Journal of Humanities and Social Sciences Studies
ISSN: 2663-7197
DOI: 10.32996/jhsss
Journal Homepage: www.al-kindipublisher.com/index.php/jhsss

JHSSS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Cloud Computing Task Scheduling using Genetic Algorithm: A Survey

Esraa Masadeh[1], Raja Masadeh[2]✉ Omar Almomani[3], Kholoud Alshqurat[4] and Shaymaa Masadeh[5]
[1]*Engineering and Artificial Intelligence Department, Al-Salt Technical College, Al-Balqa Applied University, Al-Salt, Jordan*
[2]*Computer Science Department, the World Islamic Sciences and Education University, Amman, Jordan*
[3]*Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan*
[45]*Academic Services Department, The World Islamic Sciences and Education University, Jordan*
Corresponding Author: Raja Masadeh, **E-mail**: raja.masadeh@wise.edu.jo

| ABSTRACT
Cloud computing has become a hot research topic due to the robust development and migration of many services to this cloud environment. The main problem that appears is regarded with both management's efficiency and the large amount of resources utilization. These resources are managed by data centers as well as distributed to internet users dynamically depending on their availability, request and quality parameters that are in request to be usable. Thus, task scheduling is a major concern that can affect system performance. This study presents a survey of employing genetic algorithm for task scheduling in cloud environments. This survey provides a comprehensive overview of task scheduling approaches in cloud environments, with a particular focus on the application of genetic algorithms. It discusses fundamental cloud computing concepts, scheduling criteria, and classification of scheduling methods. Furthermore, it analyses a wide range of GA-based scheduling algorithms, comparing their performance, task characteristics, and simulation tools.

| KEYWORDS

Task scheduling, Cloud computing, Genetic Algorithm, Optimizations

| ARTICLE INFORMATION

ACCEPTED: 01 June 2025 **PUBLISHED:** 29 June 2025 **DOI:** 10.32996/jhsss.2025.7.6.8

1. Introduction

Nowadays, scheduling has become a daily life technique in different life approaches. Accordingly, scheduler aims at finding a suitable strategy that distributes single or multi-objective tasks to limited resources [1]. Various applications use scheduling technique such as power system control, manufacturing of printed circuit boards and on the World Wide Web (WWW) scheduling of multimedia data objects [1].

From the 1980s till now, scheduling has been used in distributed computing systems, to assign tasks of internet users to limited resources. Different changes have been made to these systems. One of these changes was the innovation of cluster systems. A cluster system is merging multiple computers to work together as one system [2]. However, cluster systems were suffering from local resources usage, to solve the problem, grid technique appeared in order to integrate all heterogeneous resources that are available through an expanded geographically institutions [3]. A novel technique was invented that merges the grid and cluster systems which were called cloud computing systems.

According to the apparently unlimited cloud resources enabled by the latest computing architectures, regrettably, there are no scheduling algorithms exist that can efficiently optimize the distribution of these resources in polynomial time. This is due to most scheduling methods are NP-complete or NP-hard problems [4]. An example is introduced by [5] to inform how much challenging to find the optimal solution for a large-scale problem. For this reason, studies focused on searching a suitable algorithm to solve cloud scheduling problem. Mainly, two types of algorithms are used to solve problems using scheduling: heuristic algorithms and metaheuristic algorithms.

The current computer systems use two scheduling methods: deterministic algorithm (DA) and exhaustive algorithm. Practically, the researchers find that the Das [6, 7] can solve scheduling problems faster, that makes the DAs better than traditional exhaustive algorithms. However, two main drawbacks of DAs existed. DAs were not designed to meet all the data distributions. Also, they were unsuitable for large-scale scheduling problems.

Metaheuristic algorithms are different from DAs and the exhaustive algorithm. They use iterative strategies in order to reach solutions through specific times. Research has demonstrated that metaheuristic scheduling algorithms often outperform traditional scheduling strategies analyzed with respect to efficiency and effectiveness. However, the environment of cloud computing is not the main center of interest for metaheuristic algorithms. Although various scheduling techniques have proven successful in other computing contexts, their adaptation to cloud environments remains limited.

2. Background

2.1 Cloud Computing

Due to no explicit definition of cloud, several ways can be used to define it and by considered several means. Cloud computing (CC) is considered as an internet-connected method of supercomputing. Moreover, it is a kind of shared infrastructure that sets the massive pools of system together utilizing several means. This leads to giving clients a diversity of storage, resources and networking in cloud environment through internet; clients lay much of information and access much of power with the assistance of their own computer devices.

Based on Buyya [2], the definition of cloud is as “cloud is a form of parallel and distributed computing that depends on connected network of virtual machines. These systems are dynamically assigned and offered to clients as a unified computing environment, according to service-level agreements (SLAs) established through negotiation between clients and cloud providers”.

CC is a computing model that is spread on a large-scale, that relies on the cloud economic size of operator such abstract, virtualized and dynamic. CC mainly manages storage, computing power, multiple services and platforms that the external users upload through internet demand [8-13]. Also, CC is a fast computation model that aims at moving the responsibility of managing hardware, software, networks and data resources from cloud users to cloud service providers.

Clouds in CC can provide a massive number of resources which involves platforms for firewalls, data centers, computation, storages and software as services. In addition, it presents many ways of treating these resources in which cloud clients can access them without encountering any problems that is related to performance.

As shown in Fig.1, there are three kinds of services supplied by the CC. The first one is Software as a Service (SaaS), a model that delivers software applications to end users. In other words, there is no needs to install any software by users on their machines and can employ the software immediately from the CC. Second one is Platform as a Service (PaaS) which offers users a platform. Thus, to implement their programs and applications on the presented platform. Last one is Infrastructure as a Service (IaaS), which gives the infrastructure to the clients for different reasons such as computation resources and system's storage [14].

Deployment models are classified into Public, Private, Community, and Hybrid Clouds computing as illustrated in Fig.2 [4]. Clients can subscribe to these types of clouds in CC depending on his/her needs. As home use or business use, client will mostly use public cloud services. Any client can access the public cloud computing with an internet connection. While private cloud computing is created for a particular group and access to this group is limited. A community cloud computing is participated between at least two organizations which have the same CC requirements. A combination between at least two of these previous types of clouds is called a hybrid CC [4].

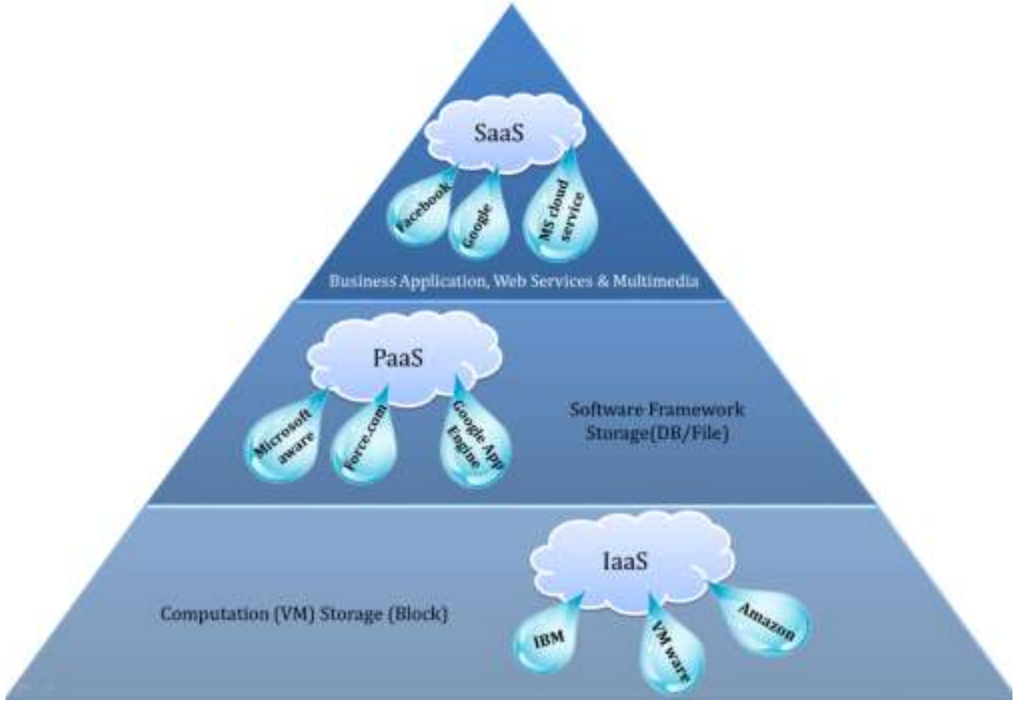

Fig.1: Service Model

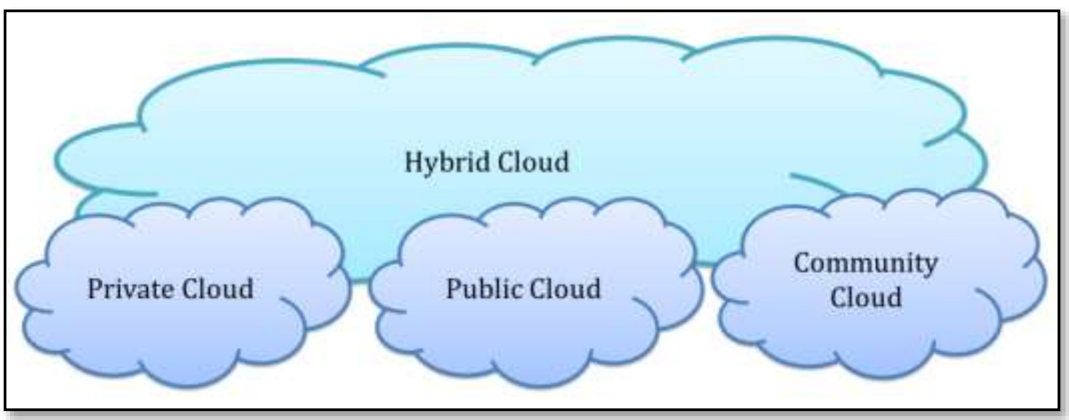

Fig.2: Deployment models

2.2 Cloud Computing Task Scheduling

In CC environments, task scheduling involves selecting the most appropriate and available resources to allocate VM to tasks or to execute tasks in a way that reduces completion time. By allocating priority to each job, the scheduling process creates a list of jobs, with each job allocated a priority relies on a variety of factors. As a result, tasks are selected and assigned to suitable VM that meet a defined objective function [15].

Virtualization techniques are used in CC environment for mapping CC resources to VM layer and execute the task of user. Therefore, cloud task scheduling is achieved at the application layer and VM layer as shown in Fig.3. Scheduling is considered as the mapping of resources and tasks according to some particular parameters with the aim of reaching the desired outcome. Paradigm of CC simplifies the mapping of tasks into VMs. The desired resources together are formed to be VMs. The search operation of the required resource package is similar to the search process of different VMs.

CC classifies scheduling into multiple categories as presented in Fig.4. The first category depends on Task. This type is divided into dynamic and static scheduling. Static scheduling is usually applied on periodic tasks, because when tasks arrive concurrently with the processor, the scheduling decision is taken before tasks are submitted to the resources, they are submitted to the available resources. The processing time of a task is determined when the task completes its work. However, in dynamic scheduling no information such as tasks numbers, machine location and resource allocation are known before task arrival. Also, task arrival time is not known before task submission.

Furthermore, dynamic scheduling is divided into batch mode and online mode [16]. In batch mode, when tasks arrive, they get grouped together and after specific time they get scheduled. However, in online mode tasks get scheduled directly after arrival.

The online mode depends on multiple metrics used in cloud. This type is divided into batch system, interactive system and real-time system. In the case of a batch system, we are interested in measuring turnaround time and throughput. While in interactive system, response time and fairness are calculated. However, in a real-time system, measuring deadline is very important.

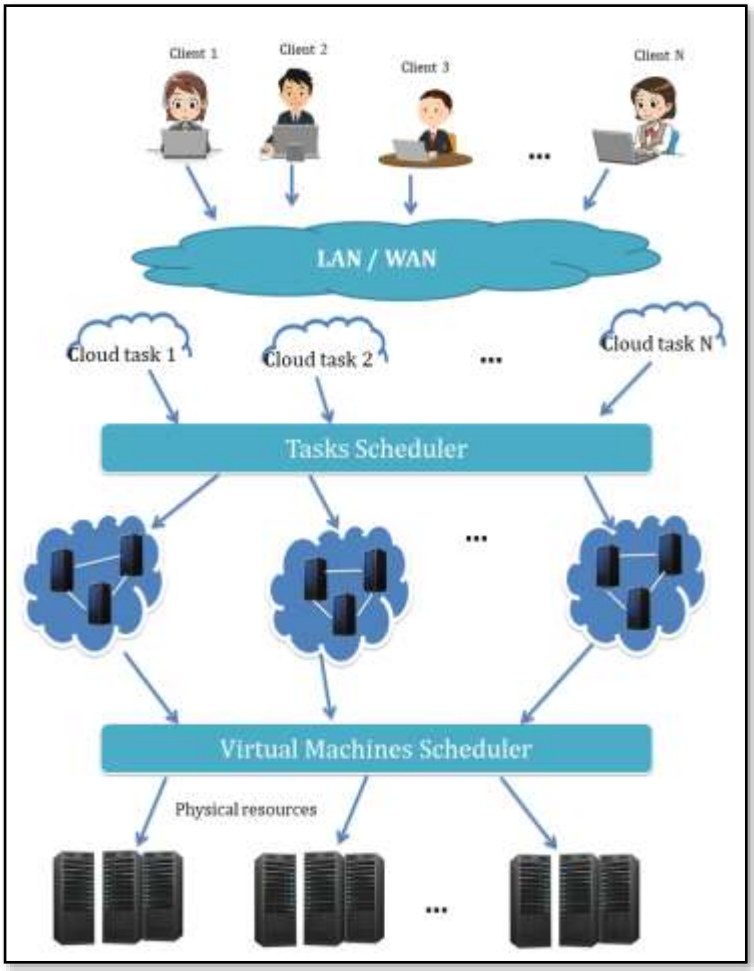

Fig.3: General view of cloud task scheduling

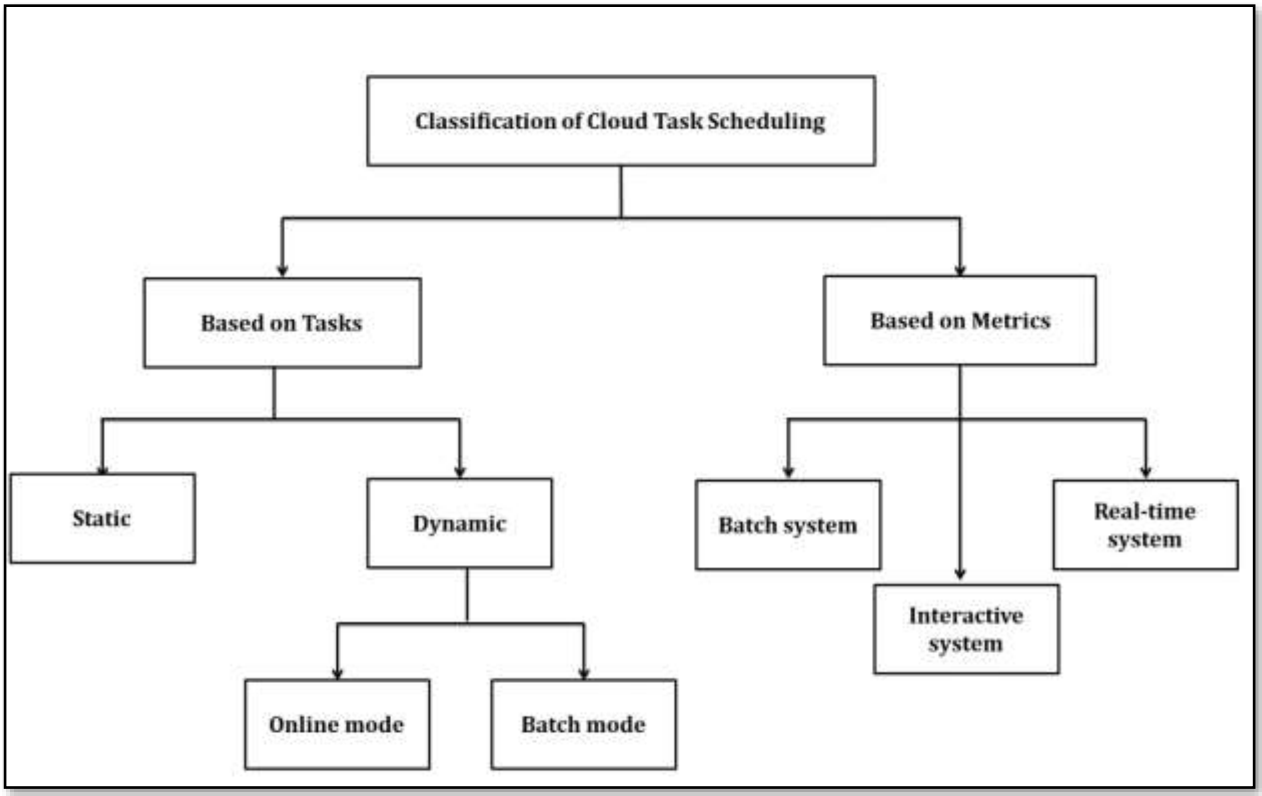

Fig.4: Classification of cloud task scheduling

Various mechanisms are utilized to address one of the primary challenges in cloud computing: task scheduling. These mechanisms are classified into three classes; traditional algorithms, heuristic and meta-heuristic algorithms. Traditional algorithms exist job scheduling techniques that are used in distributed computing system. Some of these mechanisms are applied in CC system with appropriate verification. The main advantages of task scheduling are to achieve the best throughput of the system as well as a high computing performance. The main examples of traditional algorithms are First-Come-First Serve (FCFS), Shortest Job First (SJF), Preemptive SJF, Round Robin (RR) and Most Fit Task Scheduling (MFTS). Many studies employed these types of mechanisms with modification for mapping tasks to available resources such as [17]. In CC environment, to optimize results, heuristic algorithms-based scheduling can be done. These algorithms are Min- Min algorithm, Max-Min algorithm, Minimum Execution Time (MET), Minimum Completion Time (MCT), Xsuffrage [18], Backfilling algorithms [18] and RASA which is a combination of min-min and max-min algorithms. Many researchers deploy these types of algorithms with some modification such as in [19]. Meta-heuristic algorithms are categorized into evolutionary and swarm-intelligence algorithms. Evolutionary optimization algorithms are considered as generic population-based metaheuristic optimization algorithms. These algorithms are inspired by biological evolution. Genetic Algorithm (GA), Chemical Reaction Optimization (CRO), Lion algorithm (LA) and Harmony Search (HS) are evolutionary optimization algorithms and have been discussed in context with meta-heuristic task scheduling section. Swarm intelligence optimization algorithms are inspired from biological populations and social behavior of the collective intelligence of swarms such as Whale Optimization Algorithm (WOA), Grey Wolf Optimization (GWO), Bat Algorithm (BA), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Firefly Algorithm (FA), Cat Swarm Optimization (CSO), Cuckoo Search (CS), Particle Swarm Optimization (PSO), Social Learning Optimization Algorithm (SLO), Glowworm Swarm Optimization (GSO) and Sea Lion Optimization (SLnO).

3. CC Task Scheduling Methods

Cloud computing contains various resources. More precisely, these resources are different from each other's via some means as well as different cost of execution tasks using CC resources. Thus, task mapping in CC is different from traditional task scheduling techniques. In CC, it requires better awareness to be paid due to services.

Tasks- resources mapping is essential in developing systems in term of reliability and flexibility in cloud computing. The major purpose behind mapping tasks to resources according to the specific time bound includes discovering a best and complete series in which different tasks can be implemented effectively and efficiently.

Task scheduling is recognized as an NP-complete problem. Task scheduling is illustrated by employing heuristic and meta-heuristic algorithms to find optimal or nearly optimal solutions efficiently because traditional methods often fail to produce optimal solutions within a reasonable time frame in many cases. However, the problem that is faced with the heuristic algorithms solutions usually is being stocked in local minima of sarsolution set. Whilst available meta-heurist methods proved more efficient approaches to avert this case as was stated in [20]. Glover and Laguna stated "meta-heuristic" concept, from amending of heuristic algorithms in order to find out best solutions in the context of the optimal local search. In this study, we have presented and discussed almost all meta-heuristic algorithms that are employed to solve the task scheduling problem in CC.

3.1 Criteria in CC Task Scheduling

The scheduling procedure assigns the tasks to available and suitable resources based on specific scheduling standards. The parameters of scheduling makespan, cost, budget, deadline and reliability have an influence on the success of both dependent and independent task scheduling problem. These parameters are categorized as customer service and provider's service as shown in Fig.5. Besides, there is another classification of scheduling presented which depends on the number of objectives: single objective, bi-objective and multi-objective.

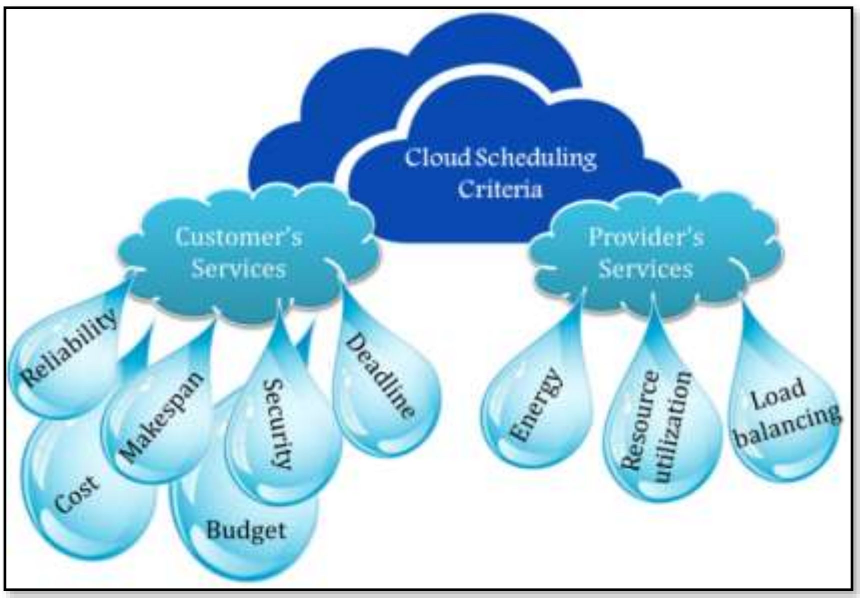

Fig. 5: Cloud computing scheduling criteria

The customers are looking for services with best affordable budget, less cost, within a deadline, very secure, reduced makespan, and more reliability.

The budget is regarded as a constraint which is determined by customer in order to exploit the requested service from service provider in the CC environment. By using this constraint, the decisions of scheduling are made to reduce the makespan of the given tasks as well as it has to finish within the budget constraint [21]. Many VMs are used to implement all given tasks. Thus, the total execution cost is the aggregate price of all used VMs.

The cost of an application relies on the computation cost, data transfer cost and storage. According to [22] when an available resource is allocated to a given task, then data is transmitted among tasks and transfer cost is computed using equation (1). The transfer cost becomes zero in case the parent task and child task are scheduled on the same VM.

Transfer Cost = Size of output data given by a task / Average bandwidth between VMs … (1)

The deadline is presented for applications for which time it is critical to finish its implementation within a specific period of time to submit outcomes before the cutoff time. Moreover, the deadline requires to take financial cost into consideration in case it schedules all given tasks. There is a great need for robust scheduling within a deadline in applications for which time is important and critical. Many studies considered the deadline with other parameters such as in [23, 24]. Task scheduler defines the configuration of VMs to schedule the task in order to reduce the costs as well as meet the deadline parameter which is determined by customer in hybrid CC environment [25].

The makespan indicates to the total time desired to implement all given tasks by taken into account the completion time of the tasks, their implementation and the time at which they are presented [26]. In other words, it is known as the period from submitting the tasks by customers to the time they finish and present outcomes. Many studies focused on minimizing and optimizing makespan such as in [27, 28]. Reducing the overall execution time leads to minimizing the execution cost during allocating tasks across the available resources. Thus, it is regarded as one of the most significant goals of scheduling objectives which minimizes the length of schedule in order to meet customer requirements.

Security is regarded as a vital priority in CC because resources in CC are distributed and heterogenous. Privacy and security exist in CC unlike traditional systems because of virtualized and multi-tenant nature. During the assigning of suitable and available resources to jobs, the level of client satisfaction is considered [29]. Security challenges, including data leakage and hypervisor vulnerabilities, are regarded as resource virtualization. Therefore, there is a pressing need to implement secure and efficient virtualization techniques to address these security concerns [30].

Failing of workflow implementation may be potential due to many causes such as unavailable resources and failing of resources. Therefore, a process of scheduling should be responsible for resource failure as well as to ensure dependable implementations despite failures and synchronization. More precisely, reliability measures the probability of successfully performing and finishing the execution of all assigned tasks. Reliability ensures that all resources work efficiently while the application is being implemented [31]. This objective participates to reduce the occurrence of failures to finish the allocated workflow. The calculation of the failure rate is required in order to reduce both reliability and rates of failure [32, 33]. Workflow application contains three levels of failure [34]: task level, VM level and workflow level. There are many alternative solutions to tackle failure cases such as redundancy, alternate resources, salvage workflow, restart and retry.

From the side of the providers of services less energy consumption, better resources utilization and load balancing are very important for tasks scheduling in CC. Both resource utilization and CPU utilization have direct effect on energy consumption by a given task. Energy consumption will be high if CPUs are not properly used because idle energy is not efficiently utilized. However, at times it grants high consumed energy because the request of resources is heavy [35]. This may lead to reduce the performance [36]. Decisions of tasks scheduling are considered as a significant to discover the effective series of tasks implementation in order to minimize power consumption by the allocated resources [37]. There are many CC tasks scheduling methods designed to help minimize power consumption. High resource utilization is useful to service providers. In order to bring extreme profit by hiring limited resources to customer in which these resources are fully used.

In CC environments, the VMs are major processing elements. In task scheduling, there is a specific situation that appears when many tasks are allocated to VMs in order to run these tasks at the same time. Thus, this situation leads to imbalanced loads through VMs. To prevent excessive load on specific resources, the scheduler should efficiently distribute tasks among the available and suitable resources [38]. Balance the load over resources; developing resources leads to exploitation. Consequently, the whole performance of the scheduling process. Load balancing techniques are generally classified into static and dynamic approaches. Prior knowledge of the jobs is required for static load balancing [38]. On the other hand, dynamic load balancing does not depend on such information. When maximizing resource usage rather than limiting makespan is the goal, dynamic load balancing can be useful [39]. Many researchers have discussed load balancing techniques based on meta-heuristic algorithms. In addition, many scheduling objectives are considered along with load balancing.

4. Task Scheduling Approaches using GA

Meta-heuristic methods are designed to obtain optimal or near-optimal solutions within a reasonable time frame, offering improved efficiency compared to traditional approaches [40]. This type of algorithms is classified into two categories [40]: single solution-based metaheuristics and population-based metaheuristics. There are two main different characteristics that can be used to distinguish between them. First, a wide range of solutions is explored during each iteration. Second, different iteration is used to search the information of propagated iteration [40]. Many meta-heuristic algorithms presented solutions in order to solve task scheduling problem. This study presents genetic algorithm that introduced solutions for task scheduling in CC environment as shown in Table 1.

Genetic algorithm (GA) is one of the most significant population-based metaheuristic algorithms due to its performance and easy to apply and adopt many problem domains [41, 42]. According to the report of Google Scholar as mentioned in [43], it was cited 14712 times and in [43], it was cited 24941 times. Both are ranked as one of the top 100 most cited researches in Google Scholar; 31 and 10 respectively.

Genetic Algorithm (GA) operates through a set of key operators: initialization, selection, reproduction, crossover, and mutation, which collectively simulate the evolutionary process found in nature. Solutions are denoted as chromosomes (also called individuals), which are produced randomly in the initialization phase. The fitness function is utilized to evaluate which solutions are more appropriate, unlike the other metaheuristic algorithms that use the fitness function in order to determine which solutions are better than other solutions. In addition, using the fitness function to separate the solutions according to other criteria can be helpful. The selection operator plays a crucial role in guiding the search direction in subsequent iterations. The crossover and mutation operators are employed in order to pass the solutions, while mutation helps to avoid convergence to local optima by introducing diversity.

The most common encoding representations of the scheduling solutions of GA are binary, n-task sequence, tree, random key, and n × m matrix. The binary encoding was employed in [44] in order to connect the generations and makespan with individuals with length m × mi where m indicates to the number of tasks and m represents the number of operations that necessary by task i. Each individual represents the arrangement of all processes of all tasks. For instance, assuming that there are two tasks, each of them requires two operations. Thus, if the solution is (1, 1, 2, 2), then it represents the arrangement with which the processes are executed. More precisely for performing the operations of the previous tasks, the operation of task 1 is executed first, then the second operation of task 1, then the first operation of task 2 and so on. The n-task sequence encoding is employed in [45] to encode the GA solutions. A sequence = {S1, S2, S3, … Sn}, where Si indicates to task i is designated to available resource; which is used to represent a solution. For example, S2=4 indicates that task 2 is designated to the fourth resource. However, in order to represent the mapping relationship among VMs. The tree structure is used in [46] and [47] to encode the GA scheduling solutions. To ensure the validity of individuals, the transition operator needs to be carefully reconsidered. In order to maintain the feasibility (task prioritizing) of individuals, a motivating representation was utilized in [48] that utilized the random key. However, this type of representation demands more encoding and decoding techniques for the other GA operators. The last type of representation is n × m matrix which is employed to represent the model of task that contains the relationships among tasks and resources. The model of expected execution time (EET) is used in [49] which employs the matrix representation to register the EET of task i on resource j.

A direct representation is employed [50, 51] in which a chromosomes matrix is denoted as mapping tasks into virtual machines. The chosen strategy of GA can be considered as the estimation and definition operators of metaheuristic algorithms [52]. More precisely, various assumptions may lead to various respects in the fitness functions' design. For instance, the aims of task scheduling are makespan and the consumed power [52]. The cost of communication, computation and profit are taken into consideration on order to design the fitness functions. Whereas some studies employ constant length binary string to encode solutions such as in [53]. Moreover, the proposed algorithm that is based on GA reduces makespan with optimizes the load balancing.

A new task scheduling approach on heterogeneous computing systems using a multiple priority queues genetic algorithm (MPQGA) has been designed [54]. The main idea of the suggested algorithm is to take the features of both heuristic-based and evolutionary-based algorithms with avoiding their disadvantages. In other words, the suggested algorithm integrated GA technique in order to designate a priority to each subtask and employed a heuristic- based earliest finish time (EFT) technique in order to search for a solution for mapping a task into a processor. Besides, crossover, mutation and fitness function are put forward for directed acyclic graph (DAG) scenario. Based on the experimental results, it proved that the suggested algorithm (MPQGA) outperforms both random search method and non-evolutionary heuristics in terms of large size of problems.

Some researchers presented an improved GA by merging GA with traditional scheduling algorithms. A new task scheduling algorithm based on merging GA with Min-Min and Max-Min scheduling methods is suggested [55]. The final results of the improved GA prove that its performance is better than the standard GA. A modification of GA is designed [56] by creating an initial population using Max-Min and Min-Min algorithms. Moreover, resource utilization and makespan are taken into accounts [56]. The simulation results proved that the modification GA reduces the makespan as well as uses the available resources efficiently comparing with standard GA. In [57], a task scheduling approach based on GA is designed in which an initial population is created using Max-Min algorithm to obtain more optimum outcomes in terms of makespan. Whereas the experimental results proved that the suggested approach displays better scheduling of tasks compared with Improved Max Min, GA with LCFP (Longest Cloudlet Fastest Scheduler) and IGA (Improved Genetic Algorithm). In [58], a solution based on GA is proposed with considering to makespan. A modification of GA is proposed in [59] by creating an initial population of chromosomes with output schedules of algorithms Smallest Cloudlet to Fastest Processor (SCFP), Longest Cloudlet to Fastest Processor (LCFP) and 8 Random Schedules. The experimental outcomes proved that modification of GA presents a perfect performance in case of heavy loads. Similar parameters of [60, 61] are used in [59] but different strategies are employed based on GA. In [62], they proposed a new task scheduling algorithm based on double-fitness adaptive algorithm-job spanning time and load balancing genetic algorithm (JLGA) in terms of load balancing and minimizing makespan. In order to initialize the population, the researchers adopted greedy algorithm, which makes variance in characterizing the load of dense nodes, multi-fitness function weights. Based on the experimental results, it proved that the proposed algorithm balances the whole system load effectively and took little in overall task and average task consumption compared with adaptive GA. Whereas in [63], they designed a modification of the standard GA with assistance of fuzzy theory in order to minimize the number of iterations of making the population. Experimental results proved that the suggested algorithm presented an improvement of system performance in terms of both makespan and cost. A hybrid heuristic algorithm based on GA [64] is proposed for dependent tasks scheduling in CC environment. In such a way, it optimizes both load balancing and speedup ratio, besides minimizes makespan.

Bi-objective dynamic workflow scheduling based on GA algorithm called (DWSGA) is designed [65]. The experimental results demonstrated that DWSGA algorithm achieves better performance compared to state-of-art algorithms in terms of both makespan and speedup ratio. Whilst other study introduced a novel dependent task scheduling technique based on GA [66].

Many researchers presented multi-objective (MO) task scheduling model in order to solve task-resource mapping problem taken into accounts the cost of CPU, memory, bandwidth, power consumption, makespan and so on. The first study [67] that presented a new GA task scheduling which considers more than two parameters, which are quality of service (QoS), makespan, cost and deadline. In [68], multi-objective GA was presented taken into consideration the consumed energy of data centers, CO2 emissions and created profits in order to minimize the interest rate of consumed energy. The well-known multi-objective GA (NSGA II) was employed and took into accounts the cost of memory, CPU and bandwidth at the same time [69]. A solving technique based on multi-objective GA (MO-GA) is proposed in [70] as well as focused on the methods of Pareto solutions, encoding rules, selection and crossover operators. Moreover, the minimization of consumed energy and the maximization of the service's profit are taken into considerations.

One of the main problems of task scheduling is assigning the available and correct resource to the arrived task. Dynamic scheduling has uncleared arrival task at run time and resources allocation is complicated when multiple tasks are reached simultaneously. Thus, GA is employed [71] to avert this issue and gain global optimization. Resource utilization and time utilization are taken into accounts during designing the dynamic task scheduling. Using parallel processing, the execution time is minimized. [72] introduced the Balancer Genetic Algorithm (BGA), a novel task scheduling optimization technique in CC. BGA goals to increase the efficiency of task scheduling by balancing the workload among cloud resources and lowering the total makespan. More precisely, BGA is based on genetic algorithm principles, with an emphasis on resource balance during job assignment. It improves on the classic GA by including a "balancing" technique that distributes the workload evenly among resources, assuring optimal resource use and lowering job execution time. A job scheduling technique based on GA in CC environment employing the MapReduce framework is suggested by [78], which aimed to optimize utilized resources and makespan by allocating jobs to VMs. More precisely, the usage of the MapReduce framework improves scalability and parallel processing capabilities.

In order to optimize task-resource allocation, the study [73] presented a GA-based job scheduling method for CC settings. The study concentrated on using GA's evolutionary capabilities to optimize important performance indicators, taking into consideration makespan and resource usage. a customized chromosome structure and fitness function are designed in [73] to accommodate scheduling objectives. The study [74] introduced a Genetic-Based Multi-objective Task Scheduling Algorithm (G-MOTSA) for CC environments, aiming to improve significant performance factors, including makespan, throughput, energy consumption, and load balancing. The technique presented a new chromosome representation in the form of a $3\times n$ matrix, which simultaneously encodes server allocation, VM assignment, and task scheduling. The proposed technique showed its efficacy in improving resource utilization, energy efficiency, and system performance in CC environments.

The Balancer Genetic Algorithm (BGA) was suggested by [75] as a new job scheduling algorithm, which is designed to maximize load balancing and resource utilization in CC environments. BGA improves scheduling efficiency by merging balance-aware techniques into the traditional GA framework. The results showed that the proposed algorithm outperformed traditional techniques in terms of makespan and resource allocation. While A multi-objective hybrid GA was introduced by [76] as a solution to the task scheduling issue in CC environment. The suggested algorithm aimed to optimize makespan and cost, by merging the exploration capabilities of GA with additional heuristics. A hybrid job scheduling mechanism for CC environment is introduced by [77], which merged GA and PSO techniques. The hybrid algorithm aimed to optimize the makespan, resource utilization and load balancing. Based on the experimental outcomes, the hybrid algorithm showed better performance than standalone PSO and GA. In order to enhance scheduling efficiency in a CC environment, a hybrid task scheduling technique is introduced by [78], which combines GA and PSO algorithms. The main objectives of the suggested mechanism are to reduce makespan and enhance resource utilization.

Table 1: Summary of CC task scheduling based on GA

Reference	Year	Performance Criteria	Task Nature	Used Tools
[50]	2009	• Makespan	Dependent	CloudSim
[44]	2010	• Makespan	Dependent	MapReduce
[51]	2010	• Makespan Speedup ratio	Independent	Hadoop MapReduce
[52]	2011	• Makespan • Energy consumption	Independent	Not mentioned
[46]	2011	• Makespan • Load balancing • Budget • Deadline	Dependent	Java OpenNebula Cloud API 3.0
[45]	2011	• Makespan • Load balancing • Quality of Service	Independent	CloudSim
[68]	2011	• Energy consumption. • CO2 emissions • Profit	Independent applications	realistic workloads traces from Feitelson's Parallel Workload Archive (PWA)
[69]	2011	• Makespan • Budget • Deadline	Dependent	CloudSim
[36]	2012	• Energy consumption • Resource utilization	Independent	Hadoop MapReduce
[58]	2012	Makespan	Independent	Hadoop MapReduce
[56]	2012	• Makespan • Resource utilization	Independent	CloudSim
[59]	2012	• Makespan • Execution cost	Dependent	Java based simulator
[55]	2012	• Makespan • Resource utilization	Independent	CloudSim
[53]	2013	Load balancing	Dependent	CloudAnalyst
[67]	2013	• Makespan • Energy consumption • Quality of Service • Reliability	Dependent	Not mentioned
[70]	2013	• Power consumption • Profit	Independent	CloudSim
[71]	2013	• Resource utilization	Independent	Open Nebula
[61]	2013	• Makespan • Budget Deadline	Dependent	Java-based simulator

[63]	2014	• Makespan	Dependent	Cloud simulation environment
[57]	2014	• Makespan	Independent	CloudSim
[62]	2014	• Makespan • Load balancing	Independent	Matlab
[64]	2014	• Makespan • Load balancing • Speedup ratio	Dependent	Not mentioned
[65]	2014	• Makespan • Load balancing	Dependent	Not mentioned
[54]	2014	• Makespan • Cost • Budget • Deadline	Dependent	CloudSim
[66]	2018	• Makespan	Dependent	WorkflowSim
[73]	2019	• Makespan • Resource utilization	Dependent	Workflowsim
[72]	2021	• Makespan • Load balancing • Quality of Service	Dependent	CloudSim
[74]	2021	• Makespan • Resource Utilization • Energy Consumption • Imbalance Degree. • Throughput	Independent	CloudSim
[75]	2021	• Load balancing • Resource utilization	Independent	CloudSim
[76]	2021	• Makespan • Cost	Independent	custom-developed simulator implemented in Java
[77]	2021	• Makespan • Resource utilization • Load balancing	Independent	CloudSim
[78]	2022	• Makespan • Resources utilization	Independent	Hadoop platform

5. Conclusion

Cloud computing has become an essential infrastructure for delivering scalable and cost-effective computational resources. However, task scheduling remains a significant challenge due to the complexity of dynamic environments and diverse user requirements. Genetic algorithms, as a class of metaheuristic optimization techniques, have been widely explored to address this challenge effectively. This survey reviewed a broad spectrum of GA-based approaches for task scheduling in cloud environments, covering both single and multi-objective models, hybrid methods, and performance enhancing modifications. The analysis reveals that GA-based solutions are capable of achieving reduced makespan, improved load balancing, and better resource utilization.

Funding: This research received no external funding

Conflicts of Interest: The author declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

ORCID

1. https://orcid.org/0009-0003-3839-157X
2. https://orcid.org/0000-0003-4519-7110
3. https://orcid.org/0000-0003-3160-6542
5. https://orcid.org/0000-0002-3237-3675

References

[1] Abualhaj, M., Mahran Al-Zyoud, M (2024). A fine-tuning of decision tree classifier for ransomware detection based on memory data. *International Journal of Data and Network Science*: 733-742.

[2] Abualhaj, Mosleh M., and Sumaya N. Al-Khatib (2024). Using decision tree classifier to detect Trojan Horse based on memory data. TELKOMNIKA (Telecommunication Computing Electronics and Control): 393-400.

[3] Ai, L., Tang, M., & Fidge, C. J. (2010). QoS-oriented sesource allocation and scheduling of multiple composite web services in a hybrid cloud using a random-key genetic algorithm.

[4] Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European journal of operational research, 187*(3), 985-1032.

[5] Almaiah, M and Omar A (2024). Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems. Computers, Materials & Continua

[6] Almomani, O (2020). A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. Symmetry: 1046.

[7] Almomani, O., & Madi, M. (2014). A GA-Based Replica Placement Mechanism for Data Grid. *International Journal of Advanced Computer Science and Applications (IJACSA), 5*(10).

[8] Almomani, O., Adeeb A., Ahmad A., Abdullah A., Mohammed A A., and Qusai S (2025). Enhance URL Defacement Attack Detection Using Particle Swarm Optimization and Machine Learning. *Journal of Computational and Cognitive Engineering* .

[9] Alsaaidah, A and Anusha A (2024). ARP Spoofing Attack Detection Model in IoT Network using Machine Learning: Complexity vs. Accuracy. *Journal of Applied Data Sciences*: 1850-1860.

[10] Alshraideh, M., Mahafzah B and Al-Sharaeh, S. (2011). A multiple-population genetic algorithm for branch coverage test data generation *Software Quality Journal*, 19(3), 489-513.

[11] Alwada, N., & Falah, T. (2018). Cloud Computing and Multi-Agent System: Monitoring and Services. *Journal of Theoretical & Applied Information Technology, 96*(9).

[12] Annette J, R., Banu W, A., & Shriram, S. (2013). A taxonomy and survey of scheduling algorithms in cloud: based on task dependency. *International Journal of Computer Applications, 82*(15), 20-26.

[13] Aryan, Y., & Delavar, A. G. (2014). A bi-objective workflow application scheduling in cloud computing systems. Int J Integr Technol Educ, 3(2), 51-62.

[14] Atiewi, S., Abuhussein, A., & Saleh, M. A. (2018, September). Impact of Virtualization on Cloud Computing Energy Consumption: Empirical Study. In Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control (p. 24). ACM.

[15] Atiewi, S., Yussof, S., & Ezanee, M. (2015). A comparative analysis of task scheduling algorithms of virtual machines in cloud environment. *Journal of Computer Science, 11*(6), 804.

[16] Belgacem, A., Beghdad-Bey, K., & Nacer, H. (2018, April). Task Scheduling in Cloud Computing Environment: A Comprehensive Analysis. In International Conference on Computer Science and its Applications (pp. 14-26). *Springer, Cham.*

[17] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems, 25*(6), 599-616.

[18] Chunlin, L., & Layuan, L. (2006). QoS based resource scheduling by computational economy in computational grid. *Information Processing Letters, 98*(3), 119-126.

[19] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (GA) based load balancing strategy for cloud computing. *Procedia Technology, 10*, 340-347.

[20] Davis, L. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, NY.

[21] Delavar, A. G., & Aryan, Y. (2011). A synthetic heuristic algorithm for independent task scheduling in cloud systems. International Journal of Computer Science Issues (IJCSI), 8(6), 289.

[22] Delavar, A. G., & Aryan, Y. (2014). HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. Cluster computing, 17(1), 129-137.

[23] Emara, F. A., Gad-Elrab, A., Sobhi, A., & Raslan, K. R. (2021). Genetic-Based Multi-objective Task Scheduling Algorithm in Cloud Computing Environment. *International Journal of Intelligent Engineering & Systems, 14*(5).

[24] Foster, I. (2002). The grid: A new infrastructure for 21st century science. Phys. Today, 55(ANL/MCS/JA-42173).

[25] Fu, X., Sun, Y., Wang, H., & Li, H. (2023). Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm. *Cluster Computing, 26*(5), 2479-2488.

[26] Ge, Y., & Wei, G. (2010, October). GA-based task scheduler for the cloud computing systems. In Web Information Systems and Mining (WISM), 2010 International Conference on (Vol. 2, pp. 181-186). IEEE.

[27] Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addion Wesley. Reading.

[28] Gonzalez, N., Miers, C., Redigolo, F., Simplicio, M., Carvalho, T., Näslund, M., & Pourzandi, M. (2012). A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications, 1*(1), 11.
[29] Gu, J., Hu, J., Zhao, T., & Sun, G. (2012). A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of computers, 7*(1), 42-52.
[30] Gulbaz, R., Siddiqui, A. B., Anjum, N., Alotaibi, A. A., Althobaiti, T., & Ramzan, N. (2021). Balancer genetic algorithm—A novel task scheduling optimization approach in cloud computing. Applied Sciences, 11(14), 6244.
[31] Gulbaz, R., Siddiqui, A. B., Anjum, N., Alotaibi, A. A., Althobaiti, T., & Ramzan, N. (2021). Balancer genetic algorithm—A novel task scheduling optimization approach in cloud computing. *Applied Sciences, 11*(14), 6244.
[32] Holland, J. H. (1975). An introductory analysis with applications to biology, control, and artificial intelligence. Adaptation in Natural and Artificial Systems. First Edition, The University of Michigan, USA.
[33] Huang, D. W., & Lin, J. (2010, November). Scaling populations of a genetic algorithm for job shop scheduling problems using MapReduce. In Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on (pp. 780-785). IEEE.
[34] Huang, J. (2014). The Workflow Task Scheduling Algorithm Based on the GA Model in the Cloud Computing Environment. JSW, 9(4), 873-880.
[35] Hwang, E., & Kim, K. H. (2012, September). Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud. In Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 130-138). IEEE Computer Society.
[36] Jain, A., & Upadhyay, A. (2017). Cloud Scheduling using Meta Heuristic Algorithms. *International Journal of Computer Sciences and Engineering.*
[37] Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., & Abraham, A. (2014). Hybrid job scheduling algorithm for cloud computing environment. In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 (pp. 43-52). Springer, Cham.
[38] Kaleeswaran, A., Ramasamy, V., & Vivekanandan, P. (2013). Dynamic scheduling of data using genetic algorithm in cloud computing. *International Journal of Advances in Engineering & Technology, 5*(2), 327.
[39] Kalra, M., & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egyptian informatics journal, 16*(3), 275-295.
[40] Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS), 4*(10), 74-79.
[41] Kessaci, Y., Melab, N., & Talbi, E. G. (2011, July). A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures. In High Performance Computing and Simulation (HPCS), 2011 International Conference on (pp. 456-462). IEEE.
[42] Khurma, R. A., AL Harahsheh, H. E. B. A., & Sharieh, A. (2018). Task Scheduling Algorithm in Cloud Computing Based on Modified Round Robin Algorithm. *Journal of Theoretical & Applied Information Technology, 96*(17).
[43] Kumar, P., & Verma, A. (2012). Independent task scheduling in cloud computing by improved genetic algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(5).
[44] Kumar, P., & Verma, A. (2012, August). Scheduling using improved genetic algorithm in cloud computing for independent tasks. In Proceedings of the International Conference on Advances in Computing, *Communications and Informatics (pp. 137-142).* ACM.
[45] Liu, J., Luo, X. G., Zhang, X. M., Zhang, F., & Li, B. N. (2013). Job scheduling model for cloud computing based on multi-objective genetic algorithm. *International Journal of Computer Science Issues (IJCSI),* 10(1), 134.
[46] Maqableh, M., Karajeh, H., & Masadeh, R. (2014). Job scheduling for cloud computing using neural networks. Communications and Network, 6(3), 191-200.
[47] Marcon, D. S., Bittencourt, L. F., Dantas, R., Neves, M. C., Madeira, E. R., Fernandes, S., ... & da Fonseca, N. L. (2013, December). Workflow specification and scheduling with security constraints in hybrid clouds. In Cloud Computing and Communications (LatinCloud), 2nd IEEE Latin American Conference on (pp. 29-34). IEEE.
[48] Masa'deh, R., Louzi, M. & Darawsheh, S. (2018). Cloud Computing Adoption in Jordanian Universities. *Journal of Business & Management (COES&RJ-JBM), 3*(4).
[49] Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y. C., Talbi, E. G., Zomaya, A. Y., & Tuyttens, D. (2011). A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing, 71*(11), 1497-1508.
[50] Mocanu, E. M., Florea, M., Andreica, M. I., & Ţăpuş, N. (2012, March). Cloud computing—task scheduling based on genetic algorithms. In Systems Conference (SysCon), 2012 IEEE International (pp. 1-6). IEEE.
[51] Morton, T., & Pentico, D. W. (1993). Heuristic scheduling systems: with applications to production systems and project management (Vol. 3). John Wiley & Sons.
[52] Nagar, R., Gupta, D. K., & Singh, R. M. (2018). Time Effective Workflow Scheduling using Genetic Algorithm in Cloud Computing.
[53] Peng, Z., Pirozmand, P., Motevalli, M., & Esmaeili, A. (2022). Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework. *Mathematical Problems in Engineering, 2022*(1), 4290382.
[54] Pirozmand, P., Hosseinabadi, A. A. R., Farrokhzad, M., Sadeghilalimi, M., Mirkamali, S., & Slowik, A. (2021). Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. Neural computing and applications, 33, 13075-13088.
[55] Prodan, R., & Wieczorek, M. (2010). Bi-criteria scheduling of scientific grid workflows. *IEEE Transactions on Automation Science and Engineering, 7*(2), 364-376.
[56] Radulescu, A., & Van Gemund, A. J. (2000). Fast and effective task scheduling in heterogeneous systems. In Heterogeneous Computing Workshop, 2000.(HCW 2000) Proceedings. 9th (pp. 229-238). IEEE.
[57] Sahraei, S. H., Kashani, M. M. R., Rezazadeh, J., & Farahbakhsh, R. (2019). Efficient job scheduling in cloud computing based on genetic algorithm. *International Journal of Communication Networks and Distributed Systems, 22*(4), 447-467.
[58] Sawant, S. (2011). A genetic algorithm scheduling approach for virtual machine resources in a cloud computing environment.
[59] Sellami, K., Ahmed-Nacer, M., Tiako, P. F., & Chelouah, R. (2013). Immune genetic algorithm for scheduling service workflows with QoS constraints in cloud computing. *South African Journal of Industrial Engineering, 24*(3), 68-82.

[60] Sharieh, A., & Albdour, L. (2017). A Heuristic Approach for Service Allocation in Cloud Computing. *International Journal of Cloud Applications and Computing (IJCAC), 7*(4), 60-74.
[61] Shirazi, B., Hurson, A., & Kavi, K. (1995). Introduction to scheduling and load balancing. IEEE Computer Society.
[62] Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems, 52*(1), 1-51.
[63] Singh, S., & Kalra, M. (2014, November). Scheduling of independent tasks in cloud computing using modified genetic algorithm. In Computational Intelligence and Communication Networks (CICN), 2014 International Conference on (pp. 565-569). IEEE.
[64] Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European journal of Operational research, 47*(1), 65-74.
[65] Tayal, S. (2011). Tasks scheduling optimization for the cloud computing systems. *International journal of advanced engineering sciences and technologies, 5*(2), 111-115.
[66] Valarmathi, R., & Sheela, T. (2017). Ranging and tuning based particle swarm optimization with bat algorithm for task scheduling in cloud computing. Cluster Computing, 1-14.
[67] Verma, A., & Kaushal, S. (2013). Budget constrained priority based genetic algorithm for workflow scheduling in cloud.
[68] Wang, T., Liu, Z., Chen, Y., Xu, Y., & Dai, X. (2014, August). Load balancing task scheduling based on genetic algorithm in cloud computing. In Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on (pp. 146-152). IEEE.
[69] Wang, X., Wang, Y., & Zhu, H. (2012). Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm. *Mathematical Problems in Engineering*
[70] Wang, X., Yeo, C. S., Buyya, R., & Su, J. (2011). Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Generation Computer Systems, 27*(8), 1124-1134.
[71] Wu, Q., Yun, D., Lin, X., Gu, Y., Lin, W., & Liu, Y. (2012, May). On workflow scheduling for end-to-end performance optimization in distributed network environments. In Workshop on Job Scheduling Strategies for Parallel Processing (pp. 76-95). Springer, Berlin, Heidelberg.
[72] Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences, 270*, 255-287.
[73] Yassa, S., Chelouah, R., Kadima, H., & Granado, B. (2013). Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. *The Scientific World Journal*
[74] Yu, J., & Buyya, R. (2006). Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming, 14*(3-4), 217-230.
[75] Zanoon, N., Al-Haj, A., & Khwaldeh, S. M. (2017). Cloud Computing and Big Data is there a Relation between the Two: A Study. *International Journal of Applied Engineering Research, 12*(17), 6970-6982.
[76] Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009, September). Independent tasks scheduling based on genetic algorithm in cloud computing. In Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on (pp. 1-4). IEEE.
[77] Zhao, J., Zeng, W., Liu, M., & Li, G. (2011). Multi-objective optimization model of virtual resources scheduling under cloud computing and it.
[78] Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *IEEE Access, 3*, 2687-2699