DOI: 10.24996/ijs.2024.65.2.33

ISSN: 0067-2904

Task Scheduling in a Cloud Environment Based on Meta-Heuristic Approaches: A Survey

Dina Riadh Abdulrazzaq[1]*, Narjis Mezaal Shati [1], Haider K. Hoomod [2]

[1]*Department of Computer Science, College of Science, Mustannsiriyah University, Baghdad, Iraq*

[2]*Department of Computer Science Department, College of Education, Mustannsiriyah University, Baghdad, Iraq*

Received: 28/9/2022 Accepted: 25/2/2023 Published: 29/2/2024

Abstract

Cloud computing is one of the emerging technologies that expands the boundaries of the internet by using centralized servers to maintain data and resources. It allows users and consumers to use various applications provided by the cloud provider, but one of the major issues is task scheduling. Task scheduling is employed for the purpose of mapping the requests of users to the appropriate resources available. This paper provides a detailed survey of the available scheduling techniques for cloud environments based on six common metaheuristic algorithms. Those algorithms are the Cuckoo Search Algorithm (CSA), Chicken Swarm Optimization (CSO), Genetic Algorithm (GA), Bat Algorithm (BA), Whale Optimization Algorithm (WOA), and Grey Wolf Optimization (GWO). The literature is analyzed from three perspectives: task type, objectives to be optimized, simulation environment, and quality of service performance metrics. In addition, the research gaps and future directions for future investigation are presented.

Keywords: Cloud, task scheduling, meta-heuristic, energy, cost, pareto optimality, single objective, weighted sum, make-span, multi- objective.

جدولة المهام في البيئة السحابية بناءً على نهج الخوارزميات الذكية الفوقية: دراسة استكشافية

دينا رياض عبد الرزاق [1]*, نرجس مزعل شاتي [1], حيدر كاظم حمود [2]

[1] قسم علوم الحاسوب, كلية العلوم, الجامعة المستنصرية, بغداد, العراق

[2] قسم علوم الحاسوب, كلية التربية, الجامعة المستنصرية, بغداد, العراق

الخلاصة

تعد الحوسبة السحابية إحدى التقنيات الناشئة التي توسع حدود الإنترنت باستعمال الخوادم المركزية للحفاظ على البيانات والموارد. إنها تمكن المستخدمين والمستهلكين من استعمال التطبيقات المختلفة التي يوفرها مزود السحابة. لكن إحدى المشاكل الرئيسية أمام هذا هي جدولة المهام. يتم استعمال جدولة المهام لغرض ربط طلبات المستخدمين بالموارد المناسبة المتاحة. تقدم هذه الورقة مسحًا تفصيليًا لتقنيات الجدولة المتاحة للبيئة السحابية بناءً على ست خوارزميات meta-heuristic شائعة: خوارزمية الوقواق (CSA) ، سرب الدجاج (CSO) ،

*Email: dinashibani@uomustannsiriyah.edu.iq

الخوارزمية الجينية (GA) ، خوارزمية الخفاش (BA) ، خوارزمية الحوت (WOA) و Gray Wolf Optimization (GWO). يتم تحليل الدراسات من ثلاث وجهات نظر ، بما في ذلك: نوع المهمة ، والأهداف التي يتعين تحسينها ، وبيئة المحاكاة وجودة مقاييس أداء الخدمة. أيضا ، قدمت الثغرات البحثية والتوجيهات المستقبلية للتحقيق في المستقبل.

1. Introduction

Cloud computing (CC) is a widely distributed technology that provides memory, a central processing unit (CPU), storage, software, and other computing resources. It also provides on-demand services as an online pay-per-use service. Virtualization technology has been used to build and operate the environment of cloud computing. Three different types of services can be offered by CC, such as infrastructure as a service, platform as a service, and software as a service, based on their needs [1].

The resources of the cloud are subject to massive demand for tasks from all over the world. A scheduling strategy is required for each cloud so that the task execution order can be defined and processed. Consequently, scheduling introduces a significant challenge and a growing area of study known as the "scheduling issue." The process of assigning tasks for the provision of appropriate resources is known as "scheduling." It is considered to be an NP-hard problem (i.e., a nondeterministic polynomial time problem). The scheduling in the CC environment is classified into two levels: the first level involves determining the best sequencing order for task execution on a given virtual machine (VM). The second level, also known as "VM scheduling," involves assigning VMs to the appropriate physical machines. VM scheduling consists of two operations: initial VM placement and VM migration. Several optimization objectives related to task scheduling algorithms are considered when designing the algorithms, for example, make-span, minimizing fairness, reducing energy consumption, minimizing response time, and minimizing cost [1, 2].

These optimization objectives are typically divided into two categories based on the use of cloud services: cloud service providers' (CSP) desires and cloud users' desires. User-desired criteria include make-span (completion time), cost, and reliability. Desired criteria for providers include resource utilization, throughput, load balancing, service level agreements (SLAs), and energy efficiency [3].

Meta-heuristic algorithms are very popular these days for task scheduling, and they can find solutions that are close to the optimal solution. Although meta-heuristic algorithms have demonstrated their efficiency in solving a number of standard benchmark problems as well as real-world problems, some of them have some inherent shortcomings, like getting stuck in local optima. Hence, these shortcomings can be overcome either by using an improved version of the algorithm or by hybridization with other algorithms [4, 5].

A clear understanding of scheduling techniques and the different issues related to them is needed in order to develop a fruitful scheduling algorithm. Due to this, the aim of this paper is to introduce the major notions of scheduling by providing a comparative analysis of six nature-inspired meta-heuristic-based task scheduling approaches. This will help academic researchers determine the best strategy for recommending an effective method for scheduling user applications in a cloud environment and focus on the scheduling issues related to cloud computing based on nature-inspired meta-heuristics, rather than the entire distributed system.

The recently accomplished survey has been focused only on meta-heuristic based task and workflow scheduling in respect of the number of scheduling objectives (single, multi), which is considered a limitation to the detailed analysis of the multi-objective scheduling approach. Therefore, we found the need for an article survey to cover more specific details of scheduling objectives in cloud computing, such as multi-objective fitness function type classification, quality of service (QoS) such as energy-based scheduling, and cost-based scheduling.

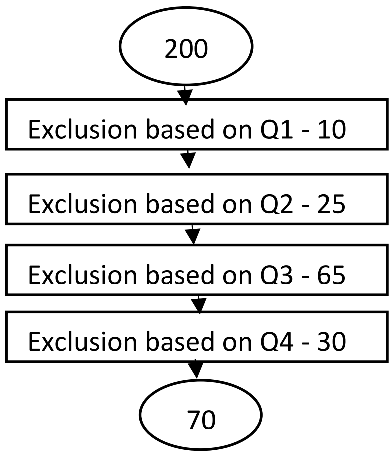
The paper is structured as follows: Section 2 introduces the basic concept of the scheduling objective function; Section 3 introduces the basic concept of task and workflow scheduling; Section 4 introduces energy-based task and workflow scheduling; Section 5 presents cost-based task and workflow scheduling; Section 6 introduces the basic concepts of the meta-heuristic approach to solving task and workflow scheduling, as well as related work for several well-known algorithms; Section 7 provides the survey analysis and discussion.

2. Data Selection Source and Selection Strategy

In this study, combinations of keywords such as (i) “computing cloud” and “task scheduling” and (ii) “workflow scheduling” and “objective” or "metric" were examined. In addition to these keyword combinations, the study needs to include at least one MH technique, including genetic algorithms (GA), gray wolf optimization, chicken swarm optimization, etc., which have been used to obtain fruitful searches in the database scholar articles (Springer, Hindawi, IEEE Explore, Wiley Online Library, etc.). To ensure that only high-quality articles are selected, the studies must have a “yes” answer to the following:

– Is the paper published in peer-reviewed journals?
– Is the paper related to cloud computing scheduling?
– Is the paper related to MH-based task scheduling?
– Is the paper developed a population MH based task scheduling?

Based on the answers to the suggested questions, 130 articles out of 200 have been excluded. The process of papers' selection (exclusion criteria) is summarized in Figure 1, and the selected papers’ classification, both inclusions and exclusions, from the twelve academic data sources is given in Figure 2.



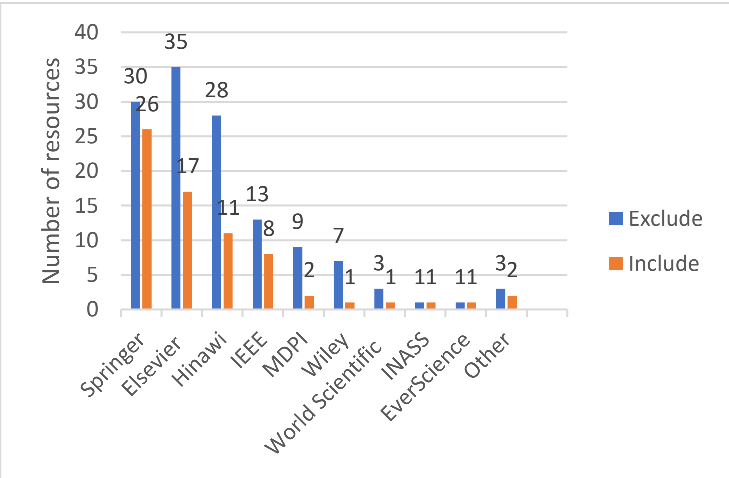


Figure 1: Data selection procedure **Figure 2:** Data source

3. Scheduling Objective Function

Objective functions are designed for the scheduling algorithm. The main aim is to maximize or minimize this objective function according to the criteria defined by the CSP or users. In traditional approaches, a single objective function is considered, but now a multi-criterion objective function is created. In this, two or more parameters are considered simultaneously. With respect to the users, it can be application-centric. Economic cost and make-span, for example, can be application-specific parameters. The designed algorithm should reduce either the economic cost or the manufacturing time, or both. With respect to providers, it can be resource-centric. Economic profit and resource utilization, for example, can be resource-centric parameters. The designed algorithm should maximize either economic profit or resource utilization or both [6].

There are two techniques to deal with the multi-objective function: the a priori or posteriori approach. The priori approach makes decisions before searching (i.e., decide then search); an example of this approach is weighted sum. This class of techniques includes the types of approaches that assume that the decision-maker can implement certain desired, achievable goals or a certain pre-ordering of objectives for research. The posteriori method searches before taking decisions (i.e., search, then decide). These techniques do not require prior preference information from the decision-maker. Pareto optimality is one of the techniques included in this category [7].

4. Independent Task and Workflow Scheduling

Tasks can be run independently with independent scheduling. Tasks are interdependent in the case of workflow scheduling. Dependency indicates that the tasks have precedence orders. Hence the task cannot start until its parents are completed. A directed acyclic graph (DAG) notation is utilized to represent the workflows. Task execution can only be started when all previous tasks in the DAG have been completed. In DAG, tasks are represented by nodes, and edges represent dependencies between them [6, 8].

5. Energy-based Scheduling

The massive computational demands of cloud users increase the energy consumption of cloud data centers, which poses a serious threat to the environment. Energy consumption can be reduced in an appealing way by using efficient scheduling techniques. The energy consumption at cloud datacenters can be reduced by using various scheduling techniques, according to research. During application execution, the CPU consumes approximately 80-90% of the available power [9]. Energy reduction techniques are classified into two categories, as shown in Figure 3. Hardware techniques that include dynamic voltage frequency scaling (DVFS) algorithms [10-16] and software techniques that include VM initial placement and VM migration [12-20].

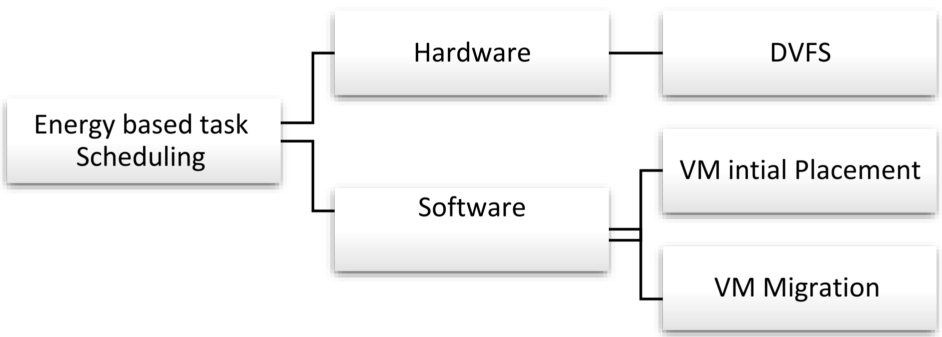


Figure 3: Energy based task scheduling

Mezmaz et al. [10] proposed a hybrid, parallel, bi-objective algorithm based on genetic algorithms (GA) and (DVFS) that takes make-span as well as energy consumption into account for efficient task scheduling. A fast Fourier transformation (FFT) task graph was used to evaluate the presented approach, which is a real-world application. One of the shortcomings of this study is that it considers only the CPU's energy consumption and neglects communications' energy.

For workflow scheduling in cloud computing infrastructures, Yassa et al. [11] present a hybrid algorithm based on combining DVFS and Multi-Objective Discrete Particle Swarm Optimization (DVFS-MODPSO). DVFS-MODPSO optimizes several competing objectives in a discrete space, including make-span, cost, and energy. Worth mentioning that swarm initialization is implemented with the Hetero Earliest Finish Time algorithm (HEFT). Details of the simulation environment are not given where the performance of the algorithm is evaluated, and that is considered a shortcoming of the study.

Srichandan et al. [12] proposed a hybrid multi-objective algorithm that combines the GA and Bacterial Foraging (BF) algorithms to optimize make-span and energy consumption for efficient cloud task scheduling. The outcomes demonstrated that, in terms of stability, solution diversity, and convergence, the proposed algorithm performed better than the other algorithms. However, it required extra timing for the mutation, crossover, chemotaxis, and reproduction processes.

Fu et al. [13] presented a binary PSO (BPSO) algorithm for designing a VM placement strategy to reduce energy consumption. Simulation experiments are used to compare the proposed strategy to other existing strategies. Comparing to the other existing strategies, results indicate that the presented VM placement strategy performs better in terms of reducing energy consumption. On the other hand, PSO is causing a slow convergence rate. These difficulties are transferred to the original BPSO, resulting in the algorithm failing to converge.

For an efficient execution of the workflow application in a cloud environment, Rehani and Garg [14] have proposed a multi-objective hybrid algorithm that is based on combining the non-dominated sorting genetic algorithm (NSGA-II), the efficient non-domination level update mechanism (ENLU), and the DVFS technique to obtain the Pareto optimal solutions with good convergence and more uniform diversity, and that too with less computational overhead. The study’s drawback is that it considered only the CPU for the analysis; other resources, however, need to be considered.

The multi-objective optimization method is presented by Alresheedi [15], which combines the Salp Swarm and Sine-Cosine Algorithms (MOSSASCA) to set a suitable solution for virtual machine placement (VMP). The proposed MOSSASCA objectives are to minimize power consumption, maximize mean time before a host shutdown (MTBHS), and reduce service level agreement violations (SLAVs).

For efficient workflow scheduling in terms of minimizing both the energy consumption and the execution time while maximizing throughput, a meta-heuristic GA named Energy Aware, Time, and Throughput Optimization (EATTO) is based on the Bat Algorithm (BA), which is presented by Gu and Budati [16]. The superiority of the EATTO algorithm over other existing methods is demonstrated by a wide range of performance comparisons; however, communication energy is neglected.

Balaji et al. [17] presented an adaptive cat swarm optimization (CSO) algorithm for optimizing system resources in terms of optimal power usage and load balancing. The proposed method's results demonstrated energy savings and the shortest time required for load balancing. Task size, CPU utilization, and migration cost are not considered.
Shukla [18] presents a hybrid multi-objective optimization model for efficient task scheduling that integrates NSGA-II and DVFS to minimize energy consumption and make-span. When compared to other traditional algorithms, the results analysis shows that the proposed model significantly reduces the make-span and energy consumption. Other essential objectives, such as load balancing, were not considered.

A multi-objective Emperor Penguin Optimization (EPO) algorithm is presented by Samriya et al. [19] for optimal allocation of VMs to the fewest number of active physical machines in cloud data centers in terms of energy reduction. When compared to existing methods, EPO is more efficient in terms of lowering energy consumption, increasing resource utilization, reducing execution time, and lowering the number of active servers. But the conductive analysis takes only CPU resources into consideration to decrease the power consumption. An energy-aware scheduling approach with VM consolidation called EASVMC for workflow scheduling is presented by Medara and Singh [20]. The EASVMC algorithm runs in two phases: task scheduling and VM consolidation (VMC). EASVMC experiments indicated energy consumption improvement and resource utilization. Yet, parameters like make-span and effective QoS are not considered. A brief summary of the mentioned methods is presented in Tables 1 and 2.

6. Cost-based Scheduling

One of the difficult problems in a cloud environment is cost optimization. While users seek to complete tasks as quickly and cheaply as possible, CSPs aim to maximize profit from infrastructure [9]. To reduce the overall cost of scheduling Li et al. [21] have presented an improved version of the PSO algorithm by using a chaotic sequence and an adaptive inertia weight factor. The variety of solutions was improved by the chaotic sequence's high randomness. While the consistency of its occurrence guarantees good global convergence, the estimated value of the cost determines the adaptive inertia weight factor. The disadvantage of the proposed method is that it focuses on the optimal solution for one goal from the user's point of view.

To assign each task to the best VM, Aziza and Krichen [22] provided two bi-objective task scheduling algorithms in CC, which are the space-shared GA (SSGA) and time-shared GA (TSGA) algorithms. Both algorithms are depended on the basic GA to reduce total cost and minimize the make-span. It is worth mentioning that the presented method is only implemented in one data center; there is no implementation with multiple data centers.

In order to minimize the total cost and total execution time of users' applications, Kothyari and Singh [23] have introduced an adaptive privileged multi-objective workflow scheduling algorithm (APMWSA) based on PSO. APMWSA is examined using a computer-based simulation created with the CloudSim toolkit. For task resource mapping, the concept of "novel adaptive elite-based PSO" (NAEB-PSO) is used. It is worth mentioning that a detailed description of workflow is not given.

Choudhary et al. [24] proposed a bi-objective hybridization approach that is a combination of the popular meta-heuristic Gravitational Search Algorithm (GSA) and HEFT for efficient scheduling of workflow applications that consider the minimization of make-span and cost. Monetary Cost Ratio (MCR) and Schedule Length Ratio (SLR) are used to evaluate and compare the performance of the proposed algorithm with the existing algorithms. Variable bandwidth between the VMs was not considered in the experimental results.

For an efficient task's schedule in terms of cost and make-span reduction, a multi-objective task schedule based on an adaptive version of the Artificial Flora (AF) swarm intelligence algorithm is presented by Bacanin et al. [25]. When compared to existing methods, the results show that the presented scheduler outperforms them in terms of make-span and cost. Transmission costs are not considered.

Zhou et al. [26] present two GA-based approaches for hybrid clouds that consider both make-span and monetary costs for efficient workflow scheduling. The first is a single-objective workflow scheduling optimization algorithm, "Deadline Constrained Cost Optimization for Hybrid Clouds (DCOH)," that aims to reduce the monetary cost of scheduling workflows under deadline constraints. The second is a multi-objective workflow scheduling optimization algorithm named "Multi-Objective Optimization for Hybrid Clouds (MOH)," which optimizes both the make-span and the monetary cost of scheduling workflows at the same time. The presented method considers the execution time of tasks in workflow applications to be fixed, and this assumption is not always true for many real-world scenarios.

In order to minimize the execution and transfer costs of executing cloudlets on virtual machines, Chaudhary and Kumar [27] have proposed a cost-optimized hybrid genetic GSA (HG-GSA). The proposed method employs the GA and GSA capabilities to find the next-best-fit position of the particle in the search space for scheduling the load in a cloud computing environment. HG-GSA significantly reduces the total cost of computation compared to the existing algorithms. However, performance parameters like effective QoS and make-span are not considered.

For workflow-efficient load balancing, Kaur and Kaur [28] have presented two hybridized approaches. The first one is the hybrid Predict Earliest Finish Time (PEFT) heuristic with Ant Colony Optimization (ACO) meta-heuristic, and the second one is the hybrid HEFT heuristic with ACO. Both HEFT and PEFT are used to initialize the seed population for ACO. The ACO is used to detect underutilized VMs. In terms of cost and life span, the hybrid PEFT outperformed the hybrid HEFT, according to the results. Though the proposed approaches have the limitation of offering only self-comparison and no comparison with the state of the art.

Hosseini Shirvani and Noorian Talouki [29] presented a hybrid population for bi-objective optimization using the task duplication algorithm (BOSA-TDA) and simulated annealing (SA) for workflow-efficient, cost-aware scheduling. BOSA-TDA uses two important heuristics and duplication techniques to improve canonical SA. The simulation results reported for different well-known scientific workflows demonstrate that the proposed BOSA-TDA has improvements over other existing approaches in terms of maximizing the make-span, minimizing the monetary cost, and reducing SLA, respectively. The lack of a detailed description of the simulation environment where algorithm performance is evaluated is the main limitation of the proposed method.

For optimized resource allocation and efficient task scheduling, Wan and Qi [30] have presented an Improved Coral Reef Optimization (ICRO). The proposed algorithm employs multiple crossover strategies and load balance aware mutation to increase convergence speed and enhance the load balance among virtual machines by adjusting the number of resources provided to users. According to the findings, ICRO reduced the make-span and scheduling cost while maintaining a better system load balance. Due to structural differences, comparing a meta-heuristic algorithm to basic scheduling algorithms such as First Come, First Served (FCFS) is not reliable.

Bezdan et al. [31] present a hybridized multi-objective meta-heuristic approach based on BA, the BEES search procedure, and the quasi-reflective learning (QRBL) mechanism for reducing financial costs and minimizing make-span. Synthetic workloads and standard parallel workloads are used to examine the presented scheduler. The obtained results are compared to other existing methods that were assessed under the same conditions. Simulation results show the great potential of the presented approach. There is no implementation with multiple data centers. A brief summary of the presented methods is offered in Tables 3 and 4.

7. Related Work of Meta-heuristic Scheduling Algorithm

Many meta-heuristic algorithms are exploited in the field of cloud computing environments to quickly find an approximation to an NP-complete problem. Task scheduling is considered an NP-complete problem since much time is required for an optimal solution with a large solution space. MHs are categorized into three main groups: physics-based, chemistry-based, and nature-inspired, as represented in Figure 4. The first of these classifications is called “physics-based,” which is inspired by physics law and imitates the physics rules in the universe. The second is chemical reaction optimization, which is inspired by chemical reactions. Cuckoo Search Algorithm (CSA) [34-37], Chicken Swarm Optimization (GSO) [38-42], Genetic Algorithm (GA) [44-48], BA [50-53], Whale Optimization Algorithm (WOA) [55-61], and Gray Wolf Optimization (GWO) [63-70] are examples of nature-inspired algorithms. A brief summary of the mentioned methods is given in Tables 5 and 6.

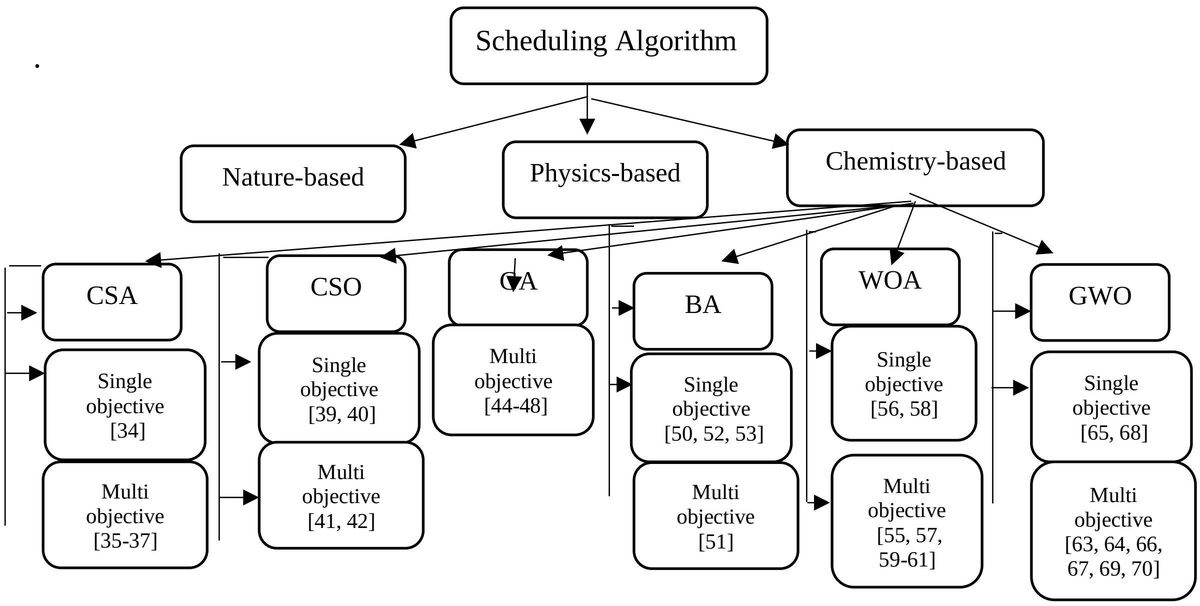


Figure 4: Taxonomy of survey

i.Cuckoo Search Based Task Scheduling

CSA was based on the obligate brood parasitic behavior of some cuckoo species in combination with the Levy flight behavior of some fruit flies and birds [33].

For efficient task scheduling, Agarwal and Srivastava [34] have presented an algorithm that assigns the tasks among the VMs based on their processing power, which is measured in million instructions per second (MIPS), and task length, which depends on CSA. Greedy-based scheduling and first-in-first-out (FIFO) algorithms are used for a comparison issue with the proposed algorithm. The results showed that the proposed CSA outperformed other algorithms in terms of minimizing the execution time. The limitation of this method is that it focuses on the optimal solution for a single goal from the user's point of view.

For optimized task scheduling, Krishnadoss and Jacob [35] have presented a new hybrid algorithm called Oppositional CSA (OCSA). It integrates CSA and opposition-based learning (OBL) to overcome the limitations of CSA. In terms of cost and make-span, the OCSA algorithm outperformed the other task scheduling algorithms. For further analysis of OCSA, more Qos could be integrated.

To improve and optimize scheduling performance and costs, a multi-objective optimization scheduling method (CPSO) is proposed by Prem Jacob and Pradeep [36]. The proposed method merges the CSA and PSO algorithms. Experiments have shown that CPSO has better results compared to baseline scheduling algorithms. The comparison in the proposed method cannot be dependable because the algorithms are not equivalent.

To optimize the energy consumption and workload make-span of the cloud resources Chhabra et al. [37] provided an efficient task's scheduling method by integrating the CSA, DE, and OBL algorithms. The proposed CSDEO algorithm enables the trade-off between exploitation and exploration that is achieved by switching between CS and DE algorithms using single-solution fitness.

ii.Chicken Swarm Based Task Scheduling

X.B. Meng et al. introduced CSO, inspired by the chicken swarm’s behavior while searching for food. Each swarm is divided into several groups consisting of one rooster, hens, and chicks. The fitness value establishes the hierarchical order of the swarm [38].

For an efficient single-object task schedule, aim to reduce both the execution time and the response time, in addition to maximizing the throughput. Torabi and Safi-Esfahani [39] presented a scheduling framework that depends on a hybrid of the improved raven roosting optimization algorithm and the CSO algorithm for solving the problem of premature convergence. The results showed an improvement in the reduction of execution time compared to the existing algorithms, ignoring other criteria such as energy consumption and computational cost.

For an efficient resource schedule, an improved CSO is presented by Han [40]. First, opposition-based learning (OBL) is used to initialize the chicken population and improve the global search ability. Then, the weight value and learning factor in PSO are helping to improve the chicken's positions and optimize the individual chicken's positions. The proposed method did not clarify the simulation environment that was used to evaluate its performance.

The multi-objective Chaotic Quantum Behaved Chicken Swarm Optimization (CQCSO) method for task scheduling is presented by Kiruthiga and Mary Vennila [41]. CQCSO is produced by applying the chaotic and quantum theories to the standard CSO in order to solve local optima and premature convergence problems. Simulation results conducted in one data center showed improvement in the reduction of execution time, increasing both response time and throughput compared with the existing algorithms.

To provide energy-efficient, QoS, and load-balancing-aware task scheduling, Kiruthiga and Vennila [42] introduced an innovative multi-objective Chaotic Darwinian CSO (CDCSO) system. The multi-objective CDCSO algorithm incorporates chaotic and Darwinian theories into the standard CSO to increase its global exploration and maximize the convergence rate. The simulation results outperformed the compared ones; however, performing in only one data center is considered a weakness.

iii.Genetic Algorithm Based Task Scheduling

GA's is based on the biological concept of population generation, which is inspired by Darwin's theory of evolution. GA is used as a method of scheduling that talks about which resource is to be assigned to which task [43].

Zhan et al. [44] presented a hybrid approach that combines Min-Min and Max-Min with GA to optimize load balancing and make-span. Min-Min and Max-Min are adapted for initial population initialization. The Time Load Balance (TLB) model is integrated with make-span to establish the fitness function. The simulation results show that the load-balancing properties are good.

A hybrid approach of ACO and GA is introduced by Cui et al. [45] to optimize cost. In cloud task scheduling, make-span and flow time are considered algorithmic performance criteria. GA operations, including selection, mutation, and cross-over, are used to update pheromones in ACO to overcome the problem of local optima. The results demonstrated that the cost was reduced compared to the existing methods. One can notice a lack of performance due to the use of single data centers.

Manasrah and Ba Ali [46] propose a hybrid GA/PSO approach for workflow cost and make-span scheduling. GA operations, including selection, mutation, and cross-over, are integrated with PSO to overcome the problem of speed convergence, which may lead to local optima. Simulation results offered no convergence analysis to prove that the hybrid algorithm overcomes the problem of the local optima.

For efficient resource allocation, Senthil Kumar and Venkatesan [47] have integrated GA and PSO algorithms (HGPSO). GA is used to initiate the initial population for PSO rather than the output of PSO, which is enhanced by the cross-over and mutation operations of GA. The proposed method offers a fair result in terms of computation time, availability, and scalability. However, it did not include a convergence speed analysis for HGPSO.

For efficient task scheduling in terms of load-balancing, maximization of make-span, maximization of resource utilization, and adaptively minimizing the SLA violation, Mubeen et al. [48] integrated the best features of GA and ACO to provide an efficient hybrid algorithm. The proposed approach has revealed improvements over the existing approaches. At the same time, other feasible improvement concepts, such as energy consumption, are missing.

iv. BAT Algorithm Based Task Scheduling

The original BA simply mimics some of the echolocation properties of a microbat. The basic structure of BA is built using three main properties of the microbe. Echolocation behavior is the first characteristic, and the frequency transmitted by the micro bat is the second feature. The wavelength λ of the frequency f varies is the third characteristic used by the microbat to find prey [49].

Raghavan et al. [50] used the binary version of the original BA for scheduling the workflow in the cloud. The optimal resources are selected such that the overall cost of the workflow is minimal. The proposed method has revealed improvements over the existing methods. The downside is that it misses the details of the simulation environment where the performance of the algorithm is evaluated.

Kaur and Singh [51] provided the BA to solve the multi-objective problem of the workflow scheduling, which reduces the execution time and increases reliability while staying within the user-set budget. The results were compared to the Basic Random Evolutionary Algorithm (BREA), which employs a greedy approach to allocate resources to workflows while taking into account high reliability, low cost, and optimized execution times. Experiments have shown that BA performs better than BREA while the transmission cost is neglected.

To discover the resources and allocate them in a cost-effective manner, Kalaiselvi and Selvi [52] have introduced a hybridization approach based on BA and the Multiple Kernel Fuzzy C Means Clustering Algorithm (MKFCM). The proposed approach comprises two phases, such as (1) resource discovery and (2) resource allocation. The experimental results showed the proposed method performed better than the existing methods. The proposed approach suffers from the loss of other feasible optimization concepts, such as energy consumption.

Jayswal [53] introduced a BA-inspired task allocation algorithm for a cloud infrastructure to improve the performance of the cloud in terms of execution and start time as compared to existing algorithms. The results showed the proposed method performed better than the existing methods. At the same time, it focuses only on the optimal solution for one goal from the user's point of view.

v.Whale Optimization Algorithm Based Task Scheduling

WOA is a recently developed population-based approach that is simple and has a good convergence rate. Randomness in the initial population and within the design parameters impacts the algorithm’s effectiveness. For that, WOA is integrated with other meta-heuristic algorithms [54].

Sreenu and Sreelatha [55] proposed a W-Scheduler algorithm based on the multi-objective model and WOA for optimal task scheduling to the appropriate VM while maintaining the shortest make-span and cost. Experiment results showed that the proposed algorithm outperformed existing techniques in terms of both make-span and cost. Nevertheless, the proposed method is in the local optima.

For an efficient task's scheduling considering make-span optimization, Hemasian-Etefagh and Safi-Esfahani [56] introduced an improved version of the WOA. It proposes a novel approach to avoiding premature convergence by grouping whales. The results demonstrated throughput improvements, minimizing both execution and response times. In other respects, important parameters, such as energy consumption, are not involved in the optimization of scheduling.

Narendrababu Reddy and Phani Kumar [57] have introduced the regressive work-flow optimization algorithm (RWO) for multi-objective workflow scheduling in the cloud computing environment. The difference between the RWO and standard WOA is the position update step. The modification is made by the RWO with the introduction of a regression-based position update. When compared to existing methods, the RWO outperforms them. One can notice that the initial solution is generated randomly without considering the resource type.

Sanaj and Prathap [58] presented an algorithm for an efficient task's scheduling. The suggested algorithm is based on a hybridized map-reducing framework and GA-WOA. First, the client task's task features are extracted. The large-scale tasks are then subdivided using a map-reduce framework. Finally, the GA-WOA algorithm is used to schedule the tasks. The results indicate that GA-WOA achieves improvements in processing time, turnaround time, and throughput compared to other existing algorithms. Again, the main limitation of the proposed algorithm is its focus on the optimal solution to a single goal.

Ni et al. [59] presented a multi-objective task scheduling method based on WOA and a Gaussian cloud model (GCWOAS2) to optimize the completion time and reduce the system operating cost. The OBL mechanism kicks off the scheduling strategy, generating the best initial scheduling solutions. According to the results, GCWOAS2 has perfect performance when processing small and large tasks, and it can reduce the task completion time and balance the virtual machine load. With respect to the operating costs, the proposed method degrades his performance.

For minimizing costs, energy consumption, and the total execution time, a hybridized multi-objective meta-heuristic approach based on GWO and WOA is submitted by Ababneh [60]. The results of the study indicate that the proposed method has the potential to perform at a higher level compared to the original GWO and WOA algorithms in terms of energy consumption, cost, make-span, degree of imbalance, and use of resources. The absence of another potential concept, such as a deadline, is considered a drawback of the proposed approach.

An improved scheduling algorithm based on WOA called IWC is proposed by Jia and Shi [61]. To enhance the local search ability and prevent WOA from reaching premature convergence, the inertial weight strategy is used. The add and delete operators are used to screen individuals after each iteration, which is updated to improve the quality of understanding. The applicable CloudSim configuration, such as the number of data centers and VMs, is not indicated, and that is a weakness.

vi. Grey Wolf Optimization Algorithm Based Task Scheduling

The GWO was inspired by the wolf-hunting behaviors of the Canis lupus types of the Canidae family. The gray wolf (as the top predator) rules the food chain. Hence, it shows the domination of the wolf when hunting for its food (i.e., prey). The grey wolf hierarchy exists in a pack that is led by the alpha, beta, delta, and omega [62].

A multi-objective Pareto-based GWO (PGWO) is proposed by Khalili and Babamir [63] for workflow scheduling to minimize the completion time of dependent tasks and the cost of using VMs while simultaneously maximizing resource throughput. Experimental results showed that PGWO has maximized the spread of solutions, which leads to more proper solution coverage.

Natesha et al. [64] introduced a multi-objective GWO method for task scheduling. The proposed algorithm aims to optimize the use of cloud resources in order to reduce data center energy consumption as well as the make-span of the scheduler for the given list of tasks.

Experiments have shown that the proposed approach outperforms existing algorithms in terms of resource utilization, make-span, and energy consumption. Once again, performance in a single data center is a major limitation.

A single objective task's scheduling algorithm using a meta-heuristics approach is presented by Bacanin et al. [65]. The proposed scheduler is based on the GWO. Despite the fact that the results demonstrate the quality and robustness of the proposed method for optimizing the make-span, there is still a limitation in focusing on the best solution to a single goal from the user's perspective.

For optimizing tasks' scheduling in terms of maximizing the make-span and minimizing energy consumption, Natesan et al. [66] presented a modified version of GWO called mean GWO. The objective of the proposed algorithm has been evaluated by the CloudSim toolkit for a standard workload. The simulation result showed that the proposed algorithm made progress compared to the other existing algorithms. It did not, however, overcome the neglect of CPU switch in, switch off, and VM migration in energy consumption reduction.

The fitness function of GWO is modified by Alzaqebah et al. in [67] by enabling it to handle multiple objectives in a single fitness. The fitness to solve the task scheduling problem has two goals: cost and make-span. The results demonstrated the performance improvement of MGWO. The MGWO also inherited the drawbacks of the original GWO, such as low solving accuracy, bad local searching ability, and a slow convergence rate.

Bansal and Singh [68] proposed a method based on the nature of the gray wolf to achieve an effective independent task schedule in terms of cost. The quality of goodness is calculated by the experiments in the CloudSim simulator. A convergence speed analysis for different numbers of iterations is presented.

In Alsadie [69], the Task Schedule using Multi-Objective GWO (TSMGWO) is used to find near-optimal task scheduling solutions while dealing with the conflicting objectives. The evaluation of TSMGWO is based on three benchmark datasets: GoCJ, HCSP, and Synthetic. The results of the FCFS, PSO, GA, and WOA methods were compared to the results of the proposed method.

To deal with the problem of workflow scheduling, aiming to minimize executions' cost, power consumption, and make-span, an improved binary version of GWO (IGWO) is provided by Mohammadzadeh et al. [70]. The proposed algorithm, based on chaos theory and the hill-climbing method, achieved better results by increasing the GWO convergence speed and preventing it from falling into the local optimum

Table 1: Energy based scheduling comparison

Ref.	*Task Type*	*No. of objective*		*Type of Multi-objective function*		*User / Provider*			*Language*
		Single	***Multi***	***priori***	***posteriori***	***User***	***Provider***	***Both***	
[10]	*Workflow*		√		√			√	*ParadisEO*
[11]	*Workflow*		√		√			√	*-*
[12]	*Workflow*		√		√			√	*MATLAB*
[13]	*Independent*	√					√		*JAVA*
[14]	*Workflow*		√		√			√	*Cloudsim*
[15]	*-*		√		√		√		*Cloudsim*
[16]	*Workflow*		√	√				√	*Python*
[17]	*Independent*		√	√				√	*Cloudsim*
[18]	*Independent*		√		√			√	*-*

[19]	-	√		√	√	Cloudsim+ JAVA
[20]	Workflow	√	√		√	Workflowsim

Table 2: Energy based scheduling evaluation metric

Ref.	Execution	Make-span	Energy Con	Execution	Response	Turnarou	Throuah	Resource	Reliabili	SLA violatio	Availability	Scalability
[10]		√	√									
[11]		√	√	√								
[12]		√	√									
[13]			√									
[14]		√	√						√			
[15]			√							√		
[16]	√		√				√					
[17]		√	√					√				
[18]		√	√									
[19]			√							√		
[20]			√					√				

Table 3: Cost based scheduling comparison

Ref.	Task Type	No. of objective		Type of Multi-objective function		User / Provider			Language
		Single	Multi	priori	posteriori	User	Provider	Both	
[21]	Workflow	√				√			-
[22]	Independent		√	√		√			JAVA+Cloudsim
[23]	Workflow		√	√		√			Cloudsim
[24]	Workflow		√	√		√			C++
[25]	Independent		√	√		√			Cloudsim
[26]	Workflow	√	√		√	√			-
[27]	Independent	√				√			Cloudsim
[28]	Workflow	√				√			Cloud Workflow Simulator (CWS)
[29]	Workflow		√		√	√			-
[30]	Independent		√	√		√			Cloudsim
[31]	Independent		√	√		√			Cloudsim

Table 4 : Cost based scheduling evaluation metric

Ref.	Execution time	Make-span	Energy Consumption	Execution Cost	Response Time	Turnaround Time	Throughput	Resource	Reliability	SLA violation	Availability	Scalability
[21]				√								
[22]		√		√								
[23]		√		√								
[24]		√		√								
[25]		√		√								
[26]		√		√								
[27]				√								
[28]		√		√								
[29]		√		√								
[30]	√	√		√								
[31]		√		√								

Table 5: Meta-heuristic based scheduling comparison

Ref.	Task Type	No. of objective		Type of Multi-objective function		User / Provider			Language
		Single	Multi	priori	posteriori	User	Provider	Both	
[34]	Independent	√				√			Cloudsim
[35]	Independent		√	√		√			Cloudsim
[36]	Independent		√	√		√			Cloudsim
[37]	Independent		√	√				√	Cloudsim+ JAVA
[39]	Independent	√				√			Matlab+Cloudsim
[40]	Independent	√				√			-
[41]	Independent		√	√		√			Cloudsim
[42]	Independent		√	√				√	Cloudsim
[44]	Independent		√	√		√			-
[45]	Independent		√	√		√			-
[46]	Workflow		√	√				√	WorkflowSim
[47]	Independent		√	√				√	Cloudsim
[48]	Independent		√	√				√	Cloudsim
[50]	Workflow	√				√			Cloudsim
[51]	Workflow		√	√				√	-
[52]	Independent	√				√			Workflowsim
[53]	Independent	√				√			Cloudsim+JAVA
[55]	Independent		√	√		√			Cloudsim+NetBean

[56]	Independent	√				√		JAVA+Cloudsim
[57]	Workflow		√	√			√	Matlab+Cloudsim
[58]	Independent	√				√		Cloudsim+JAVA
[59]	Independent		√	√		√		Cloudsim
[60]	Independent		√	√			√	Matlab2016b
[61]	Independent		√	√		√		Cloudsim+JAVA
[63]	Workflow		√		√	√		Cloudsim+matlab
[64]	Independent		√	√			√	WorkflowSim
[65]	Independent	√				√		Python
[66]	Independent		√	√			√	Cloudsim
[67]	Independent		√	√		√		Cloudsim
[68]	Independent	√				√		Cloudsim
[69]	Independent		√		√		√	Cloudsim
[70]	Workflow		√		√		√	Cloudsim

Table 6: Meta-heuristic Evaluation metric

Ref.	Execution time	Make-span	Energy	Execution Cost	Response Time	Turnaround Time	Throughput	Resource Utilization	Reliability	SLA violation	Load Balancing	Availability	Scalability
[34]	√												
[35]		√		√									
[36]		√		√									
[37]		√	√										
[39]	√				√		√						
[40]		√	√	√									
[41]		√			√		√	√					
[42]	√		√	√	√		√				√		
[43]		√											
[44]													
[45]		√	√								√		
[46]	√											√	√
[47]		√						√		√			
[48]													
[50]				√									
[51]	√			√					√				
[52]	√			√				√					
[53]	√												
[54]		√		√									

[55]	√				√		√	
[56]	√		√	√				√
[57]	√			√		√	√	
[58]				√				
[60]		√	√	√				√
[61]	√			√				
[63]		√		√			√	
[64]		√	√					√
[65]		√						
[66]		√	√					
[67]		√		√				
[68]				√				
[69]		√						√
[70]		√	√	√			√	

8. Discussion and Analysis

An overall comparison of the considered workflow and tasks' scheduling schemes is provided in this section. It mainly tries to compare the following factors:

- Applied task type: independent or workflow.
- Simulator: The environment or software used to evaluate the scheduling approaches.
- Evaluation metric: Metrics used to evaluate the scheme's performance.
- Objectives: in terms of objective number (single or multi) and multi-objective function type (prior or posterior).

Visualizing the data in Figure 5 provides a clear picture with respect to Tables 1, 2, 3, 4, and 5, which were used as references. Figure 5 (a) depicts a comparison of the recently proposed schemes based on the task's type. Out of 54 papers considered in this survey, it is found that 34 papers, which is equivalent to 63%, have been scheduling independent tasks; 18 papers, which is equivalent to 34%, have been scheduling workflow tasks; and only 2 papers, which is equivalent to 3%, have been scheduling VM. Therefore, it is observed that the majority of the work is dealing with independent tasks for their simplicity in scheduling as there are no constraints or a specific order to execute them.

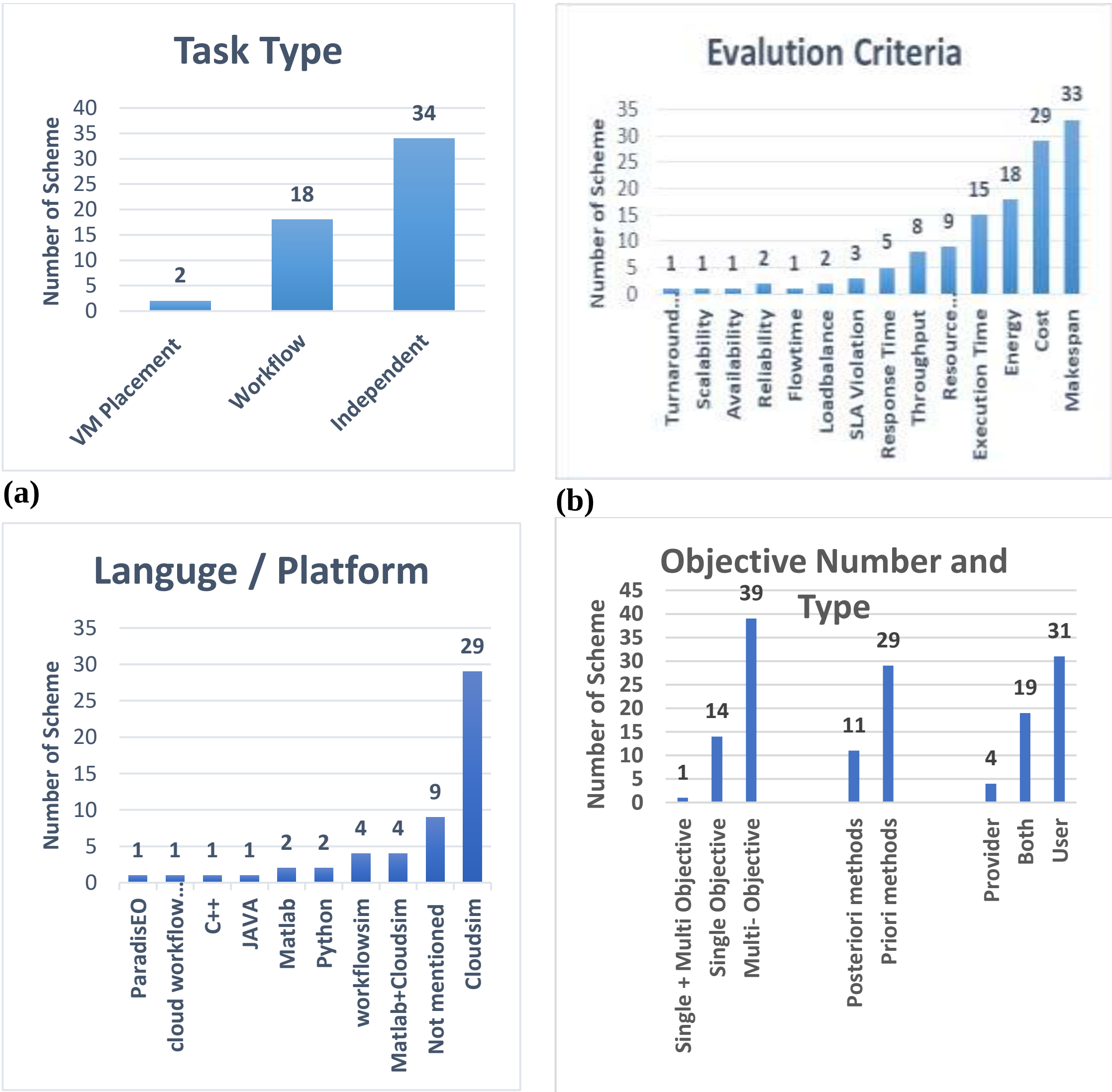
Further, from the study carried out in this survey, it is found that the majority of the work focuses on certain evaluated parameters for a different algorithm and avoids another main evaluated parameter. Out of the 54 papers collected in this survey, the most used is the make-span, which is about 61% of the papers (33), while 53% used the cost as an evaluated parameter, which is about 29 papers, 33% for energy, which is 18 papers, and 27.7% for the execution time, which is about 15 papers. 16.6% resource utilization, 14.8% throughput, and 9.2% response time 5.5% SLV, 3.7% for load balancing, and 2% for flow time, reliability, availability, scalability, and turnaround time, respectively, as illustrated in Figure 5 (b).

For the simulation, it was found that about 54% of the survey articles used CloudSim for simulation, which is around 29 papers; 17% have not mentioned the language that was used for the simulation. The rest are 7% for workflowsim 7% for both fusions of MATLAB and

CloudSim, 4% for Python and MATLAB, and 2% for Java, C++, and ParadisEO, respectively, as shown in Figure 5 (c). Another critical point is the number of objectives.

As shown in Figure 5 (d), about 72% of the papers dealt with multiple objectives, and only 26% dealt with a single objective. 2% provided two algorithms: one for the single objective and the other for the multi-objective. Out of 72% of multi-objective studies, it was found that around 27% used the posteriori approach, while 73% used the priori. Finally, it was found that 57% of the presented papers dealt with user-desired objectives, 35% dealt with both user and CSP objectives, and only 8% dealt with CSP objectives.

Most of the energy-based scheduling reductions focus on reducing the energy of the CPU and neglect other parameters like the energy caused by RAM and storage. While most cost-based scheduling methods concentrate on execution costs while ignoring other parameters such as transmission and memory costs.



(a) **(b)** **(c)** **(d)**

Figure 5: This figure illustrate the statistical analysis of this survey (a) Task type statistical (b) Evaluation criteria statistical (c) Language statistical (d) Objective statistical

9. Gabs and Future Considerations

This survey has found that most of the task scheduling work has focused on the primary optimizer objectives (make-span, cost, and energy) and ignored the least significant objectives. Due to this, a comprehensive scheduling technique is needed to support as many users as possible. Besides, the most existing work tests the transfer and execution times using simulated clouds like CloudSim, and this gives approximate and unrealistic results. Therefore, supporting a real cloud environment, such as vSphere and VMware, is an urgent issue.

On the other hand, security and privacy of user and task information are crucial issues in cloud environments that need to be maintained using meta-heuristic approaches.

10. Conclusions

Providing a variety of resources virtually to run workflows and tasks delivered by users is the aim of cloud computing. In general, scheduling plans are often aimed at allocating customer requests to the suitable VMs in the infrastructure of the computing cloud. However, this process should take into account the conflicting goals of energy reduction, good QoS, lower cost, and efficiency. Scheduling schemes use a variety of single and multi-objective optimization algorithms to achieve a trade-off between several conflicting objectives at the same time. The challenging problems of a single and multi-objective scheduling in various cloud computing environments have been addressed in this paper through a survey. To begin, the paper covers the fundamentals of single and multi-objective algorithms, an independent and dependent task's scheduling, the evaluation criteria, and energy and cost-based scheduling. Then, a classification of the examined single- and multi-objective scheduling schemes based on their features**,** limitations, and optimization algorithms has been presented.

11. Acknowledgements

The work offered in this paper was supported by Mustansiriyah University (www.uomustansiriyah.edu.iq), which is gratefully acknowledged**.**

References

[1] S. Liu , and Y. Yin, "Task scheduling in Cloud Computing Based on Improved Discrete Particle Swarm Optimization," *in Proceedings of IEEE 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, IEEE, pp. 594-597, 2019.

[2] M.A. Alworafi, A. Dhari, S. A. El-Booz, A. Nasr, A. Arpitha and S. Mallappa. (2019). "An enhanced task scheduling in cloud computing based on hybrid approach," in Data *Analytics and Learning*. [Online]. https://doi.org/10.1007/978-981-13-2514-4_2.

[3] R. A. Al-Arasi, and A. Saif, "Task scheduling in cloud computing based on metaheuristic techniques: A review paper," *EAI Endorsed Transactions on Cloud Systems*, *vol. 6, no. 17,* pp.1-18, Jan 2020.

[4] S. Deb, X. Z. Gao, K. Tammi, , K. Kalita, and P. Mahanta, , "Recent Studies on Chicken Swarm Optimization algorithm," *Artificial Intelligence Review*, vol. 53, pp.1737-1765, 2020.[online]. https://link.springer.com/article/10.1007/s10462-019-09718-3.

[5] K. Sreenu and M. Sreelatha, " W-Scheduler: whale optimization for task scheduling in cloud computing," Cluster *Computing*, *vol. 22*, pp.1087-1098, 2017. [Online]. https://link.springer.com/article/10.1007/s10586-017-1055-5.

[6] G. Venu and K. S. Vijayanand, " Task scheduling in cloud computing: a survey," International *Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. *8*, no. 5 pp.2258-2266,2020.

[7] G. Chiandussi, M. Codegone, S. Ferrero and F. E. Varesio," Comparison of multi-objective optimization methodologies for engineering applications, "Computers *& Mathematics with Applications*, vol. 63, no. 5, pp.912-942, 2012.

[8] M. Hosseinzadeh, M. Y. Ghafour, H. K. Hama, B. Vo, and A. Khoshnevis , " Multi-objective task and workflow scheduling approaches in cloud computing: a comprehensive review ," *Journal of Grid Computing*, vol.18, no. 3, pp.327-356, 2020.
[9] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh," A comprehensive survey for scheduling techniques in cloud computing," *Journal of Network and Computer Applications*, vol. 143, pp.1-33, 2019.[online]. https://doi.org/10.1016/j.jnca.2019.06.006
[10] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E. G. Talbi, A. Y. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp.1497-1508, 2011.
[11] S. Yassa, R. Chelouah, H. Kadima and B. Granado," Multi-objective approach for energy-aware workflow scheduling in cloud computing environments," *The Scientific World Journal*, vol. 2013, no. 350934, *2013.*
[12] S. Srichandan, T. A. Kumar and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, vol. *3*, no. 2, pp.210-230, 2018.
[13] X. Fu, Q. Zhao, J. Wang, L. Zhang and L. Qiao," Energy-aware vm initial placement strategy based on bpso in cloud computing,"*Scientific Programming*, *vol. 2018*, no. 9471356, 2018.
[14] N. Rehani, and R. Garg, R., "Meta-heuristic based reliable and green workflow scheduling in cloud computing," International *Journal of System Assurance Engineering and Management*, vol. 9, no. 4, pp.811-820, 2018.
[15] S. S. Alresheedi, S. Lu, M. Abd Elaziz and A. A. Ewees," Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp.1-24, 2019.
[16] Y. Gu and C. Budati," Energy-aware workflow scheduling and optimization in clouds using bat algorithm," *Future Generation Computer Systems*, vol. 113, pp.106-112, 2020.
[17] K. Balaji, P.S. Kiran and M. S. Kumar," An energy efficient load balancing on cloud computing using adaptive cat swarm optimization," *Materials Today: Proceedings*, 2021. [Online]. https://doi.org/10.1016/j.matpr.2020.11.106.
[18] D. K. Shukla, D. Kumar and D. S Kushwaha, "Task scheduling to reduce energy consumption and makespan of cloud computing using NSGA-II," *Materials Today: Proceedings*, 2021. [Online] https://doi.org/10.1016/j.matpr.2020.11.556.
[19] J. K. Samriya, S. Chandra Patel, M. Khurana, P. K. Tiwari and O. Cheikhrouhou, "Intelligent SLA-Aware VM Allocation and Energy Minimization Approach with EPO Algorithm for Cloud Computing Environment," *Mathematical Problems in Engineering*, vol. *2021*, no. 9949995, 2021.
[20] R. Medara, and R.S. Singh, "Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization," *Simulation Modelling Practice and Theory*, vol. 110, pp.102323, 2021. [Online]. https://doi.org/10.1016/j.simpat.2021.102323**.**
[21] X. Li, J. Xu, and Y. Yang, "A chaotic particle swarm optimization-based heuristic for market-oriented task-level scheduling in cloud workflow systems," *Computational intelligence and neuroscience*, vol.2015, no. 718689, 2015.
[22] H. Aziza, and S. Krichen," Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing," *Computing*, vol. 100, no.2, pp.65-91, 2018.
[23] A. Singh, "A multi-objective workflow scheduling algorithm for cloud environment," *in Proceedings of IEEE 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, IEEE, pp. 1-6, 2018.
[24] A. Choudhary, I. Gupta, V. Singh and P.K. Jana, "A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing," *Future Generation Computer Systems*, vol. 83, pp.14-26, 2018. [Online]. https://doi.org/10.1016/j.future.2018.01.005.
[25] N. Bacanin, E. Tuba,T. Bezdan, I Strumberger and M. Tuba," Artificial flora optimization algorithm for task scheduling in cloud computing environment, " in *Proceedings of International Conference on Intelligent Data Engineering and Automated Learning*, Springer, pp. 437-445, 2019.
[26] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei and M. Chen," Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, pp.101631, 2019. [Online]. https://doi.org/10.1016/j.sysarc.2019.08.004.

[27] D. Chaudharyand B. Kumar, "Cost optimized hybrid genetic-gravitational search algorithm for load scheduling in cloud computing," *Applied Soft Computing*, vol.83, pp.105627, 2019. [Online]. https://doi.org/10.1016/j.asoc.2019.105627.
[28] A. Kaur, and B. Kaur," Load balancing optimization based on hybrid Heuristic-Metaheuristic techniques in cloud environment," Journal *of King Saud University-Computer and Information Sciences*, vol. 34, no. 3,pp. 813-824 ,2019.
[29] M. Hosseini Shirvani and R. Noorian Talouki," Bi-objective scheduling algorithm for scientific workflows on cloud computing platform with makespan and monetary cost minimization approach," *Complex & Intelligent Systems*, vol. 8, no. 2, pp.1085-1114, 2022.
[30] S. Wan, S. and Qi, L., "An Improved Coral Reef Optimization-Based Scheduling Algorithm for Cloud Computing," *Journal of Mathematics*, vol. 2021, no. 5532288, 2021.
[31] T. Bezdan, M. Zivkovic,N. Bacanin, L. Strumberger, E. Tuba and M. Tuba, "Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm", *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 1, pp.411-423, 2022.
[32] S. Li, W. Gong and Q. Gu," A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models," *Renewable and Sustainable Energy Reviews*, vol. 141, pp.110828, 2021.
[33] N. J. Navimipour and F. S. Milani," Task scheduling in the cloud computing based on the cuckoo search algorithm," *International Journal of Modeling and Optimization*, vol. 5, no. 1, pp.44, 2015.
[34] M. Agarwal and G. M. S. Srivastava, A cuckoo search algorithm-based task scheduling in cloud computing. In Janusz Kacprzyk (eds.) .*Advances in computer and computational sciences*, pp. 293-299, Springer, 2018.
[35] P. Krishnadoss and P. Jacob," OCSA: task scheduling algorithm in cloud computing environment," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 3, pp.271-279, 2018.
[36] T. Prem Jacob and K. Pradeep," A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization", *Wireless Personal Communications*, *vol. 109*, no.1, pp.315-331, 2019.
[37] A. Chhabra, G. Singh and K. S. Kahlon," Multi-criteria HPC task scheduling on IaaS cloud infrastructures using meta-heuristics", *Cluster Computing*, vol. 24, no. 2, pp.885-918, 2021.
[38] S. C. B. Semlali, M. E. Riffi and F. Chebihi, "Memetic chicken swarm algorithm for job shop scheduling problem", *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, p.2075, 2019.
[39] S. Torabi and F. Safi-Esfahani," A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing," *The Journal of Supercomputing*, vol. 74, no. 6, pp.2581-2626, 2018.
[40] L. Han," Chicken Swarm Optimization in Task Scheduling in Cloud Computing," *International Journal of Performability Engineering*, vol. 15,no. 7, pp.1929, 2019.
[41] G. Kiruthiga and S. M. Vennila, Multi-objective Task Scheduling Using Chaotic Quantum-Behaved Chicken Swarm Optimization (CQCSO) in Cloud Computing Environment. In F. Xhafa (eds.). In Evolutionary Computing and Mobile Sustainable Networks, pp. 803-814, Springer, 2020.
[42] G. Kiruthiga and S. M. Vennila," Energy efficient load balancing aware task scheduling in cloud computing using multi-objective chaotic Darwinian chicken swarm optimization," *Int J Comput Netw Appl (IJCNA)*, vol. 7, pp.82-92, 2020.
[43] R. Kaur and S. Kinger, "Enhanced genetic algorithm based task scheduling in cloud computing," *International Journal of Computer Applications*, vol. 101, no. 14, pp.1-6, 2014.
[44] Z. H. Zhan, G. Y. Zhang, Y. J. Gong and J. Zhang, "Load balance aware genetic algorithm for task scheduling in cloud computing," in proceeding of *Asia-Pacific Conference on Simulated Evolution and Learning* , Springer, Cham ,pp. 644-655. 2014.
[45] H. Cui, X. Liu, T. Yu, H. Zhang, Y. Fang and Z. Xia," Cloud service scheduling algorithm research and optimization," *Security and Communication Networks*, vol. 2017, no. 2503153, 2017.
[46] A. M. Manasrah and H. Ba Ali," Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1934784 ,2018.
[47] A. M. Senthil Kumar and M. Venkatesan, "Task scheduling in a cloud computing environment using HGPSO algorithm", *Cluster Computing*, *22*(1), pp.2179-2185**,** 2019.

[48] A. Mubeen, M. Ibrahim, N. Bibi, M. Baz, H. Hamam and O. Cheikhrouhou, " Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing," *Processes*, vol. *9*, no.9, pp.1514, 2021.

[49] T. K. Dao, T. S. Pan and J. S. Pan," Parallel bat algorithm for optimizing makespan in job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. *29*, no.2, pp.451-462, 2018.

[50] S. Raghavan, P. Sarwesh, C. Marimuthu and K. Chandrasekaran, "Bat algorithm for scheduling workflow applications in cloud," *in proceeding of International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)*, IEEE, pp. 139-144, 2015.

[51] N. Kaur and S. Singh," A budget-constrained time and reliability optimization bat algorithm for scheduling workflow applications in clouds," *Procedia Computer Science*, vol. 98, pp.199-204, 2016.[Online]. https://doi.org/10.1016/j.procs.2016.09.032.

[52] S. Kalaiselvi and K. Selvi ," Hybrid cloud resource provisioning (HCRP) algorithm for optimal resource allocation using MKFCM and bat algorithm," *Wireless Personal Communications*, *vol. 111*, no. 2, pp.1171-1185, 2020.

[53] A. K. Jayswal, "Efficient Task Allocation for Cloud Using Bat Algorithm," *in proceeding of Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, pp. 186-190, 2020.

[54] Z. Movahedi and B. Defude, "An efficient population-based multi-objective task scheduling approach in fog computing systems," *Journal of Cloud Computing*, vol.10, no. 1, pp.1-31, 2021.

[55] K. Sreenu and M. Sreelatha," W-Scheduler: whale optimization for task scheduling in cloud computing", *Cluster Computing*, *vol. 22*, no. 1, pp.1087-1098, 2019.

[56] F. Hemasian-Etefagh and F. Safi-Esfahani, "Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing," *The Journal of Supercomputing*, vol. 75, no. 10, pp.6386-6450, 2019.

[57] G. Narendrababu Reddy and S. Phani Kumar," Regressive whale optimization for workflow scheduling in cloud computing," *International Journal of Computational Intelligence and Applications*, vol. 18, no. 04, pp.1950024, 2019.

[58] M.S. Sanaj and P. J. Prathap, "An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment," *Materials Today: Proceedings*, vol. 37, pp.3199-3208, 2021. [Online]. https://doi.org/10.1016/j.matpr.2020.09.064.

[59] L. Ni, X. Sun, X. Li and J. Zhang, "GCWOAS2: Multi-objective task scheduling strategy based on Gaussian cloud-whale optimization in cloud computing," *Computational Intelligence and Neuroscience*, vol. 2021, no. 5546758, 2021.

[60] J. Ababneh, "A Hybrid Approach Based on Grey Wolf and Whale Optimization Algorithms for Solving Cloud Task Scheduling Problem," *Mathematical Problems in Engineering*, vol. 2021, no. 3517145, 2021.

[61] L. Jia, K. Li and X. Shi," Cloud Computing Task Scheduling Model Based on Improved Whale Optimization Algorithm", *Wireless Communications and Mobile Computing*, Vol. 2021, no. 4888154, 2021.

[62] N.M. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, and Y. Yusoff," Recent studies on optimization method of Grey Wolf Optimizer (GWO): a review (2014–2017)," *Artificial Intelligence Review*, vol. 52, no. 4, pp.2651-2683, 2019.

[63] A. Khalili, S. M. Babamir," Optimal scheduling workflows in cloud computing environment using Pareto-based Grey Wolf Optimizer," *Concurrency and Computation: Practice and Experience*, vol. 29, no.11, pp.e4044, 2017.

[64] B. V. Natesha, N.K. Sharma, S. Domanal and R.M.R. Guddeti," GWOTS: Grey wolf optimization based task scheduling at the green cloud data center," in proceeding of*14th International Conference on Semantics, Knowledge and Grids (SKG)*, IEEE, pp. 181-187, 2018.

[65] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba and M. Zivkovic, "Task scheduling in cloud computing environment by grey wolf optimizer," in proceeding of *27th Telecommunications Forum (TELFOR)* , IEEE, pp. 1-4, 2019.

[66] G. Natesan and A. Chokkalingam," Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm," *ICT Express*, *vol. 5*, no.2, pp.110-114, 2019.

[67] A. Alzaqebah, R. Al-Sayyed and R. Masadeh," Task scheduling based on modified grey wolf optimizer in cloud computing environment," in proceeding of 2nd *International Conference on new Trends in Computing Sciences (ICTCS)*, IEEE, pp. 1-6, 2019.
[68] N. Bansal and A. K. Singh," Grey wolf optimized task scheduling algorithm in cloud computing," *Frontiers in Intelligent Computing: Theory and Applications*, pp. 137-145, Springer, Singapore, 2020.
[69] D. Alsadie, "TSMGWO: optimizing task schedule using multi-objectives grey Wolf optimizer for cloud data centers," IEEE Access, vol. 9, pp.37707-37725, 2021. [Online]. 10.1109/ACCESS.2021.3063723.
[70] A. Mohammadzadeh, M. Masdari, F. S. Gharehchopogh, and A. Jafarian, "Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing," *Evolutionary Intelligence*, *vol. 14*, no.4, pp.1997-2025, 2021.