_{task}}{\sum R} + \frac{\sum DL_{task}}{\sum N_{PE} * \frac{R}{B}} \quad (5)$$

fs_i = fitness function.

$\sum S_{task}$ = total number of internal tasks (or subtasks).

$\sum BW_{task}$ = total bandwidth required by all the tasks.

$\sum RAM_{task}$ = total RAM required by all the tasks.

$\sum DL_{task}$ = combined deadlines of all the tasks.

$\sum B$ = the total bandwidth available across all VMs.

$\sum R$ = total RAM available across all VMs.

$\sum N_{PE}$ = total number of processing elements (or cores) available across all VMs.

R/B = the ratio of available RAM to bandwidth, indicating the overall capacity and balance of the VMs.

The most feasible configuration is generated via Equation (6),

$$fs_{out} = \text{Min}(\cup fs_i) \quad (6)$$

fs_{out} = represents the result of the optimization process which denotes the minimum fitness score among all the fitness scores calculated for different configurations.

$\cup fs_i$ = value from set of fitness score.

Due to selection of minimum f_s , the model can map high computational-requirement tasks to high-capacity VMs, thereby improving computational efficiency of the scheduling process.

After repeating this procedure for every crow, Equation (7) is used to estimate a crow-level fitness threshold.

$$f_{th} = \sum_{i=1}^{N_h} fs_i * \frac{L_r}{N_h} \quad (7)$$

f_{th} = Crow-level fitness threshold.

N_h = The number of configurations (or hypotheses) considered by the algorithm.

L_r = Crow-level learning rate (consider).

Check each of the Crow-configurations, and update them as per the following process, stochastically identify 2 Crow-configurations with,

$$fs > f_{th} \quad (8)$$

Crow with lower fitness is used to tune the configuration of crow with higher fitness via the following process,

From the higher fitness crow, select $L_r * N_{VM}$ stochastic VMs and replace these VMs with the VMs of Crow-configuration that is having lower fitness as per Equation (9),

$$S(New)|_{L_r * N_{VM}} = S(fs(Low))|_{L_r * N_{VM}} \quad (9)$$

where, $S(New)$ = The new VM configuration & $S(fs(Low))$ = The VM configuration of crow with lower fitness levels.

Repeat the same operations for N_i iterations, and for individual cluster sets in order to efficiently map SLA-aware tasks with capacity-aware VMs.

Once all iterations are completed for the 3 clusters, then VM-to-task mapping process is converged, and its parametric evaluation is done for different number of tasks. The detailed explanation of above concept is given in the form of algorithm as below:

Algorithm 1: Pseudo Code for the proposed model.

Input: Available VM configurations, Task Characteristics.

Output: Mapping of VMs to Tasks.

1. // **Phase 1: SLA-based Dependency Resolution**
 2. **for** $i = 1$ to (total no. of tasks - 1) **do**
 3. Compute $I_{SLA} = t(i + 1) - t(i)$
 4. **if** $I_{SLA} > T_{SLA}$ **then**
 5. Move task $t(i)$ to end of queue
 6. **end if**
 7. Compute $I_{USLA} = t(i + 1)|_U - t(i)|_U$
 8. **if** $I_{USLA} > U_{SLA}$ **then**
 9. Move next task to end of queue
 10. **end if**
 11. Shift tasks if SLA not satisfied; else proceed to next set
 12. **end for**
 13. // **Phase 2: Task-Level Clustering**
 14. Compute $TLM = \frac{MS}{Max(MS)} + \frac{DL}{Max(DL)}$
 15. Apply K-Means / Fuzzy C-Means / Hierarchical Clustering on TLM $\rightarrow$ 3 clusters
 16. (low, medium, high computational requirement)
 17. // **Phase 3: VM Capacity Estimation**
 18. Compute $C(VM) = \sum_{i=1}^{N_{PE}} \frac{BW_i}{Max(BW)} + \frac{MIPS}{Max(MIPS)} + \frac{RAM}{Max(RAM)}$
 19. // **Phase 4: CSO based VM-Task Mapping (for each cluster)**
 20. **for** each cluster (3 iterations) **do**
 21. **for** $i = 1$ to no. of crows **do**
 22. Compute fitness: $fs_i = \frac{\sum S_{task}}{\sum C(VM)} + \frac{\sum BW_{task}}{\sum B} + \frac{\sum RAM_{task}}{\sum R} + \frac{\sum DL_{task}}{\sum N_{PE} * \frac{R}{B}}$
 23. Select feasible configuration: $fs_{out} = Min(\cup fs_i)$
 24. **end for**
 25. Compute fitness threshold: $f_{th} = \sum_{i=1}^{N_h} fs_i * \frac{L_r}{N_h}$
 26. Identify crow pairs with $fs > f_{th}$
 27. Update configuration: $S(New)|_{L_r * N_{VM}} = S(fs(Low))|_{L_r * N_{VM}}$
 28. Repeat for N_i iterations
 29. **end for**
 30. // **Phase 5: Output**
 31. Select crow with optimal fitness $\rightarrow$ final VM-Task mapping
-

The proposed dependency aware clustering model are different from existing methods because of following features,

- **Focus on Scalability:** Existing methods often face scalability limitations, particularly when dealing with large-scale data centers DC comprising thousands of servers. The proposed approach acknowledges this

challenge and aims to address it by introducing a dependency-aware clustering methodology. This methodology is designed to enhance the efficiency of task scheduling, specifically targeting scalability concerns prevalent in cloud environments.

- **Integration of Self-Learning and Optimization:** While self-learning methods are acknowledged as crucial for dealing with the uncertainty and flux of user expectations, existing methods may not fully leverage these techniques. The proposed approach incorporates bioinspired optimizations, such as the CSO model, to enhance the scalability and adaptability of the scheduling process (Kumar et al., 2021). By integrating self-learning mechanisms with optimization algorithms, the proposed method can dynamically adjust to changing resource demands and user expectations, thereby improving overall resource efficiency and meeting performance goals more effectively.
- **Dependency-Aware Clustering:** Traditional cloud resource management approaches may overlook inter-task dependencies, leading to suboptimal scheduling decisions. In contrast, the proposed methodology emphasizes dependency-aware clustering, which considers the relationships and dependencies between tasks when organizing them into clusters. This approach ensures that tasks with dependencies are grouped together appropriately, allowing for more efficient resource allocation and scheduling.
- **SLA Proactive Renegotiation:** The proposed method places a significant emphasis on SLA management, particularly from the perspective of the CSP. It introduces proactive and continuous renegotiation techniques to maintain SLA compliance and enhance customer satisfaction. By continuously assessing service performance and anticipating potential SLA breaches, the proposed approach aims to mitigate risks and improve overall service quality.

With a focus on scalability, self-learning, dependency awareness, SLA management, and real-time usability, the suggested problem and solution provide an all-encompassing and creative way to tackle the difficulties of cloud resource management and task scheduling. These aspects distinguish the proposed approach from existing methods and highlight its potential contributions to advancing cloud computing technologies. The overall assessment is explained in the next section of paper, where execution delay, deadline hit ratio, scheduling efficiency, and energy consumed during scheduling are calculated and compared with existing methods for different task types.

4. Results

The proposed model initially scans input tasks and resolves dependencies using a SLA. Based on a score metric that combines make-span and deadline requirements, these dependency-resolved tasks are grouped. Combining K-Means, Fuzzy C-Means, and Hierarchical Clustering techniques makes task clustering easier for similar kind of task that includes low, medium and high computational requirements tasks. A Crow Search Optimization Model analyses the clusters and VM configurations, helping to map the VMs to the appropriate tasks. A fitness function that is deadline- and task-capacity-aware supports the mapping process and helps to improve scheduling performance for a variety of task types.

4.1 Dataset

This dataset is made up of a variety of logs from several sources, including the IBM SP2 logs from the Cornell Theory Center (CTC), the T3D logs from the Lawrence Livermore National Lab (LLNL), the logs from the Laboratory of Corpuscular Physics, and the Intel Net-batch (A, B, C, and D) logs. Together, these logs created 500k task records, which were then scheduled using the CSO-based scheduling method among 100 distinct virtual machines.

4.2 Simulation Results

Efficiency of this model is estimated in terms of task execution delay (or computational effort) (D) is measured in milliseconds (ms), Scheduling Efficiency (SE) is measured in percentage (%), Deadline Hit Ratio (DHR) is measured in percentage (%), and energy required (E) is measured in millijoules (mJ) for various VM & task configurations. Performance of this model is compared with Multi-Objectives Symbiotic Organism Search (MOSOS) (Belgacem et al., 2022), Reinforcement Learning (RL) (Chen et al., 2022a), and Multi Agent Optimization (MAO) (Gao et al., 2021a), where performance of these models was referred from their individual texts, and approximated on custom task-sets.

Efficiency of this scheduling process was initially estimated in terms of scheduling delay, which is calculated via Equation (10),

$$D = \frac{1}{NET} \sum_{i=1}^{NET} EC_i * (t_{complete_i} - t_{start_i}) \quad (10)$$

where, t_{start} & $t_{complete}$ are the timestamps at which each of the Number of Evaluation Tasks (NETs) have arrived, and task has completed execution on the given set of VMs, while EC represents total number of execution cycles for individual tasks. The evaluation of Delay (D) can be observed from **Table 2** as follows.

Table 2. Scheduling delay for different tasks.

Number of tasks	D(ms) MOSOS	D(ms) RL	D(ms) MAO	D(ms) DCMTSB (proposed)
5k	0.36	0.43	0.49	0.18
10k	0.42	0.51	0.58	0.21
15k	0.49	0.62	0.7	0.25
20k	0.57	0.76	0.85	0.3
25k	0.68	0.94	1.03	0.36
30k	0.82	1.15	1.25	0.43

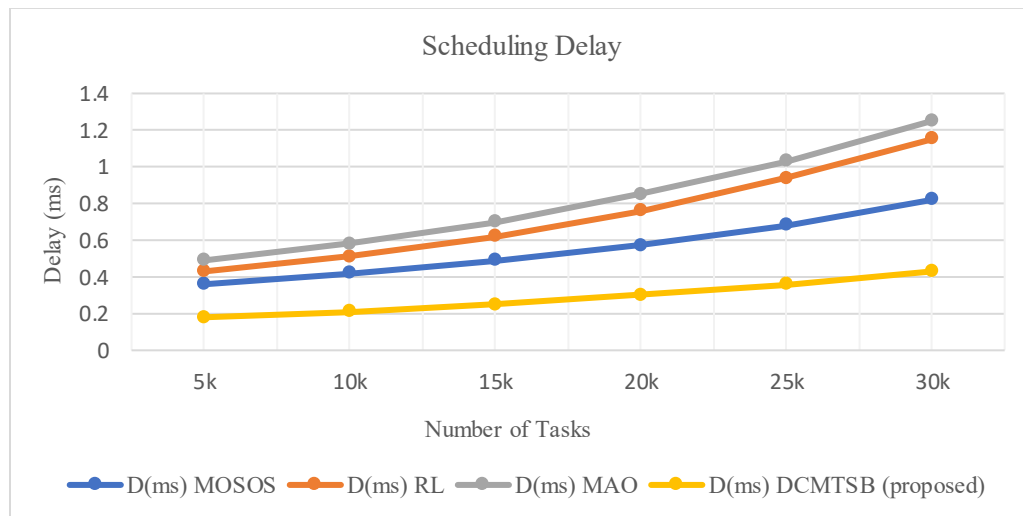


Figure 2. Scheduling delay for different tasks.

Based on this evaluation and **Figure 2**, it can be observed that the proposed CSO-based scheduling model is able to improve the scheduling speed by 10.5% when compared with MOSOS (Belgacem et al., 2022),

18.5% when compared with RL (Chen et al., 2022a) and 15.4% when compared with MAO (Gao et al., 2021a), which makes it highly useful for a wide variety of scheduling use cases. The reason for this reduction in scheduling delay is use of SLA-aware operations along with clustering processes, which assist in improving task-to-VM Mapping efficiency under real-time scenarios. Similar to this, the Deadline Hit Ratio (DHR) is evaluated via Equation (11),

$$DHR = \sum_{i=1}^{NET} \frac{N_{t_d}}{NET * T_t} \quad (11)$$

where, N_{t_d} represents count of tasks that are executed within the given deadline, while T_t represents total count of tasks that are executed by the VMs. The evaluation of DHR can be observed from **Table 3** as follows:

Table 3. Scheduling DHR for different tasks.

Number of tasks	DHR (%) MOSOS	DHR (%) RL	DHR (%) MAO	DHR (%) DCMTSB (proposed)
5k	90.5	94.2	90.4	98.5
10k	92.3	94.5	90.5	98.6
15k	93.5	94.9	91.3	99.5
20k	93.7	96.5	92.5	99.8
25k	93.8	96.6	94.2	99.9
30k	93.9	97.2	94.5	99.95

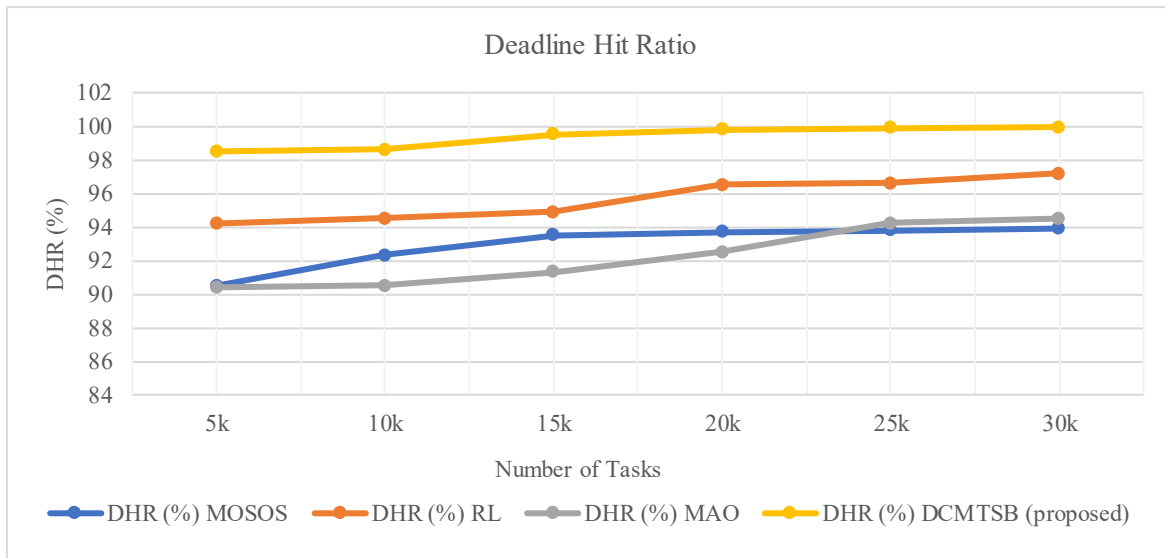


Figure 3. Scheduling DHR for different tasks.

Based on this evaluation and **Figure 3**, it can be observed that the proposed clustering-based SLA-aware model is able to improve the scheduling DHR by 4.9% when compared with MOSOS (Belgacem et al., 2022), 3.5% when compared with RL (Chen et al., 2022a) and 5.4% when compared with MAO (Gao et al., 2021a), which makes it highly useful for deadline-aware scenarios. The reason for this improvement in DHR is use of CSO's fitness function that maximizes probability of meeting deadlines while maintaining higher scheduling performance, which assist in improving task-to-VM Mapping efficiency under real-time scenarios. Similarly, the efficiency of scheduling is estimated via Equation (12),

$$SE = \sum_{i=1}^{NET} \frac{NCC_{opt}}{NET * NCC} \quad (12)$$

where, NCC_{opt} represents optimal number of scheduling cycles for the task, and NCC represents actual number of cycles needed to schedule individual tasks. This scheduling efficiency is tabulated in the following **Table 4**.

Table 4. Scheduling efficiency for different tasks.

Number of tasks	SE (%) MOSOS	SE (%) RL	SE (%) MAO	SE (%) DCMTSB (proposed)
5k	70.5	75.5	70.9	85.5
10k	70.9	75.9	72.3	86.9
15k	75.4	76.4	73.5	89.5
20k	76.5	77.9	74.6	90.4
25k	77.95	78.5	75.9	91.74
30k	78.53	80.05	77.61	92.25

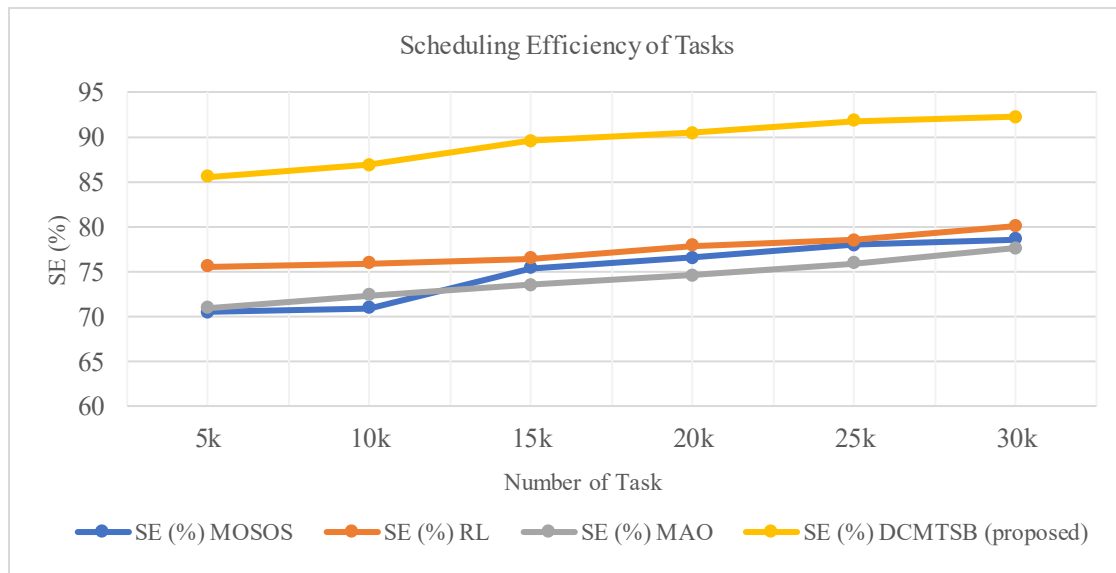


Figure 4. Scheduling efficiency for different tasks.

The suggested SLA-aware CSO-based model can enhance scheduling efficiency by 12.5% when compared to MOSOS (Belgacem et al., 2022), 15.4% when compared to RL (Chen et al., 2022a) and 14.9% when compared to MAO (Gao et al., 2021a), according to this evaluation and **Figure 4**. This makes it beneficial for high-performance scheduling scenarios. The usage of makespan and deadline when mapping virtual machines with tasks is the cause of this increase in scheduling efficiency. This helps to improve resource utilization in real-time applications. Similarly, Equation (13) is used to estimate the energy needed for scheduling.

$$E = \frac{1}{NET} \sum_{i=1}^{NET} E_{start_i} - E_{end_i} \quad (13)$$

where, E_{start} & E_{end} are VM energy levels during the start & completion of tasks. These energy levels can be observed from **Table 5** as follows,

Table 5. Energy required for scheduling different tasks.

Number of tasks	E (mJ) MOSOS	E (mJ) RL	E (mJ) MAO	E (mJ) DCMTSB (proposed)
5k	150.5	125.5	90.5	80.5
10k	155.4	128.9	93.5	83.5
15k	158.9	135.4	94.9	87.9
20k	160.4	140.9	96.8	88.4
25k	162.3	143.8	99.5	95.98
30k	165.5	149.5	105.4	96.51

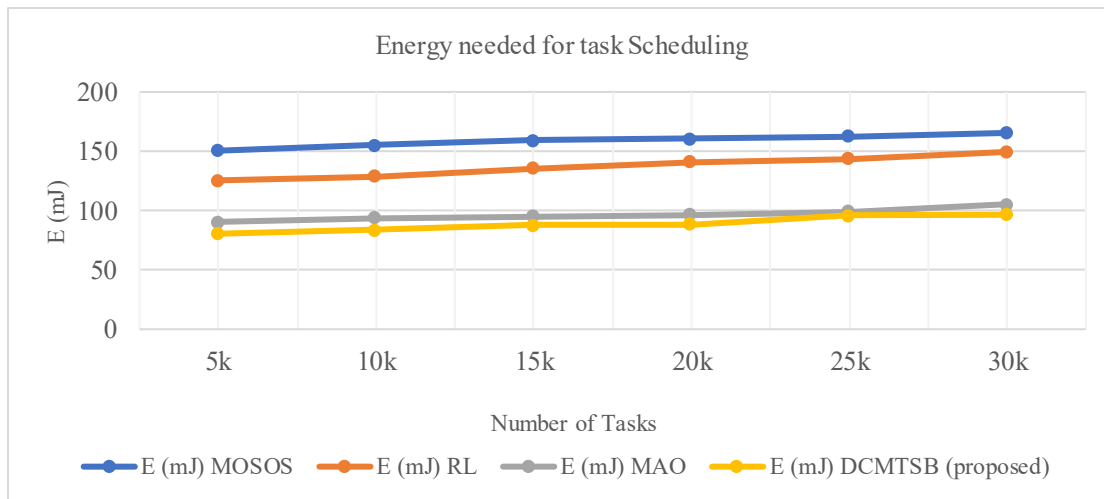
**Figure 5.** Energy required to schedule different tasks.

Figure 5 and this evaluation show that the suggested fused-clustering based model can reduce the energy required for scheduling by 18.3% compared to MOSOS (Belgacem et al., 2022), 12.5% compared to RL (Chen et al., 2022) and 3.9% compared to MAO (Gao et al., 2021). This makes it useful for high-lifetime scheduling scenarios. The application of various clustering models in conjunction with SLA-aware strategies, which aid in enhancing resource utilization for heterogeneous scenarios, is the cause of this increase in energy efficiency. The suggested model can enhance scheduling performance for many job types in real-time environments as a result of these enhancements.

5. Conclusion and Future Scope

In this research, we developed a dependency aware clustering architecture that uses bioinspired optimization strategies to improve task scheduling performance using the Crow Search Algorithm. A Crow Search Optimization Model analyses clusters and VM configurations to assist in assigning VMs to the correct tasks. A deadline and task-aware fitness function supports the mapping process and improves scheduling performance for a variety of task types. Experimental outcomes show that efficient task-scheduling models seek to maximize cloud use while retaining a greater deadline hit ratio, reduced cloud effort, and higher levels of job diversity. Because of these improvements, the proposed model can increase task scheduling performance for a variety of job categories. The performance of the proposed model should be assessed for more tasks in the future, and it could be enhanced by using Convolutional Neural Networks based on deep learning, which aid in the proactive mapping of tasks to relevant virtual machines. Its performance can also be improved by combining hybrid bioinspired learning with incremental learning procedures, allowing the model to adapt the capacity of VMs for high-efficiency real-time scheduling tasks.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the support and resources provided by their institution in completing this research.

AI Disclosure

The author(s) declare that no assistance is taken from generative AI to write this article.

References

- Alam, A.B., Fadlullah, Z.M., & Choudhury, S. (2021). A resource allocation model based on trust evaluation in multi-cloud environments. *IEEE Access*, 9, 105577-105587. <https://doi.org/10.1109/access.2021.3100316>
- Belgacem, A., Beghdad-Bey, K., & Nacer, H. (2022). Dynamic resource allocation method based on symbiotic organism search algorithm in cloud computing. *IEEE Transactions on Cloud Computing*, 10(3), 1714-1725. <https://doi.org/10.1109/tcc.2020.3002205>
- Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X., & Rong, C. (2022a). Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Transactions on Cloud Computing*, 10(2), 1117-1129. <https://doi.org/10.1109/tcc.2020.2992537>
- Chen, Z., Hu, J., Min, G., Luo, C., & El-Ghazawi, T. (2022b). Adaptive and efficient resource allocation in cloud datacenters using actor-critic deep reinforcement learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(8), 1911-1923. <https://doi.org/10.1109/tpds.2021.3132422>
- Chhetri, M.B., Forkan, A.R.M., Vo, Q.B., Nepal, S., & Kowalczyk, R. (2021). Exploiting heterogeneity for opportunistic resource scaling in cloud-hosted applications. *IEEE Transactions on Services Computing*, 14(6), 1739-1750. <https://doi.org/10.1109/tsc.2019.2908647>
- Du, J., Jiang, C., Benslimane, A., Guo, S., & Ren, Y. (2022). SDN-based resource allocation in edge and cloud computing systems: an evolutionary Stackelberg differential game approach. *IEEE/ACM Transactions on Networking*, 30(4), 1613-1628. <https://doi.org/10.1109/tnet.2022.3152150>
- Dubey, P., & Rajavat, A. (2023). Effective K-Means clustering algorithm for efficient data mining. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies* (pp. 1-6). IEEE, Vellore, India. <https://doi.org/10.1109/vitecon58111.2023.10157179>
- Fan, Y., Wang, L., Wu, W., & Du, D. (2021). Cloud/edge computing resource allocation and pricing for mobile blockchain: an iterative greedy and search approach. *IEEE Transactions on Computational Social Systems*, 8(2), 451-463. <https://doi.org/10.1109/tcss.2021.3049152>
- Fang, C., Xu, H., Yang, Y., Hu, Z., Tu, S., Ota, K., Yang, Z., Dong, M., Han, Z., Yu, F.R., & Liu, Y. (2022). Deep-reinforcement-learning-based resource allocation for content distribution in fog radio access networks. *IEEE Internet of Things Journal*, 9(18), 16874-16883. <https://doi.org/10.1109/jiot.2022.3146239>
- Ferdouse, L., Woungang, I., Anpalagan, A., & Erkucuk, S. (2021). Energy efficient downlink resource allocation in cellular IoT supported H-CRANs. *IEEE Transactions on Vehicular Technology*, 70(6), 5803-5816. <https://doi.org/10.1109/tvt.2021.3076825>
- Gamsiz, M., & Özer, A.H. (2021). An energy-aware combinatorial virtual machine allocation and placement model for green cloud computing. *IEEE Access*, 9, 18625-18648. <https://doi.org/10.1109/access.2021.3054559>
- Gao, G., Xiao, M., Wu, J., Huang, H., Wang, S., & Chen, G. (2021a). Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud. *IEEE Transactions on Services Computing*, 14(6), 1702-1716. <https://doi.org/10.1109/tsc.2019.2902549>

- Gao, X., Liu, R., & Kaushik, A. (2021b). Hierarchical multi-agent optimization for resource allocation in cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 32(3), 692-707. <https://doi.org/10.1109/tpds.2020.3030920>
- Gao, X., Liu, R., Kaushik, A., & Zhang, H. (2022). Dynamic resource allocation for virtual network function placement in satellite edge clouds. *IEEE Transactions on Network Science and Engineering*, 9(4), 2252-2265. <https://doi.org/10.1109/tnse.2022.3159796>
- He, F., Sato, T., Chatterjee, B.C., Kurimoto, T., Urushidani, S., & Oki, E. (2022). Robust optimization model for primary and backup resource allocation in cloud providers. *IEEE Transactions on Cloud Computing*, 10(4), 2920-2935. <https://doi.org/10.1109/tcc.2021.3051018>
- Hosseinalipour, S., & Dai, H. (2021). A two-stage auction mechanism for cloud resource allocation. *IEEE Transactions on Cloud Computing*, 9(3), 881-895. <https://doi.org/10.1109/tcc.2019.2901785>
- Jawade, P., Borkar, G.M., & Ramachandram, S. (2022). Confinement forest-based enhanced min-min and max-min technique for secure multicloud task scheduling. *Transactions on Emerging Telecommunications Technologies*, 33(9), e4515. <https://doi.org/10.1002/ett.4515>
- Jayalatchumy, D., Ramalingam, R., Balakrishnan, A., Safran, M., & Alfarhood, S. (2024). Improved crow search-based feature selection and ensemble learning for IoT intrusion detection. *IEEE Access*, 12, 33218-33235. <https://doi.org/10.1109/access.2024.3372859>
- Kim, I.K., Hwang, J., Wang, W., & Humphrey, M. (2022a). Guaranteeing performance SLAs of cloud applications under resource storms. *IEEE Transactions on Cloud Computing*, 10(2), 1329-1343. <http://doi.org/10.1109/TCC.2020.2985372>
- Kim, I.K., Wang, W., Qi, Y., & Humphrey, M. (2022b). Forecasting cloud application workloads with cloudinsight for predictive resource management. *IEEE Transactions on Cloud Computing*, 10(3), 1848-1863. <https://doi.org/10.1109/tcc.2020.2998017>
- Kumar, K.R.P., Kousalya, K., Vishnupriya, S., Ponni, S., & Logeswaran, K. (2021). Enhanced crow search algorithm for task scheduling in cloud computing. In *2021 IOP Conference Series: Materials Science and Engineering* (Vol. 1055, pp. 012102). IOP Publishing. Erode, India. <https://doi.org/10.1088/1757-899x/1055/1/012102>
- Kumar, N., & Kumar, S. (2021). Conceptual service level agreement mechanism to minimize the SLA violation with SLA negotiation process in cloud computing environment. *Baghdad Science Journal*, 18(2), 1020-1029. [https://doi.org/10.21123/bsj.2021.18.2\(suppl.\).1020](https://doi.org/10.21123/bsj.2021.18.2(suppl.).1020)
- Lattuada, M., Barbierato, E., Gianniti, E., & Ardagna, D. (2022). Optimal resource allocation of cloud-based spark applications. *IEEE Transactions on Cloud Computing*, 10(2), 1301-1316. <https://doi.org/10.1109/tcc.2020.2985682>
- Li, H., & Wang, J. (2024). From soft clustering to hard clustering: a collaborative annealing fuzzy c-means algorithm. *IEEE Transactions on Fuzzy Systems*, 32(3), 1181-1194. <https://doi.org/10.1109/tfuzz.2023.3319663>
- Li, S., Huang, J., & Cheng, B. (2021). Resource pricing and demand allocation for revenue maximization in IaaS clouds: a market-oriented approach. *IEEE Transactions on Network and Service Management*, 18(3), 3460-3475. <https://doi.org/10.1109/tnsm.2021.3085519>
- Liu, J., Guo, S., Liu, K., & Feng, L. (2022a). Resource provision and allocation based on microeconomic theory in mobile edge computing. *IEEE Transactions on Services Computing*, 15(3), 1512-1525. <https://doi.org/10.1109/tsc.2020.3000050>
- Liu, T., Fang, L., Zhu, Y., Tong, W., & Yang, Y. (2022b). A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing. *IEEE Transactions on Mobile Computing*, 21(8), 2687-2700. <https://doi.org/10.1109/tmc.2020.3045471>
- Mayyahi, M.A.A., & Seno, S.A.H. (2022). A security and privacy aware computing approach on data sharing in cloud environment. *Baghdad Science Journal*, 19(6), 1572-1580. <https://doi.org/10.21123/bsj.2022.7077>

- Middya, A.I., Ray, B.K., & Roy, S. (2022). Auction-based resource allocation mechanism in federated cloud environment: TARA. *IEEE Transactions on Services Computing*, 15(1), 470-483. <https://doi.org/10.1109/tsc.2019.2952772>
- Mireslami, S., Rakai, L., Wang, M., & Far, B.H. (2021). Dynamic cloud resource allocation considering demand uncertainty. *IEEE Transactions on Cloud Computing*, 9(3), 981-994. <https://doi.org/10.1109/tcc.2019.2897304>
- Misra, S., Schober, R., & Chakraborty, A. (2022). RACE: QoI-aware strategic resource allocation for provisioning Se-aaS. *IEEE Transactions on Services Computing*, 15(3), 1540-1550. <https://doi.org/10.1109/tsc.2020.3001078>
- Najm, M., Patra, M., & Tamarapalli, V. (2021). Cost-and-delay aware dynamic resource allocation in federated vehicular clouds. *IEEE Transactions on Vehicular Technology*, 70(6), 6159-6171. <https://doi.org/10.1109/tvt.2021.3079912>
- Rai, K., & Kamble, M. (2023). Incubation of a metaheuristic searching approach for intrusion detection into a new constrained crow search algorithm. In *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science* (pp. 1-9). IEEE. Bhopal, India. <https://doi.org/10.1109/sceecs57921.2023.10063120>
- Salman, S.A., Dheyab, S.A., Salih, Q.M., & Hammood, W.A. (2023). Parallel machine learning algorithms. *Mesopotamian Journal of Big Data*, 3(1), 16-28. <https://doi.org/10.58496/mjbd/2023/002>
- Saxena, D., Singh, A.K., & Buyya, R. (2022). OP-MLB: an online VM prediction-based multi-objective load balancing framework for resource management at cloud data center. *IEEE Transactions on Cloud Computing*, 10(4), 2804-2816. <https://doi.org/10.1109/tcc.2021.3059096>
- Tan, B., Ma, H., Mei, Y., & Zhang, M. (2022). A cooperative coevolution genetic programming hyper-heuristics approach for on-line resource allocation in container-based clouds. *IEEE Transactions on Cloud Computing*, 10(3), 1500-1514. <https://doi.org/10.1109/tcc.2020.3026338>
- Tao, Y., Qiu, J., Lai, S., Sun, X., & Zhao, J. (2022). Market-based resource allocation of distributed cloud computing services: virtual energy storage systems. *IEEE Internet of Things Journal*, 9(22), 22811-22821. <https://doi.org/10.1109/jiot.2022.3184750>
- Xu, J., & Palanisamy, B. (2021). Optimized contract-based model for resource allocation in federated geo-distributed clouds. *IEEE Transactions on Services Computing*, 14(2), 530-543. <https://doi.org/10.1109/tsc.2018.2797910>
- Zafari, F., Leung, K.K., Towsley, D., Basu, P., Swami, A., & Li, J. (2021). Let's share: a game-theoretic framework for resource sharing in mobile edge clouds. *IEEE Transactions on Network and Service Management*, 18(2), 2107-2122. <https://doi.org/10.1109/tnsm.2020.3044870>
- Zhang, X., Peng, M., Yan, S., & Sun, Y. (2022). Joint communication and computation resource allocation in fog-based vehicular networks. *IEEE Internet of Things Journal*, 9(15), 13195-13208. <https://doi.org/10.1109/jiot.2022.3140811>
- Zhang, Y., Shang, J., Gu, Y., Ren, Q., Ge, D., & Sun, Y. (2023). Crow search algorithm based on information interaction for epistasis detection. In *2023 IEEE International Conference on Bioinformatics and Biomedicine* (pp. 3249-3256). IEEE. Istanbul, Turkiye. <https://doi.org/10.1109/bibm58861.2023.10385883>
- Zhou, S., Chen, Z., Li, Q., Gu, M., Bao, Z., He, W., & Sheng, W. (2023). A multi-objective evolutionary algorithm with hierarchical clustering-based selection. *IEEE Access*, 11, 2557-2569. <https://doi.org/10.1109/access.2023.3234226>