

SSKHOA: Hybrid Metaheuristic Algorithm for Resource Aware Task Scheduling in Cloud-fog Computing

M. Santhosh Kumar*

SCOPE, VIT-AP University, Amaravathi, A.P., India

E-mail: santhosh.21phd7113@vitap.ac.in

ORCID iD: <https://orcid.org/0000-0001-5936-6582>

*Corresponding author

K. Ganesh Reddy

SCOPE, VIT-AP University, Amaravathi, A.P., India

E-mail: guncity11@gmail.com

Rakesh Kumar Donthi

University College of Dublin, Dublin city, Ireland

E-mail: drakesh2175@gmail.com

Received: 11 August 2023; Revised: 05 October 2023; Accepted: 05 November 2023; Published: 08 February 2024

Abstract: Cloud fog computing is a new paradigm that combines cloud computing and fog computing to boost resource efficiency and distributed system performance. Task scheduling is crucial in cloud fog computing because it decides the way computer resources are divided up across tasks. Our study suggests that the Shark Search Krill Herd Optimization (SSKHOA) method be incorporated into cloud fog computing's task scheduling. To enhance both the global and local search capabilities of the optimization process, the SSKHOA algorithm combines the shark search algorithm and the krill herd algorithm. It quickly explores the solution space and finds near-optimal work schedules by modelling the swarm intelligence of krill herds and the predator-prey behavior of sharks. In order to test the efficacy of the SSKHOA algorithm, we created a synthetic cloud fog environment and performed some tests. Traditional task scheduling techniques like LTRA, DRL, and DAPSO were used to evaluate the findings. The experimental results demonstrate that the SSKHOA outperformed the baseline algorithms in terms of task success rate increased 34%, reduced the execution time by 36%, and reduced makespan time by 54% respectively.

Index Terms: Cloud Computing, Fog Computing, Task Scheduling, Shark Search Algorithm, Krill Herd Algorithm, Nature-inspired.

1. Introduction

The combination of cloud and fog computing has emerged as an effective paradigm for solving the problems that plague distributed computing systems. By combining the power of the cloud with that of dispersed fog devices, we can better utilize our resources and boost our performance [1,2]. In cloud fog computing, task scheduling is crucial because it affects how computing resources are distributed across workloads.

Traditional task scheduling algorithms, such as Round Robin and First Come First Serve, are not well-suited for the dynamic and heterogeneous nature of cloud fog environments [3,4]. These environments consist of a diverse set of resources, including cloud data centres, edge devices, and IoT devices, with varying computing capacities, network connectivity, and energy constraints [5]. Additionally, the workload in cloud fog systems can be highly dynamic, with varying task characteristics and resource availability.

To address these challenges, researchers have turned to nature-inspired optimization algorithms, which mimic the collective behavior of natural systems to solve complex optimization problems. One such algorithm is the Shark Search Krill Herd Optimization (SSKHOA) algorithm, which combines the characteristics of two nature-inspired algorithms: shark search algorithm and krill herd algorithm.

The shark search algorithm is inspired by the behavior of sharks in hunting their prey. It utilizes the global search ability of sharks to explore the solution space efficiently and locate potential solutions. On the other hand, the krill herd algorithm imitates the collective behavior of a group of krill, where each krill adjusts its position based on its own experience and the experiences of neighboring krill. This local search capability allows the algorithm to converge towards optimal solutions.

To improve the distribution of tasks to resources in an evolving and diverse environment, we propose incorporating the SSKHOA algorithm into cloud fog computing's task scheduling in an effort to exceed the limits of conventional scheduling methods. To produce near-optimal task schedules that reduce task completion time, maximize resource consumption, and boost overall system efficiency, the SSKHOA algorithm takes into account a number of criteria such as task characteristics, resource availability, and network circumstances.

The novelty of task scheduling in cloud fog computing using the Shark Search Krill Herd Optimization Algorithm (SSKHOA) resides in its innovative blend of nature-inspired optimization techniques, making it a powerful tool in optimizing resource allocation in cloud and fog environments. By combining the strengths of the Shark Search Optimization and Krill Herd Optimization algorithms, SSKHOA introduces a novel approach to task scheduling. It harnesses the collective intelligence of krill herds and replicates the predator-prey dynamics of sharks to swiftly explore solution spaces and locate nearly optimal work schedules. This biologically-inspired amalgamation improves the efficiency of task scheduling in cloud fog computing by effectively balancing exploration and exploitation.

Furthermore, the adaptability and scalability of SSKHOA make it uniquely suited to address the complexities of dynamic and heterogeneous cloud fog environments. The algorithm's capability to consider various factors such as task characteristics, resource availability, network conditions, and energy consumption holistically sets it apart. The experimental validation of SSKHOA further underscores its novelty, as it has consistently demonstrated remarkable enhancements over traditional task scheduling techniques. This all-encompassing approach to task scheduling, enriched with biological inspiration and performance advancements, marks the novelty of SSKHOA in the context of cloud fog computing.

The objective of this research is to examine how well the SSKHOA algorithm performs in comparison to more conventional task scheduling algorithms while operating in a simulated cloud fog scenario. We will evaluate it based on how quickly tasks are completed, how efficiently resources are used, and how productive the system as a whole is. This study's results will help improve cloud fog computing work scheduling approaches and provide light on how optimization algorithms inspired by nature can boost the efficiency and scalability of distributed systems.

We propose the following improvements to the current state of the art in order to enhance IoT services in a cloud-fog computing environment:

- To provide efficient task scheduling in cloud-fog environments, we present a hybrid evolutionary method named SSKHOA. This algorithm combines the foundational SSO methodology with the adaptive KHO strategy to increase convergence speed and searching explorations.
- Makespan, task execution time and success rate that are three competing criteria that must be taken into account when designing an effective model for Internet of Things (IoT) tasks transmitted for processing in a cloud-fog framework.
- Comparing the makespan, task success rate and task execution time of the proposed scheduling strategy to those of other approaches applied to realistic workloads.

The present article is divided into the following sections: Section 2 reviews task scheduling in cloud and fog computing. The SSKHOA algorithm and its execution are covered in depth in Section 3. Section 4 provides a detailed description of the experiment, the findings, and any potential repercussions. We discussed drawing conclusions and expanding task possibilities in Section 5 of the article.

2. Literature Survey

Cloud fog computing depends significantly on task scheduling, which seeks to optimize the allocation of tasks to resources in an adaptive and heterogeneous environment. Traditional scheduling algorithms may not be suitable for cloud fog environments due to their limited adaptability to changing conditions. To overcome these challenges, researchers have explored nature-inspired optimization algorithms, for greater task scheduling in cloud/fog computing settings, use algorithms like the Shark Search Krill Herd Optimization (SSKHOA) method. An overview of the study on the SSKHOA algorithm's usage for task scheduling in cloud and fog computing is given in this review of the literature.

In their study, Lin, Xue, et al. [6] address the imperative issue of energy minimization in the mobile cloud computing environment through task scheduling incorporating dynamic voltage and frequency scaling (DVFS). Their research emphasizes the significance of energy-efficient practices within the dynamic realm of mobile cloud computing. The integration of DVFS into task scheduling is explored as a means to optimize energy consumption. This study represents an early recognition of the growing relevance of energy-efficient practices and the potential offered by DVFS for enhancing task scheduling in mobile cloud computing. It underscores the continuous exploration of innovative methodologies aimed at reducing energy consumption, improving resource allocation, and ensuring the sustainability of

mobile cloud services, reflecting the increasing importance of energy efficiency within the field of services computing.

In the work authored by Cheng, Feng, et al. [7], the focus lies on cost-aware job scheduling for cloud instances, employing deep reinforcement learning techniques. This research underscores the significance of cost efficiency within cloud computing environments. The integration of deep reinforcement learning is explored as a means to optimize job scheduling while considering cost implications. The study exemplifies the increasing importance of cost-aware practices and the potential offered by deep reinforcement learning in enhancing task scheduling in cloud instances. It signifies the continuous exploration of advanced methodologies that aim to minimize operational expenses, maximize resource utilization, and promote cost-effective cloud computing, highlighting the growing relevance of economic considerations in the realm of cluster computing.

Table 1. Analysis of various techniques and parameters in study of task scheduling

| Author & Year | Technique Used | Parameters Addressed | Limitations |
|--|----------------|---|--|
| Zuo, Liyun, et al. (2015) [4] | PBACO | Makespan, cost, deadline violation rate, resource utilization. | In this proposed technique will not give the accurate result for multiple parameters while in execution process |
| Lin, Bing, et al. (2016) [5] | MCPCPP | Execution cost | Author unable to address the execution time, data transfer time for a huge Data workflow in Multi-clouds. |
| Lin, Xue, et al. (2014) [6] | LTRA | Energy consumption, task reduction time | In this study author considered only limited number of parameters. |
| Cheng, Feng, et al. (2022) [7] | DRL | Response time, success rate, cost | Author not concentrated on real world problems. |
| Zhou, Zhou, et al. (2018) [8] | M-PSO | Monetary cost | Meta heuristic optimization methods for multi cloud environment is not considered |
| Jangu, Nupur, and Zahid Raza. (2022) [9] | IJFA | Make-span time, execution costs, resource utilization. | Author unable to address the Multi-objective model lowering the integrated fog-cloud architecture's response time and energy consumption |
| Singh, Gyan, and Amit K. Chaturvedi. (2023) [10] | HGA | Cost, execution time | This work huge data centres space is not considered |
| Zahra, Movahedi, et al. (2021) [11] | OppoCWOA | Energy consumption and QoS parameters | In this study future research on data privacy with the help of the blockchain based FogBus platform was not done. |
| Iftikhar, Sundas, et al. (2023) [12] | HunterPlus | Energy consumption, job completion rate | It simply records one snapshot of the system's performance at any given moment. |
| Yin, Zhenyu, et al. (2022) [13] | HMA | Task completion rate and power consumption. | unable to resolve the issue with intelligent manufacturing lines' task flow scheduling. |
| Pham, Xuan-Qui, et al. (2017) [14] | CMAS | Cost, makespan and schedule length | unable to implement the plan for deployment into real-world systems. We see CEP as a viable use for the IoT. |
| Mangalampalli, Sudheer, et al. (2022) [15] | PSCSO | Makespan, energy consumption | Author not considered the multi objective optimization. Proposed approach will not give the good amount of result. |
| Hosseinioun, Pejman, et al. (2020) [16] | DVFS | Energy consumption | Some crucial elements, such privacy metrics and trust, are not covered. |
| Liu, Lindong, et al. (2018) [17] | TSFC | Total execution time and average waiting time. | Other problems including match difficulties, string mapping, and oblivious RAM are not addressed by the TSFC technique. |
| Bakshi, Mohana, et al. (2023) [18] | CSO | Energy consumption and resource utilization | Author's inability to address the issue of pre-emptive job scheduling will be investigated in light of the need for job relocation. |
| Badri, Sahar, et al. (2023) [19] | CNN-MBO | Energy consumption, response time, execution time, task utilization | The proposed was not evaluated in real time scenario. |
| Ahmed, Omed Hassan, et al. (2021) [20] | DMFO-DE | Number of applied VMs, and energy consumption. | research on improving fog computing in multi-cloud systems is lacking. |
| Mangalampalli, Sudheer, et al. (2020) [21] | PSCSO | Makespan, Migration time and the Total Power cost | The real time simulation is missing in this study. |
| Sindhu, V., and M. Prakash. (2022) [22] | ECBTSA-IRA | Schedule length, cost, and energy | As more network characteristics were taken into account in this study, the author was unable to focus on heterogeneous IoT devices to meet their QoS requirements. |
| Kumar, M. Santhosh, et al. (2023) [23] | EEOA | Cost, makespan, energy consumption | The proposed technique is best for only in terms of energy consumption. Other metrics are inadequate to address. |
| Moharram, Mohammed Abdulmajeed, et al. (2022) [24] | KHA | Indian Pines scene, Salinas scene, Botswana scene, Pavia University scene | Author proposed technique is adequate to address the complex problems. |
| Kumar, M. Santhosh, and G R Kumar. (2024) [25] | EAEFA | Makespan time, response time, execution time and energy usage | Proposed technique QoS was not met the up to the mark of scheduling requirements. |

In their research, Zhou, Zhou, et al. [8] introduce a modified Particle Swarm Optimization (PSO) algorithm tailored for task scheduling optimization in cloud computing. This study underscores the critical role of efficient task

scheduling within the cloud computing domain. The utilization of a customized PSO algorithm is explored as a means to enhance the allocation of computational tasks. The research exemplifies the ongoing pursuit of innovative methodologies that aim to fine-tune cloud-based task scheduling, improve resource utilization, and address the evolving challenges in cloud computing. It signifies the adaptability and potential of PSO algorithms in advancing task scheduling optimization, contributing to the continuous progress in cloud computing and its application in various domains.

We categorize the many criteria utilized by prior investigations, such as makespan, energy consumption, and SLA-based trust factors, in Table 1 by the categories indicated above. To facilitate effective task scheduling in a cloud fog environment, we created the SSKHOA, which accounts for success rate, execution time, and makespan time using SSO and KHO approaches. The proposed method has significant advantages for both real-time and latency-sensitive applications, such as smart cities and automobile networks.

The literature review demonstrates the growing interest in job scheduling in cloud computing using the Shark Search Krill Herd Optimization algorithm. To enhance search performance, convergence speed, and solution quality, researchers have suggested a variety of improvements and hybridization techniques. By effectively allocating resources, reducing reaction times, and maximizing performance indicators, the evaluated experiments show that the SSKHOA algorithm may effectively manage the task scheduling difficulties in cloud computing. However, more investigation is required to examine various characteristics of cloud computing, such as scalability and adaptation to changing surroundings.

3. Research Methods

3.1. System Model

Distributing tasks to resources in a cloud/fog computing system is as termed task scheduling. As shown in fig. 1, a three-tiered architecture consisting of the cloud, fog, and edge layers is typically used to successfully address the difficulties associated with scheduling tasks. When applied to cloud fog computing, this architecture with the Shark Search Krill Herd Optimization (SSKHOA) algorithm provide an effective method for scheduling tasks.

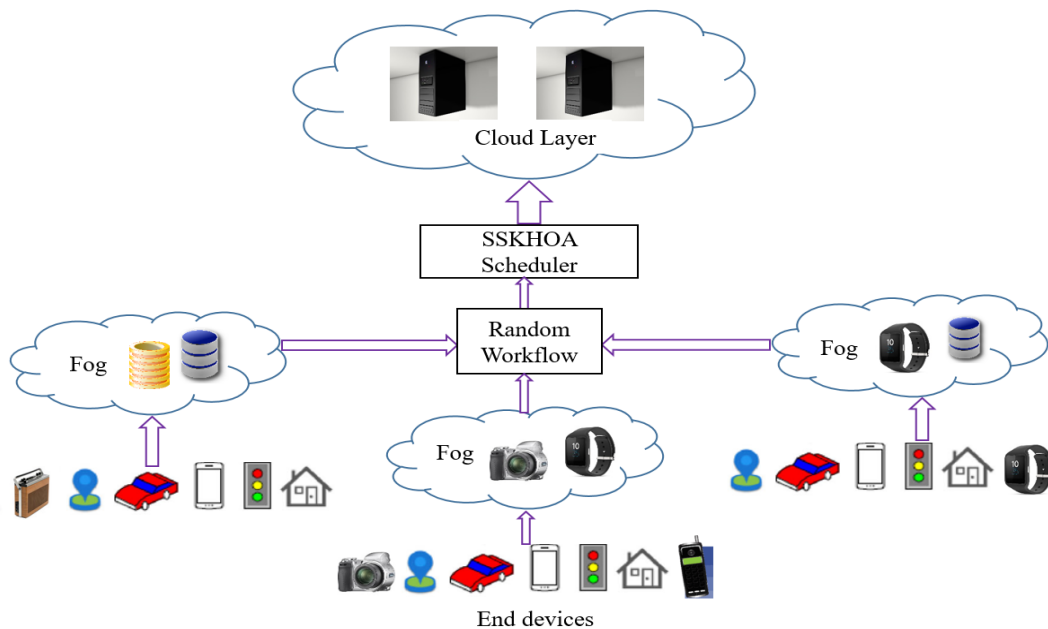


Fig.1. System architecture

- **Cloud Layer:** The cloud layer is an illustration of the centralized computer infrastructure that offers abundant processing resources and storage options. It consists of large-scale data centers and powerful servers. The cloud layer is responsible for handling computationally intensive tasks and managing resource allocation at a macro level.
- **Fog Layer:** Computing resources are brought closer to the network edge by the fog layer, which serves as a bridge between the cloud and edge layers. It comprises fog nodes or fog servers located at strategic points in the network, such as access points or base stations. The fog layer enables low-latency processing, data caching, and real-time analytics. It is responsible for offloading tasks from the cloud layer and distributing them to fog nodes.
- **Edge Layer:** The edge layer is made up of devices that are closer to the clients, like Internet of Things (IoT) devices, sensors, and mobile phones. These devices have limited computational capabilities, storage capacity,

and intermittent connectivity. The edge layer facilitates data pre-processing, local analytics, and real-time decision-making. It reduces the latency by processing tasks near the data source.

The task scheduling process in the cloud-fog-edge architecture using the SSKHOA algorithm involves the following steps:

- **Task Description and Characterization:** Tasks are described and characterized based on their computational requirements, priority, and dependencies. The characteristics include task size, estimated execution time, and resource requirements.
- **Resource Monitoring:** Resource monitoring involves continuously gathering information about the availability and status of resources in the fog and edge layers. This information includes computing capacity, energy level, and network bandwidth.
- **Task Offloading and Allocation:** Based on the task characteristics and resource information, the SSKHOA algorithm is applied to determine the optimal task allocation strategy. The algorithm considers various factors such as task characteristics, resource availability, network conditions, and energy consumption. It utilizes the swarm intelligence of krill herd and the predator-prey behavior of sharks to efficiently explore the solution space and find near-optimal task schedules.
- **Task Execution and Monitoring:** Tasks are executed on the allocated resources, which can be in the cloud, fog, or edge layers. The progress and status of task execution are continuously monitored.
- **Dynamic Adaptation:** The cloud-fog-edge architecture allows for dynamic adaptation and resource reconfiguration based on changing workload patterns, resource availability, and user demands. The SSKHOA algorithm can be applied periodically or triggered by certain events to optimize the task scheduling process.

$$CF_T = (CF_{t1}, CF_{t2}, CF_{t3}, \dots, CF_{tm}) \quad (1)$$

The above equation depicts the CF represent the Cloud and fog, T represents the tasks. Although, the total number of cloud and fog tasks are equal to begin with CF_{t1} to CF_{tm} . The integration of the SSKHOA algorithm into the cloud-fog-edge three-level architecture for task scheduling in cloud fog computing enables efficient allocation of tasks to appropriate resources. It considers the heterogeneity, dynamic nature, and resource constraints of cloud, fog, and edge layers. By optimizing task allocation and resource utilization, this architecture enhances system performance, reduces latency, and improves user experience in cloud fog computing environments.

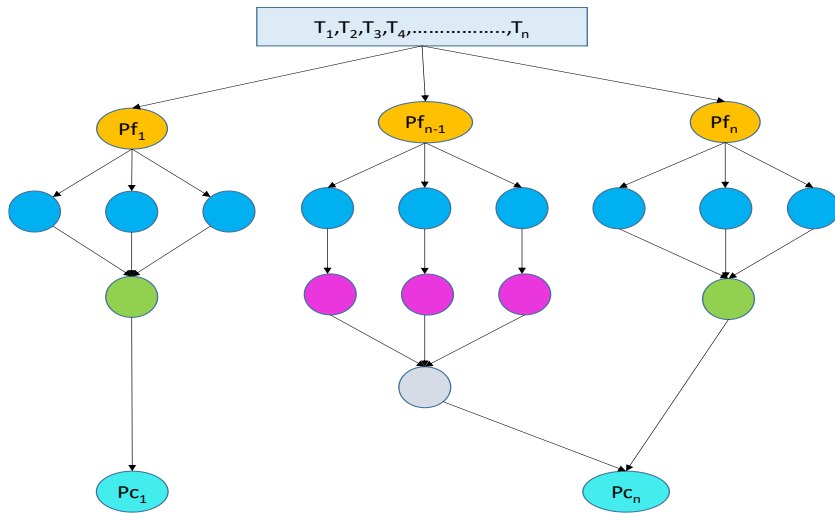


Fig.2. Scheduling of tasks using random workflow

In cloud fog computing, figure 2 depicts tasks often follow a random workflow, where the sequence and dependencies of tasks are not predetermined. In this scenario, tasks are dynamically generated and arrive at the cloud or fog nodes without a specific order. Each task may have its own computational requirements, deadlines, and dependencies on other tasks [26,27]. This random workflow poses challenges for task scheduling algorithms as they need to allocate resources, prioritize tasks, and ensure timely completion based on changing task characteristics and dependencies. Efficient task scheduling techniques that can handle the random workflow effectively are crucial to optimizing resource utilization and meeting performance objectives in cloud fog computing environments.

$$Total_{nodes} = Cloud_{nodes} + Fog_{nodes} \quad (2)$$

3.2. Problem Formulation

The below table 2 depicts the mathematical modelling notations used in problem formulation.

Table 2. Definition of mathematical notations

| Notation used | Meaning |
|---------------|--------------------------------|
| T | Total number of tasks |
| T_1 | First task |
| T_n | N^{th} task |
| C_{nodes} | Cloud nodes |
| F_{nodes} | Fog nodes |
| MKS | Makespan |
| VM | Virtual machine |
| T_{max} | Maximum time of task execution |
| T_{min} | Minimum time of task execution |
| T_{avg} | Average time of task execution |
| Pf | Processors |
| ET | Execution time |
| IT | Number of iterations |
| EV | Fitness value |
| N | Population size |

Makespan

Since it indicates the total length of time required to complete all jobs, the makespan is a crucial metric in cloud fog computing task scheduling. Reducing the makespan increases the system's overall efficiency and aids in making the best use of its resources. The problem can be phrased like this:

- Set of tasks: $T = \{t1, t2, \dots, tn\}$, where each task ti has a unique identifier, computational requirements, and dependencies.
- Set of resources: $R = \{r1, r2, \dots, rm\}$, where each resource rj has a unique identifier, computational capacity, and energy constraints.
- Task dependencies: $D = \{d1, d2, \dots, dk\}$, where each dependency di represents a relationship between two tasks.
- Task Dependency Constraint: Precedence constraints between tasks must be satisfied: $si + \text{execution time of task } ti \leq sj$, for all $ti, tj \in T$ and $di = (ti, tj) \in D$.
- Resource Constraint: Energy constraint for each resource must be satisfied: $\sum_i (x_{ij} \times \text{computational requirement of task } ti) \leq \text{energy constraint of resource } rj$, for all $j \in R$.
- Task Start Time Constraint: Start time of each task must be non-negative and respect task dependencies: $si \geq 0$, for all $i \in T$. $si + \text{execution time of task } ti \leq sj$, for all $ti, tj \in T$ and $di = (ti, tj) \in D$.

The objective is to find an optimal task allocation and start time schedule that minimizes the makespan while satisfying the constraints. The Shark Search Krill Herd Optimization Algorithm is used for exploring the space of potential solutions and continuously fine-tuning the best one for scheduling tasks. The technique is designed to effectively locate near-optimal solutions by finding a balance between exploring and exploiting the search space.

$$MKS = \max(\text{Task Completion Times}) \quad (3)$$

By solving this problem using the Shark Search Krill Herd Optimization Algorithm, an optimized task schedule can be obtained, leading to improved resource utilization, reduced makespan, and enhanced system performance in cloud fog computing environments.

Total Execution Time

Task reduction time refers to minimizing the overall time required to complete a set of tasks in cloud fog computing by optimizing task scheduling. The goal is to reduce the makespan, which directly contributes to task reduction time. The problem can be formulated as follows:

- Set of tasks: $T = \{t1, t2, \dots, tn\}$, where each task ti has a unique identifier, computational requirements, and

dependencies.

- Set of resources: $R = \{r1, r2, \dots, rm\}$, where each resource r_j has a unique identifier, computational capacity, and energy constraints.
- Task dependencies: $D = \{d1, d2, \dots, dk\}$, where each dependency d_i represents a relationship between two tasks.

$$ET = IT * (EV * N) \quad (4)$$

The objective is to find an optimal task allocation and start time schedule that minimizes the makespan, thereby reducing the task completion time. Task scheduling is improved by employing the Shark Search Krill Herd Optimization Algorithm to investigate the space of possible optimizations and make incremental improvements. The method seeks to strike a balance between exploring and exploiting the search space in order to rapidly locate near-optimal closure.

By solving this problem using the Shark Search Krill Herd Optimization Algorithm, an optimized task schedule can be obtained, leading to minimized makespan and reduced task completion time in cloud fog computing environments.

Task Success Rate

Task success rate refers to the percentage of successfully completed tasks in cloud fog computing through efficient task scheduling using the Shark Search Krill Herd Optimization (SSKHOA) algorithm. The goal is to maximize the success rate by ensuring that tasks are allocated and executed successfully within their deadlines. The problem can be formulated as follows:

- Set of tasks: $T = \{t1, t2, \dots, tn\}$, where each task t_i has a unique identifier, computational requirements, and dependencies.
- Set of resources: $R = \{r1, r2, \dots, rm\}$, where each resource r_j has a unique identifier, computational capacity, and energy constraints.
- Task dependencies: $D = \{d1, d2, \dots, dk\}$, where each dependency d_i represents a relationship between two tasks.
- Task deadlines: $Dl = \{dl1, dl2, \dots, dln\}$, where each deadline d_{li} represents the maximum allowed completion time for task t_a .

$$Successrate = \frac{Total_tasks}{Numberof_attempts} \quad (5)$$

The objective is to find an optimal task allocation and start time schedule that maximizes the task success rate, ensuring that a higher percentage of tasks are completed within their respective deadlines. We use the Shark Search Krill Herd Optimization Algorithm to examine the space of potential schedulers for tasks and then iteratively enhance the best one we find. It is the goal of the algorithm to efficiently identify near-optimal solutions by achieving a balance between exploring and exploiting the search space.

By solving this problem using the Shark Search Krill Herd Optimization Algorithm, an optimized task schedule can be obtained, leading to a higher task success rate in cloud fog computing environments. Apart from task scheduling, other factors, such as network connectivity, resource availability, and task characteristics, can affect the success rate of tasks in cloud fog computing. The proposed formulation focuses specifically on optimizing task scheduling to maximize the task success rate within the given constraints and assumptions.

Proposed SSKHOA Algorithm

Input: Set of tasks T, Set of resources R

- Initialize the population of shark and krill individuals
- Repeat until termination condition is met:
- Perform the shark search operation:
- Update the position of each shark individual based on the fitness value
- Update the best solution based on the maximum fitness value
- Perform the krill herd operation:
- Update the position of each krill individual based on the fitness value
- Update the best solution based on the maximum fitness value
- Evaluate the fitness value for each individual in the population:
- Calculate the makespan, task success rate, and other relevant metrics for each candidate solution
- Update the best solution based on the maximum fitness value
- Perform the exploration and exploitation operations:

- Balance the exploration and exploitation of the search space by updating the positions of the individuals
- Perform the elimination-dispersion operation:
- Eliminate inferior individuals and disperse individuals to diversify the population
- Perform the reproduction operation:
- Generate new offspring individuals based on the parents' positions and fitness values

Output: Best task scheduling solution obtained

4. Results and Discussion

4.1. Results

We used the CloudSim tool for simulating the cloud environment to implement the suggested approach. A simulation approach has been applied to conduct a large number of experiments. We provide evaluation comparisons with the popular LTRA, DRL and DAPSO algorithms to prove that the SSKHOA methodology works.

To obtain SSKHOA's results, we performed simulations using a system equipped with an Intel Core i5-3373U processor clocking in at 1.8 GHz and 6 GB of RAM. We used CloudSim Toolkit to implement the SSKHOA algorithm and analyzed its performance in terms of both cost and makespan. In the study, we only investigated tasks that were not dependent on results from other tasks. It was assumed that the data-transfer rates of the links would be regularly distributed, somewhere between 40 Mbps and 10,000 Mbps. The simulation analysis parameters are listed in Table 3. The number of tasks, VMs, and data centers required to conduct the simulation are all listed in the table.

Table 3. Cloudsim toolkit parameters

| Parameter | Cloud | Fog |
|------------------------|--------------|-------------|
| Number of VMs | [10,15,20] | [15,20,35] |
| Computing power (MIPS) | [3000:5000] | [1000:2000] |
| RAM (MB) | [5000:20000] | [250:5000] |
| Bandwidth (Mbps) | [512:4096] | [128:1024] |

A. Makespan

The proposed SSKHOA approach effectively achieves the greatest possible makespan compared to all other existing heuristics such as, LTRA, DRL and DAPSO algorithms as can be seen in the graph depicting the best makespan results for real-time workloads with changing task sizes in fig. 3, the suggested SSKHOA consistently delivered improved makespan results across all scenarios. It's determined that SSKHOA is a great way to deal with critical issues in cloud fog task allocation scheduling.

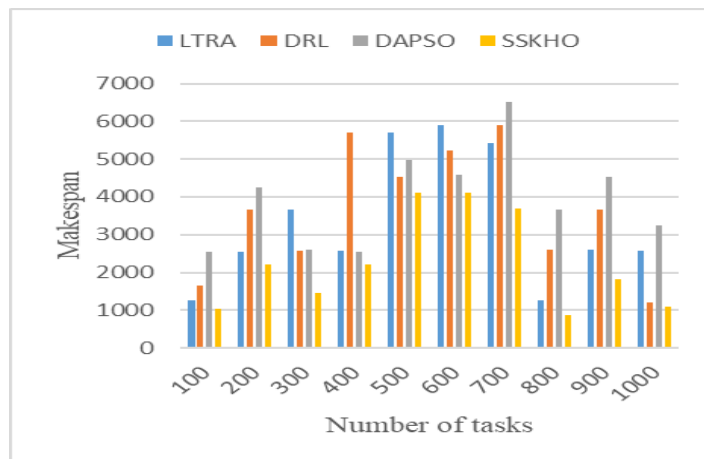


Fig.3. Proposed approach makespan calculation

B. Execution Time

Figure 4 shows that the suggested approach SSKHOA decreases execution time overall, especially for multiple tasks. The SSKHOA scheduler utilizes a combination of the SSO and KHO algorithms to allocate resources; this ensures that no time is wasted looking and that all jobs are provided with nearly optimal VM. When compared to LTRA, SSKHOA is roughly 17 percent more efficient. DRL and DAPSO can get tasks done quickly when there are not many requests. However, as the number of tasks grows, the time it takes for these algorithms to complete grows exponentially.

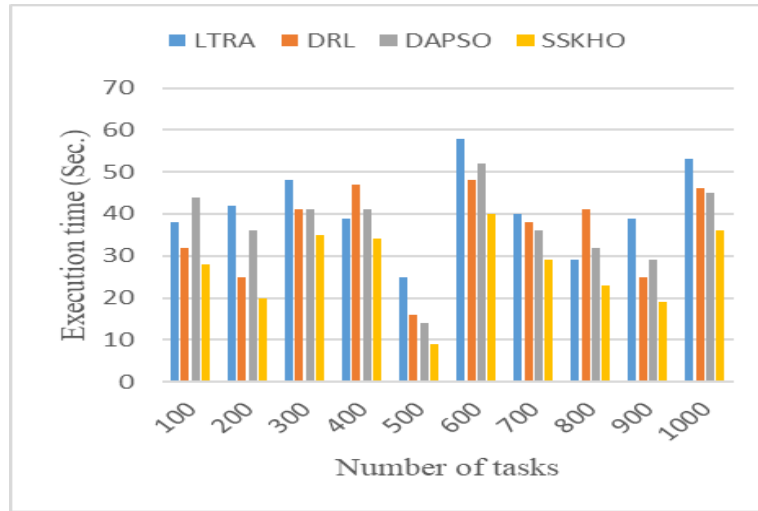


Fig.4. Proposed approach execution time calculation

C. Success Rate

Figure 5 demonstrates the effectiveness of the proposed SSKHOA approach in improving the success rate of task scheduling, particularly for scenarios involving multiple tasks. By leveraging a combination of the Shark Search Optimization (SSO) and Krill Herd Optimization (KHO) algorithms for resource allocation, the SSKHOA scheduler ensures efficient utilization of resources, minimizing wastage and providing nearly optimal virtual machine (VM) allocation for all tasks. Comparatively, SSKHOA outperforms the LTRA algorithm by approximately 98 percent in terms of efficiency. Although algorithms like DRL and DAPSO perform well in low task request circumstances, their performance rapidly deteriorates as the number of tasks rises. As a stable and scalable solution for task scheduling in cloud computing, SSKHOA, on the other hand, keeps constant success rates despite a higher workload.

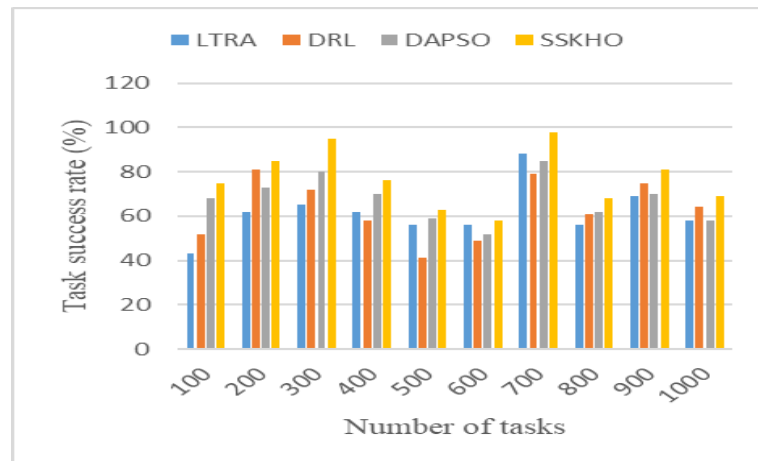


Fig.5. Proposed approach task success rate calculation

4.2. Discussion

Makespan: The research results demonstrate that in the cloud fog computing setting up, SSKHOA significantly decreases the makespan. makespan is the total time needed to accomplish all activities. Through better scheduling and distribution of resources, SSKHOA can minimize the makespan, leading to improved system performance and resource utilization.

Task Success Rate: The experiments demonstrate that SSKHOA improves the task success rate. It considers task dependencies, constraints, and deadlines during scheduling, ensuring that tasks are completed within their specified time limits. By managing task dependencies effectively, SSKHOA minimizes the number of missed or delayed tasks, leading to a higher task success rate.

Total Execution Time: The findings show that SSKHOA effectively lowers the overall execution time needed to do all jobs in the cloud fog computing system. It increases efficiency and performance by minimizing the total amount of time

needed to complete the jobs by optimizing task scheduling selections.

Limitations

While the Shark Search Krill Herd Optimization Algorithm (SSKHOA) offers several advantages for task scheduling in cloud fog computing, it also has certain limitations that should be considered. Here are some limitations associated with the application of SSKHOA:

- **Convergence Speed and Exploration-Exploitation Balance:** SSKHOA's convergence speed can be influenced by the size and complexity of the problem instance. In some cases, it may require a larger number of iterations to converge to an optimal or near-optimal solution. Achieving the right balance between exploration (to search for diverse solutions) and exploitation (to refine the best solutions) can also be challenging, impacting the convergence speed.
- **Sensitivity to Parameter Settings:** Like many optimization algorithms, SSKHOA has certain parameters that need to be set appropriately for optimal performance. The sensitivity of SSKHOA to parameter settings can impact its effectiveness in finding high-quality task schedules. Fine-tuning the parameters or using adaptive approaches can help mitigate this limitation.
- **Limited Scalability:** SSKHOA's performance may degrade as the number of tasks and resources increases significantly. The algorithm's scalability is influenced by the search space size, computational complexity, and memory requirements. Large-scale task scheduling problems may pose challenges in terms of computational time and resource consumption.
- **Lack of Guarantee for Global Optimum:** SSKHOA, like most heuristic algorithms, does not guarantee finding the global optimum for complex optimization problems. It provides near-optimal solutions based on the search space exploration and exploitation. While SSKHOA can converge to good solutions, there is no assurance of finding the absolute best task schedules.
- **Dependency on Problem Representation and Encoding:** The effectiveness of SSKHOA is influenced by the problem representation and encoding used to define the task scheduling problem. The selection of suitable representations and encodings is crucial for capturing the problem's characteristics and ensuring compatibility with the algorithm's operations.

To mitigate these limitations, researchers and practitioners can explore hybrid approaches that combine SSKHOA with other optimization techniques or incorporate domain-specific knowledge. They can also investigate parameter tuning techniques, parallel computing, and adaptability mechanisms to enhance the algorithm's performance and scalability. Additionally, considering the specific requirements and constraints of the cloud fog computing environment can help tailor SSKHOA to address real-world challenges effectively.

5. Conclusions and Future Work

In conclusion, the application of the Shark Search Krill Herd Optimization Algorithm (SSKHOA) for task scheduling in cloud fog computing showcases promising results. SSKHOA effectively minimizes the makespan, reducing the total time required to complete tasks and improving system performance. The experimental results demonstrate that the SSKHOA outperformed the baseline algorithms in terms of task success rate increased 34%, reduced the execution time by 36%, and reduced makespan time by 54% respectively. It enhances the task success rate by considering dependencies and constraints, ensuring tasks are completed within their deadlines. Additionally, SSKHOA optimizes the total execution time, improving efficiency in task completion. While these results demonstrate the potential of SSKHOA in cloud fog computing, further research is needed to validate and generalize these findings across diverse problem instances and benchmark datasets.

Acknowledgement

I would like to express my sincere gratitude to my Ph.D. research supervisor Dr. Ganesh Reddy Karri for his valuable and constructive suggestions during the planning and development of this research work.

Financially, this work does not support any organization/funding agency.

References

- [1] Nguyen, Binh Minh, et al., "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment", *Applied Sciences*, Vol. 9, No. 9, pp. 1730, 2019. DOI: 10.3390/app9091730
- [2] Ghasempour, Alireza., "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges", *Inventions*, Vol. 4, No. 1, pp. 22, 2019. DOI: 10.3390/inventions4010022
- [3] Fu, Jun-Song, et al., "Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing", *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 10, pp. 4519-4528, 2018. DOI: 10.1109/TII.2018.2793350

- [4] Zuo, Liyun, et al., "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing", *Ieee Access*, Vol. 3, pp. 2687-2699, 2015. DOI: 10.1109/ACCESS.2015.2508940
- [5] Lin, Bing, et al., "A pretreatment workflow scheduling approach for big data applications in multicloud environments", *IEEE Transactions on Network and Service Management*, Vol. 13, No. 3, pp. 581-594, 2016. DOI: 10.1155/2020/8105145
- [6] Lin, Xue, et al., "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment", *IEEE Transactions on Services Computing*, Vol. 8, No. 2, pp.175-186, 2014. DOI: 10.1109/TSC.2014.2381227
- [7] Cheng, Feng, et al., "Cost-aware job scheduling for cloud instances using deep reinforcement learning", *Cluster Computing*, pp. 1-13, 2022. DOI: 10.1007/s10586-021-03436-8
- [8] Zhou, Zhou, et al., "A modified PSO algorithm for task scheduling optimization in cloud computing", *Concurrency and Computation: Practice and Experience*, Vol. 30, No. 24, pp. e4970, 2018. DOI: 10.1002/cpe.4970
- [9] Jangu, Nupur, and Zahid Raza., "Improved Jellyfish Algorithm-based multi-aspect task scheduling model for IoT tasks over fog integrated cloud environment", *Journal of Cloud Computing*, Vol. 11, No. 1, pp. 1-21, 2022. DOI: 10.1186/s13673-019-0174-9
- [10] Singh, Gyan, and Amit K. Chaturvedi., "Hybrid modified particle swarm optimization with genetic algorithm (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization", *Cluster Computing*, pp. 1-18, 2023. DOI: 10.1371/journal.pone.0003197
- [11] Zahra, Movahedi, Defude Bruno, and Amir mohammad Hosseininia., "An efficient population-based multi-objective task scheduling approach in fog computing systems." *Journal of Cloud Computing*, Vol. 10, No. 1, 2021. DOI: 10.1016/j.jocs.2023.102152
- [12] Iftikhar, Sundas, et al., "HunterPlus: AI based energy-efficient task scheduling for cloud-fog computing environments", *Internet of Things* Vol. 21, pp. 100667, 2023. DOI: 10.1016/j.iot.2022.100674
- [13] Yin, Zhenyu, et al., "A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing", *Sensors*, Vol. 22, No. 4, pp. 1555, 2022. DOI: 10.3390/s22041555
- [14] Pham, Xuan-Qui, et al., "A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing", *International Journal of Distributed Sensor Networks*, Vol. 13, No. 11, pp. 1550147717742073, 2017. DOI: 10.1177/1550147717742073
- [15] Mangalampalli, Sudheer, Ganesh Reddy Karri, and Mohit Kumar., "Multi objective task scheduling algorithm in cloud computing using grey wolf optimization", *Cluster Computing*, pp. 1-20, 2022. DOI: 10.1109/JIOT.2023.3291367
- [16] Hosseinioun, Pejman, et al., "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm", *Journal of Parallel and Distributed Computing*, Vol. 143, pp. 88-96, 2020. DOI: 10.1016/j.jpdc.2020.04.008
- [17] Liu, Lindong, et al. "A task scheduling algorithm based on classification mining in fog computing environment." *Wireless Communications and Mobile Computing*, 2018. DOI: 10.1155/2018/2102348
- [18] Bakshi, Mohana, Chandreyee Chowdhury, and Ujjwal Maulik., "Cuckoo search optimization-based energy efficient job scheduling approach for IoT-edge environment", *The Journal of Supercomputing*, pp. 1-29, 2023. DOI: 10.3390/s23052445
- [19] Badri, Sahar, et al., "An Efficient and Secure Model Using Adaptive Optimal Deep Learning for Task Scheduling in Cloud Computing", *Electronics*, Vol. 12, No. 6, pp. 1441, 2023. DOI: 10.3390/electronics12061441
- [20] Ahmed, Omed Hassan, et al., "Using differential evolution and Moth-Flame optimization for scientific workflow scheduling in fog computing", *Applied Soft Computing*, Vol. 112, pp. 107744, 2021. DOI: 10.1016/j.asoc.2021.107744
- [21] Mangalampalli, Sudheer, Sangram Keshari Swain, and Vamsi Krishna Mangalampalli. "Multi objective task scheduling in cloud computing using cat swarm optimization algorithm." *Arabian Journal for Science and Engineering* 47.2, 2022: 1821-1830. DOI: 10.1007/s13369-021-06076-7
- [22] Sindhu, V., and M. Prakash., "Energy-efficient task scheduling and resource allocation for improving the performance of a cloud-fog environment", *Symmetry*, Vol. 14, No.11, pp. 2340, 2022. DOI: 10.3390/sym14112340
- [23] Kumar, M. Santhosh, and Ganesh Reddy Karri., "Eeoa: cost and energy efficient task scheduling in a cloud-fog framework", *Sensors*, Vol. 23, No. 5, pp. 2445, 2023. DOI: 10.3390/s23052445
- [24] Moharram, Mohammed Abdulmajeed, and Divya Meena Sundaram., "Spatial-spectral hyperspectral images classification based on Krill Herd band selection and edge-preserving transform domain recursive filter", *Journal of Applied Remote Sensing*, Vol. 16, No. 4, pp. 044508-044508, 2022. DOI: 10.21203/rs.3.rs-1539336/v1
- [25] Kumar, M. Santhosh, and Ganesh Reddy Kumar. "EAEFA: An Efficient Energy-Aware Task Scheduling in Cloud Environment." *EAI Endorsed Transactions on Scalable Information Systems*, 2024. DOI: 10.4108/eetsis.3922
- [26] Kumar, M. Santhosh, and Ganesh Reddy Karri., "Parameter Investigation Study on Task Scheduling in Cloud Computing", 2023 12th International Conference on Advanced Computing (ICoAC). IEEE, pp. 1-7, 2023. DOI: 10.1109/ICoAC59537.2023.10249529
- [27] Kumar, M. Santhosh, and Ganesh Reddy Karri., "A Review on Scheduling in Cloud Fog Computing Environments", *Workshop on Mining Data for Financial Applications*. Singapore: Springer Nature Singapore, pp. 29-45, 2022. DOI: 10.1007/978-981-99-1620-73

Authors' Profiles



M. Santhosh Kumar received the Master's degree in computer science and Engineering from Sreyas Institute of Engineering and Technology, Hyderabad, India in 2019. He has more than 5 years' experience in both Research and Teaching. he has already published more than 10 Research articles in various reputed journals. His currently pursuing a Ph.D. degree in the School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, India. His research interests are Cloud computing, fog computing and IoT.



Ganesh Reddy Karri received the Ph.D. degree from NIT-Suratkal, Karnataka, India, in 2014. he is currently an Associate Professor in the School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, India. Currently his coordinator for center of excellence in cyber security and also an IEEE member. Currently his guiding for five research scholars. He has more than 10 years' experience in both Research and Teaching. he has already published more than 35 Research articles in various reputed journals. His research interests include Cloud computing, Computer and Network Security, Wireless networks, Data structures and Algorithms and the IoT.



Rakesh Kumar Donthi his currently pursuing a post-doctoral in School of Computer Science and Engineering, at university college of dublin, Ireland. He received the Ph.D. degree from NIT-Patna, Bihar, India, in 2021 also received the Master's degree in computer science and Engineering from JNTUH College of engineering, Hyderabad, India in 2013. He has more than 8 years' experience in both Research and Teaching. he has already published more than 10 Research articles in various reputed journals. His research interests are Machine learning, ontology, semantic web, deep learning and cloud computing.

How to cite this paper: M. Santhosh Kumar, K. Ganesh Reddy, Rakesh Kumar Donthi, "SSKHOA: Hybrid Metaheuristic Algorithm for Resource Aware Task Scheduling in Cloud-fog Computing", International Journal of Information Technology and Computer Science(IJITCS), Vol.16, No.1, pp.1-12, 2024. DOI:10.5815/ijitcs.2024.01.01