Task Scheduling Algorithms for Cloud Computing: A Critical Review and Open Research Challenges

Gagandeep Kaur*, Anurag Sharma

Faculty of Computational Sciences, GNA University, Phagwara, India

*Corresponding author
doi: https://doi.org/10.21467/proceedings.114.50

Abstract

Cloud computing is the most preferred platform to access resources remotely. The benefit of cloud computing over traditional computing techniques is that it provides on-demand services and serves millions of users at the same time. However, scheduling the tasks of users is quite crucial in cloud computing. To overcome this challenge, various task scheduling algorithms have been designed for cloud computing. In this paper, we have done a critical review of various traditional and metaheuristic algorithms based on task scheduling algorithms. The critical review shows that the metaheuristic algorithms are better than traditional algorithms to find the optimal scheduling of the task. Besides that, based on the study, we have defined the open research challenges of the metaheuristic algorithms that help other authors to contribute their research in this field.

Keywords: Cloud Computing, Metaheuristic Algorithms, Task Scheduling, Virtual Machines.

1 Introduction

Recently, cloud computing is becoming one of the most widely used internet technology which allows different people as well as organizations to remote accessing computing resources such as platforms, software, and hardware. Cloud computing can be differentiated from conventional computing techniques because it provides: scalability, accessibility, adjustable costs, reliability, and on-demand services. Cloud computing can serve millions of users at the same time. It can meet the requests of all the users with high quality of services (QoS) with the best performance and guarantee of the quality of service (QoS). Therefore, an appropriate algorithm for task scheduling should be implemented that can efficiently and fairly meet user requests. Cloud performance mainly depends on the most critical parameter of task scheduling [1].

Task-scheduling algorithms are the set of policies or rules based on which tasks are assigned to suitable resources (e.g. memory, CPU, or bandwidth) for achieving the performances at the highest possible level. This type of scheduling algorithms has gained popularity due to its ability in solving the NP-hard problem and provides a solution with less effort of computational that makes it proper for large or complex task (NP-complete) problems. These algorithms normally offer the following advantages [1]:

- Managing the quality of services (QoS) and performance of cloud computing.
- Managing the CPU and memory.
- Maximizing the resource utilization ability and minimizing the task executing time.
- Increasing the success rate and fairness of all the tasks.
- Providing real-time scheduling for all the tasks.
- Achieving high throughput for the cloud computing system.
- Improving load balancing.

Further, task scheduling algorithms are differentiated into two broad categories i.e. static scheduling algorithms and dynamic scheduling algorithms. In static scheduling algorithms, all the tasks (or VMs) are known before actual scheduling. The tasks are usually independent of the state of the virtual machine. So,

Proceedings DOI: 10.21467/proceedings.114; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-8-8

a little runtime overhead is imposed on these tasks. Static scheduling is particularly suitable for cloud computing which is either small scale or medium scale. However, for dynamic scheduling algorithms, the information about the tasks is unknown in advance. So, the execution time for these tasks is also unknown and the information about VMs is not obtained until it comes into the scheduling stage. Dynamic scheduling is usually suitable for large size cloud computing environments [2-3].

The main contribution of this paper is to critically review the task scheduling algorithms that have been designed for cloud computing. Initially, we have studied the conventional task scheduling algorithms, followed by metaheuristic algorithms. From the study and analysis, we found that the metaheuristic algorithms provide better task scheduling than the traditional task scheduling algorithms. In the last, we have defined the open research challenges for the metaheuristic algorithms to enhance it.

The rest of the paper is as follows. Section 2 gives an overview of conventional and metaheuristic-based task scheduling algorithms. Section 3 shows a comparative analysis of task scheduling algorithms. Section 4 defines the open research challenges. In last, section 5 shows the conclusion of this paper

2 Overview of Conventional and Metaheuristic-based Task Scheduling Algorithms

Cloud computing is very strong in output depending on the scheduling of productivity tools. In most current planning algorithms, time, prices, pace, performance rate planning, resource use, and so on can be considered. However, primary metrics such as durability and usability should be taken into account. In the literature, various conventional and metaheuristic task scheduling techniques have been proposed. In this section, these techniques have been explained in detail.

2.1 Conventional Algorithms based Task Scheduling

In this section, we have studied the various conventional algorithms for cloud computing.

A. First Come First Served

It is also known as First in First out. This is one of the easiest and shortest techniques for scheduling. In the manner in which the process enters the central processing device specified. It is presumed that a ready queue is controlled as first, which indicates that the first work without any preferences is to be done first [21].

B. Min-Min Algorithm and Max-Min Algorithm

The execution and completion intervals of each task on each grid asset were calculated both by MAX-MIN and MIN-MIN Algorithm. The Min-min algorithm measures the execution times for each task on various resources and assigns the task with a minimum period and duration to the resource where the minimum time of execution is reached [22].

2.2 Metaheuristic Algorithms based Task Scheduling

In contrast with traditional methods, the metaheuristic algorithms have an important optimization structure plan. In the literature, the most preferred algorithms are Ant Colony Optimization (ACO), Genetic algorithm (GA), Particle Swarm Optimization (PSO), and Honeybee, etc. [4]. A new algorithm is proposed called Completion Time-Driven Hyper-Heuristic (CTDHH). The CTDHH algorithm is designed to optimize the cost of completion time in the scientific workflow for cloud computing scheduling. The algorithm utilizes four population-based meta-heuristic as a low-level heuristic algorithm (LLH). The algorithm has proven through the experimental results its usefulness in optimizing the cost of task scheduling for a cloud computing paradigm [5]. An improved genetic algorithm called N-GA is particularly useful for workflow scheduling in cloud computing. The N-GA algorithm exploits the benefits of the genetic algorithm along with heuristic methods. The algorithm evaluates its correctness and verified its behavior based on the model checker NuSMV and process analysis Toolkit (PAT) [6]. The purpose of using the PAT model is to get a platform to convert the proposed algorithm to SMV code while the NuSMV model checkers is the official base that is suitable for the modeling of the distributed system. Even more, this model has been implemented for tackling the gaps between the formal verification approach and actual application. The authors have also involved several behavioral models to enable selecting the algorithm that produces the best performance. The simulation results demonstrate that the suggested algorithm outperforms the other existing meta-heuristics algorithms in many aspects [6]. Lastly, an enhanced genetic algorithm is proposed for assigning the static workflow scheduling to the processor in a heterogeneous computing environment. The proposed algorithm introduces a new operator that guarantees sample diversity and stable coverage for all space. The proposed strategy replaces the random initial population with optimized solutions to reduce the repetitions in GA. The results of the experiments have shown that the proposed solution achieved a significant enhancement in terms of processor performance and cost [7]. Below are some of the most popular heuristic methods for task planning:

A. Firefly Algorithm

Yang et al. for the first time introduced a firefly's algorithm (FA) which was influenced by the behavior of fireflies. The observations indicated that the firefly algorithm has certain advantages like its ability to subdivide the problem and deal with constraints of modality [14]. Combining these two advantages can make exploitation and exploration of very balanced search spaces thus producing the best results. Continuous mathematic functions can be easily solved using FA [15]. FA is very promising in dealing with optimization problems and it is a sort of metaheuristic algorithm so setting parameters in any algorithm decides the ability of problem-solving. The familiar Job shop scheduling problems (JSSP) can be used to regularly test the metaheuristic algorithms [16].

Every VM or resource possesses its execution speed. The unit of execution speed is a million instructions per second (MIPS). Execution speed is denoted by$R_i \; where \; i \in [1, n]$. The length of each subtask can be represented in terms of million instructions and represented as L_i where $i \in [1,1]$. Execution time for this task can be obtained by dividing the task length with allocated VM speed and measured in seconds. This execution time for each firefly is provided by eqns. 1 and 2:

$$f_i = \sum_{i=1}^{l} \frac{L_i}{R_i} \tag{1}$$

$$F_i = \sum_{i=1}^{l} \frac{1}{f_i} \tag{2}$$

The algorithm aims at minimizing this total execution time of all the tasks (obtained by eqn. 1). Eqn. (2) provides the inverse relationship of execution time. More the execution time of tasks (f_i) of a particular firefly (i), less is the value of F for that particular firefly. Hence maximum fitness, F represents the best solution for finding a firefly (given by Eq. 3).

Most fit firefly = $max_i F_i$ (3)

The methodology for firefly algorithm is given below.

The firefly algorithms possess the following three ideal rules [17]:

1. All the fireflies can be attracted to one another without any consideration of their sex.
2. The attractiveness between one firefly and the other directly proportional to the brightness of another firefly. This attractiveness is reduced by increasing the distances between corresponding fireflies.
3. The firefly will start moving randomly if at any instant there is no brighter fireflies than it. Fig. 1 represents the Firefly algorithm.

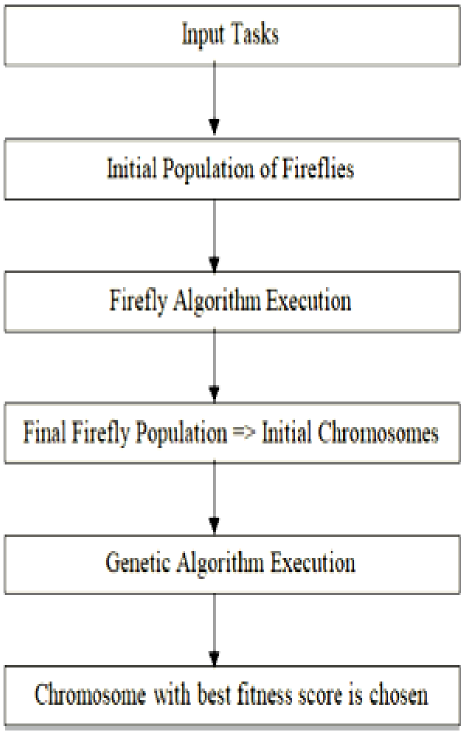

Fig. 1: Firefly Algorithm

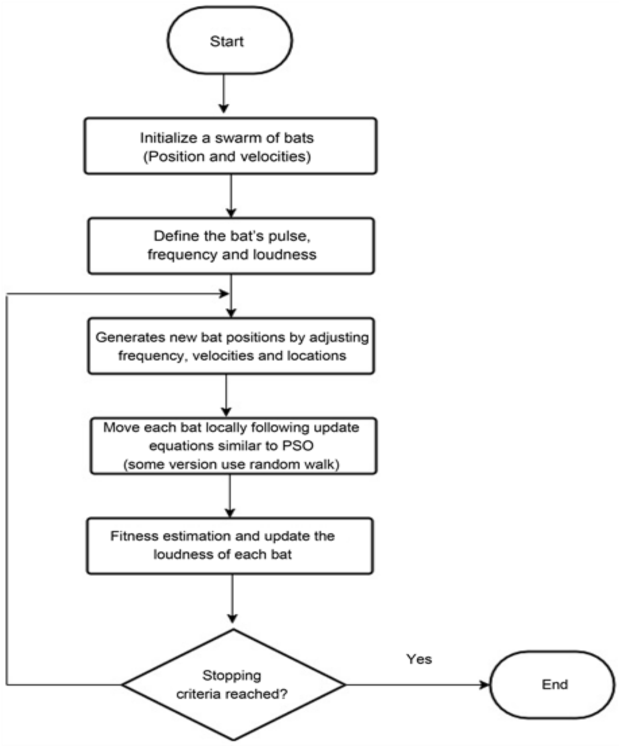

Fig. 2: BAT Algorithm

B. BAT Algorithm

Bat algorithm (BA), can be inspired by the microbat's foraging behavior and it has to turn out to be a dominant swarm intelligence technique for solving any optimization problem over discrete as well as continuous spaces. Recently, it finds applicability in successfully solving problems related to different areas of optimization [18]. Recent applications of the bat algorithm include solving discrete optimization problems. These discrete problems can be solved by converting the design variables from continuous to discrete domains for activating the Bat algorithms for solving them. The applicability and capability of Bat algorithms can be enhanced by suggesting some combinatorial problems. Hybridization and modification techniques can also be combined to enhance the capability of BA. The results indicated that BA is a potential metaheuristic algorithm inspired by nature for solving a lot of combinatorial problems. Furthermore, it offers significant benefits over other conventional algorithms [19]. Many modern optimization metaheuristic algorithms have been developed on swarm intelligence-based, which is inspired by nature. The ways to design these algorithms are diverse; however, they can be classified into three sources of inspiration: physical, social, and biological systems. Nowadays, many new nature-inspired metaheuristic algorithms have been suggested by the literature. One of them is the bat algorithm (BA) that is based on the foraging feature of microbats [20]. The firefly algorithm is as shown in fig. 2.

After initialization steps, the Bat algorithm comes its main phase where after each generation, other bats of the swarm move by updating their positions and velocities. The following equations represents the movement of bats:

$$f_i = f_{min} + (f_{min} - f_{max}) * \beta \quad (4)$$

$$sv_i^t = sv_i^{t-1} + \left[ax_i^{t-1} - x_*\right] * f_i \quad (5)$$

$$ax_i^t = ax_i^{t-1} + sv_i^t \quad (6)$$

Where β represents a number that is generated randomly in the interval of [0,1] and x_*represents the existing best solution available in the swarm. Also, sv_i^t represents the velocity of the bat 'i' and ax_i^trepresents the position of a bat 'i' at any time 't'.

C. Particle Swarm Optimization

This algorithm is based on the population search technique that uses the simulation of bird's social behavior as proposed by Awad et al. [23]. Several meta-heuristic algorithms such as PSO, ideal for the complex scheduling of the task, have been developed. Particle Swarm Optimization (PSO), on the other hand, has become common in a large variety of applications due to its simplicity and efficiency [24]. As per the PSO algorithm, the process's particles can be presented as per the below equation [23]:

$$AV_{id} = AV_{id} + ac_1 rand_1(P_{id} - AX_{id}) + ac_2 rand_2(P_{gd} - AX_{id}) \quad (7)$$

$$AX_{id} = AX_{id} + AV_{id} \quad (8)$$

In the above equations, AV_{id} and AX_{id} specify the velocity of the particle I at the particular iteration specified as d. The variables rand1 and rand2 are the randomly generated variables between 0 and 1. aC1 and ac2 specify the learning factors that are also called as the social parameters [24]. Fig. 3 shows the flowchart of the PSO algorithm.

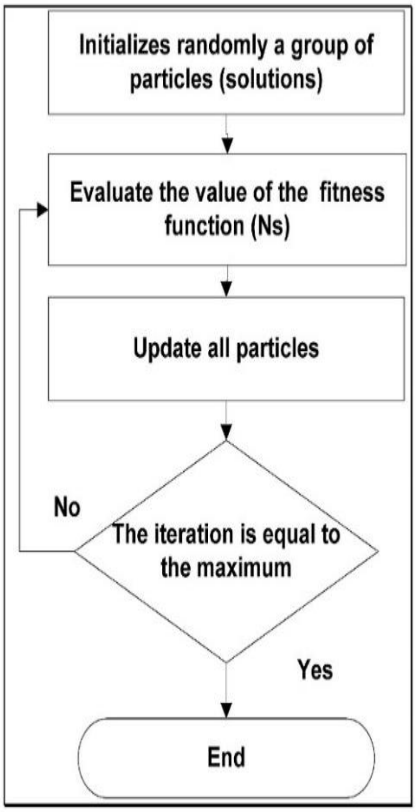

Fig. 3: PSO Algorithm

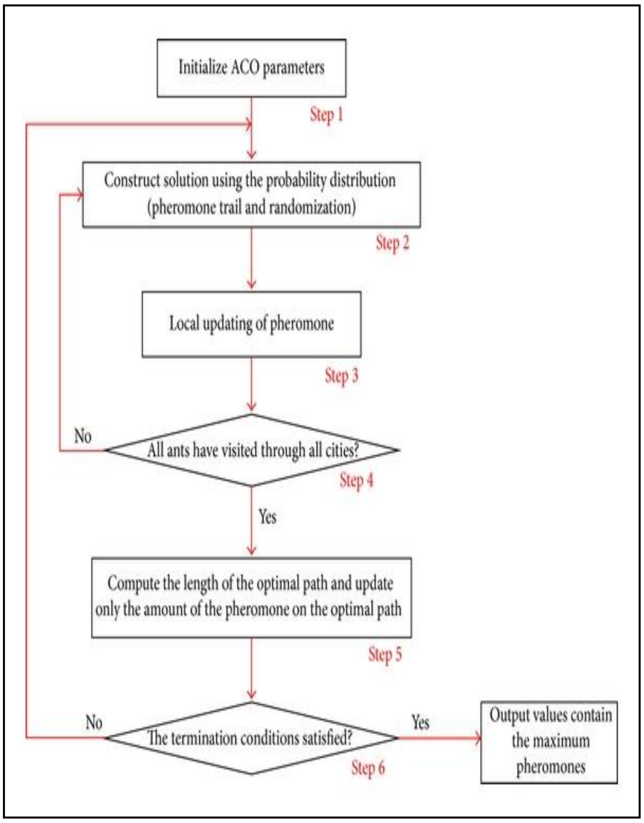

Fig. 4: ACO Algorithm

For task scheduling in cloud computing using the PSO algorithm, the tasks and virtual machines are considered as input. In the first step, the population is initialized using the required supplied parameters. Then the value of the objective function is calculated using the equations. The position of present and global best particle is updated after that. In the next step, the interim weight is updated using the proposed approaches according to the number of tasks. After that, the position and velocity are updated and rounded to the nearest integer. These steps iterate for the pre-specified number of iterations [25]. Al-Maamari and Omara in 2015 used the LBMPSO to reduce costs, decrease latency, reduce execution time, decrease transfer time, obtain load balance between activities and the virtual computer and evaluate the resources available and minimize the complexities of cloud computing. In contrast to other algorithms, LBMPSO increases cloud storage stability and the optimal allocation of works to services. They also observed that the PSO mutation will accomplish the highest relative to other algorithms during trip time load balance.

Furthermore, when dividing tasks into services, the suggested algorithm takes into consideration the load balance, assigns tasks as quickly as possible, is finalized faster, and re-plans failure tasks. It is ideal for different activities and services [24]. In 2019, Huang et al. proposed a task scheduler for cloud computing software management with different discrete versions of the particle swarm optimization (PSO) algorithm. These methods were contrasted with three well known evolutionary methods on task planning problems to test the output. The findings of the experiments show that the suggested methods are productive and reliable. A reasonable alternative is to use a logarithm reduction technique to have an optimum schedule for the suggested PSO-based scheduler. In contrast with the contrast gravitational search algorithm, artificial bee colony algorithm, and dragonfly, the average complement of the suggested PSO-based scheduler following a logarithm reduction is reduced by 19.12 percent to 21.42 percent and 15.14 percent for each [25].

D. Ant Colony Optimization

For all types of scheduling issues, the ant colony algorithm has been used, obtaining a successful outcome. However, the latest reports have neglected to detail the demands of the different tools for these projects. Tools and roles are common in cloud computing. Some functions for the CPU, for example, involve high demand and others involve extra capacity. The expense of multiple services differs. There are, thus, various job costs. Therefore, as they analyze in-depth the demand gap for services, it is beneficial to represent costs. But it is simple to collapse into a local maximum using the ant colony algorithm. This paper, therefore, recommends an enhanced ant colony algorithm that can measure and modify the solution efficiency to prevent dropping into the optimal locality. The key goals of this scheduling approach are efficiency and budgetary expenses, which are called PBACO since it is based on the colonial ant optimization algorithm [10]. The flowchart of the ACO algorithm is shown in Fig. 4. In the research by Tawfeek et al., the ACO algorithm has been introduced to schedule cloud computing activities. Firstly, experimentally calculated the strongest parameter values for an ACO algorithm. In applications of job numbers ranging between 100 and 1000, the ACO algorithm was then tested. Results of the simulation showed that FCFS and round-robin algorithms are superior to ACO's. The impact of precedence among tasks and load balance will be taken into consideration in future work [26]. Guo in 2017 introduced the MO-ACO algorithm, which considers the balance between equipment, expense, and load. The algorithm defines the limiting feature of completion time, expense and fee, strengthens the heuristic simple ant colony algorithm and the pheromone update rule, and adopts the random chance law pseudo-random in the colony method. They checked the algorithm in the cloudim, and the test showed that the algorithm was successful [27].

The ant colony optimization algorithm is an algorithm that is spread to solve problems of combinatory optimization. The scheduling phase is completed by the algorithm designed to simulate the foraging phase. First of all, the ants randomly select a path. The ants measure the fitness route, until they achieve their target objectives, and to which stage ants set pheromones on the fitness route. Finally, the pheromone and the behavioral choices must be changed to concentrate the ants on the high fitness route and to obtain the optimum response as often as possible [10]. The following steps describe the cloud-based scheduling using ant colony optimization algorithm [27]:

Step 1: Initialize the pheromone

The initial value of the pheromone on the path between the virtual machine and task is calculated using the below equation:

$$\tau_0 = v_{comp\_i} / v_{avgcomp} \quad (9)$$

In the above equation, $v_{avgcomp}$defines the average value of MIPS of virtual machines calculated as:

$$vm_{avgcomp} = \frac{\sum_{j=1}^{n} v_{comp\_i}}{m} \quad (10)$$

Step 2: Selecting the rule for the next task

For selecting the rules for upcoming tasks, the transition techniques like ACS can be used. Here, several parameters of virtual machines are considered average time for execution, load, total execution time etc.

Step 3: Local update rules for Pheromone

When a pheromone/ant on the specific scheme map path fits the right virtual machine for any mission, it is modified locally. The following rules for updates are considered:

$$\tau_{ij}(ta+1) = (1-\rho)\tau_{ij}(ta) + \rho\Delta\tau_{ij}(ta) \quad (11)$$

In the above equation, ρ represents the factor of pheromone volatilization which shows the degree of pheromone's volatilization for per unit of time. Metric 1- ρ shows the residual pheromone's degree between 0 and 1. The largest value of ρ indicates the fast volatilizing of pheromones [27].

Step 4: Global update rule for Pheromone

If all the ants perform an iteration, the optimum correspondence scheme is reached for this iteration and the optimum global matching scheme is attained by contrasting iteration with the global optimum matching scheme. Then, on the global optimal method, it changes the pheromone on the mapping direction between the task and the virtual machine [27]. The rules to update the global best is shown below:

$$\tau_{ij}(ta+1) = (1-\rho_1)\tau_{ij}(ta) + \rho_1\Delta\tau_{ij}(ta) \quad (12)$$

In the above equation, $\rho_1$represents the factor of volatilization, and 1-$\rho_1$shows the residual pheromone of degree ranging between 0 and 1. The value of $\Delta\tau_{ij}$is calculated using the below equation:

$$\Delta\tau_{ij}(ta) = \begin{cases} \frac{1}{L_{best}}, if(i,j) \in T^+, \\ 0, otherwise \end{cases} \quad (13)$$

In the above equation, L_{best} represents the objective function's optimal solution starting from the beginning iteration to the current iteration.

F. Genetic Algorithm

In 2012, Jang et al. proposed cloud computing scheduling problems and suggested a paradigm of task planning for addressing the schedule difficulties. The task developer calls the GA scheduling feature to draw up the job schedules using tasks and virtual machine details in the proposed task scheduling model. The GA programming function generates a community, a variety of task plans and tests the community by utilizing the fitness method to take into account consumer happiness and availability of virtual machines. The role iterates communities to generate the best schedule of activities. The restart procedure is often used to increase the consistency of work schedules in the GA scheduling feature. They simulated the conceptual task preparation paradigm for success evaluations and carried out different tests. The results from the analytical observations show that the suggested model of task planning outperforms from the other model of task planning which is the round-robin task schedule, the load-index task scheduling model, and the task scheduling model dependent on operation [28].

A genetic algorithm is a method of natural selection in the estimation of human development and genetic system biological processes. Following the generation of the original population, more and greater fitness solutions are achieved from generation to generation. The individual is selected in each generation based on a certain optimization goal in terms of the wellbeing of the various individuals. The individuals will then cross the genetic operators and mutate, and a new species will then be created which will constitute a new field for study. The right approach is chosen and over many years of growth, the bad solutions can be discarded. Depending on the present cloud storage scenario, an advanced genetic algorithm-based scheduling approach is proposed [29].

Step 1: Encoding and decoding of chromosomes

There are several methods to encode chromosome, either explicitly or implicitly. This study uses the way to implicitly encode the easy and intuitive workforce task. The chromosome length is precisely the same as the amount of function and gene value as the number of workers for this encoding. Assuming that six tasks, 3 technicians are available, the chromosome length is 6, and the gene differs between 1 and 3, the below chromosome is generated [29]:

$$\{2, 1, 3, 1, 3, 2\}$$

The second worker performs the first job, the first worker performs the second task, and so forth. The above encoding lets the chromosomes be decoded so that each worker may assign activities. The job shall be classified by the employee and several job sequences shall be developed with several employees. The decoders of the chromosomes are:

$$w1:\{3,5\}\ w2:\{1,4\}\ w3:\{2,6\}$$

Step 2: Population Initialization

Suppose S is the population, W is the number of employees, and N the number of tasks. The initialization then defined as: randomly generated S chromosomes; the N is the chromosome length and a gene ranges between 1 and W.

Step 3: Fitness function

The quality criterion for people in the population is the fitness function. Fitness is the basis for genetic operators, like selecting, crossover, and mutating. It directly represents individuals' performance, algorithm convergence, and searching for the optimal best solution. The higher the fitness, the stronger the performance. Individuals with improved fitness can reproduce and those may decline from generation to generation. The guiding force of genetics algorithms is the fitness function.

Task scheduling is a challenge in multi-target optimization. They describe two fitness functions in this study. The one with less variation in load and cost consumption is fitter.

$$Fit_1 = \frac{1}{totalcost} \tag{14}$$

$$Fit_1 = \frac{1}{\propto(T)} \tag{15}$$

Step 4: Strategy for Selection

About the algorithm for choosing a roulette wheel, the crossover and transition operations relate to community diversity. To avoid this scenario, excellent individuals without any operators are copied to the next stage. In this paper, some of the strongest genes can be provided to the next generation from now. It has been shown that genetic algorithms have a strong global convergence with elite selection operators.

Step 5: Operation Crossover and Mutation

Crossover decides the local genetic algorithm quest power that inherits healthy genes in the next generation and creates new individuals with improved genetic structure. Each person has a certain risk of interacting with another person. There is a convergence of one point and convergence of several points. As the populace converges globally, this will hold populations diverse and deter premature phenomena. The mutated community will broaden different study fields [29].

2.3 Hybrid algorithms for task scheduling in cloud computing

Research by Taj and Basu proposed the hybridization of the Genetic and Group Search Optimization Algorithm for scheduling the tasks in a cloud computing environment. This study aims at preparing the job based on the GGSO algorithm. The GA is built between their hybrid optimization algorithm and group optimization algorithm. The benefits of GA and group search optimization algorithms are utilized thus ignoring their disadvantages. Through hybridizing these two optimization algorithms, the drawbacks of the

ISBN: 978-81-947843-8-8
ISSN: 2582-3922

GSO and GA one can be resolved and it has the benefits of being readily and rapidly converged so that in a shorter computational period this scheduling strategy will have an ideal or suboptimal solution. In their assumptions, the result segment indicates that the GGSO optimization suggested is more effective than the person optimization. They initially used the GSO algorithm in their approach to the problem, which defines the best work preparation for the subsequent task. Finally, a crossover operator eliminates the poorest population [30].

Algorithm

Step 1: Solution encoding
Step 2: Fitness function evaluation
Step 3: Producer operation
Step 4: Scrounger Operation
Step 5: Ranger Performances
Step 6: Crossover
Step 7: Criteria of termination

In the GGSO any dimension of a method is a job, and all job goals are a solution as a whole. The key issue is how consumer roles may be delegated to optimize IaaS providers' profits while maintaining QoS. This method can obtain a high-quality scheduling solution by hybridizing two optimization algorithms such as the GA and the GSO. The approach is capable of ensuring utilization standard (QoS) and enhancing the reputation and financial advantage of IaaS providers. Experimental impacts have shown that the GGSO dependent scheduling solution is suitable for the task scheduling issue [30].

In 2020, Natesan and Chokkalingam proposed a new hybrid algorithm that can be used for task scheduling in cloud computing. In the creation of a new hybrid algorithm named Whale Genetic Optimization Algorithm they have merged two meta-heuristic strategies, Whale Optimization Algorithm (WOA) and GA (Genetic Algorithm). They aim to reduce the equipment and expenses through task preparation. The Cloudsim toolkit is used to replicate this. The findings indicate a considerable decrease in the implementation period calculated by change rates. The findings were contrasted to the traditional WOA and GA. The observations of the technologies suggested include a better performing scheduling method [31].

Pradeep and Jacob proposed the cuckoo and gravitational algorithm (CGSA) hybridization. The core principle of the strategy is to take advantage of both the CS and GSA algorithms, thus ignoring their disadvantages. Based on various measurement criteria, the performance of the algorithm is assessed. Comparative research is carried out using algorithms such as GSA, CS, PSO, and genetic algorithms (GA). The experimental findings indicate that the algorithm is stronger than the current methods [32].

In the paper by Srichandan et al., the role of scheduling algorithms was studied using a hybrid methodology incorporating beneficial characteristics of some of its most frequently utilized biological heuristic algorithms, GE and BF algorithms in the cloud. This paper has two key contributors. First, the algorithm of scheduling minimizes labeling and secondly, reducing energy use, economically and environmentally sustainable. Experimental findings reveal that, concerning consistency, reliability, and heterogeneity of solutions, the outputs of the proposed algorithm are beyond that of other algorithms [33].

3 Comparative Analysis of Task Scheduling Algorithms

Task Scheduling (TS) particularly useful for the cloud computing paradigm is an open issue and has a lot of challenges so it has been recently addressed in many studies. In this section, some of these studies are reviewed. In cloud computing environments, there are many algorithms for preparation on both sides. The most suitable functions are set by a variety of parameters including time, expense, benefit, energy usage, storage, memory requirements, throughput, etc. The stronger the trust between the customer and the service supplier, the easier the usage of services and the preparation strategy can be. Different metaheuristic scheduling algorithms are reviewed for cloud computing (CC) and the main issues and challenges of metaheuristic algorithms are presented. Further, the authors provided extensive discussions about different metaheuristics algorithms in cloud computing [8]. Next, a comparative study of different metaheuristic techniques is presented for the cloud computing environment which includes: BAT algorithm, Ant colony optimization (ACO), League Championship Algorithm (LCA), Particle Swarm Optimization (PSO), and Genetic Algorithms (GA), etc. The results presented in the paper suggested that the PSO algorithm offers faster performance as compared with the GA algorithm for solving a static TS problem in a cloud computing environment [9]. Different task scheduling algorithms like GA, Minimum Completion Time (MCT), Large Cloudlet Fast Processor (LCFP), and Small Cloudlet Fast Processor (SCFP). Their solution has minimized the makespan and maximized the processor utilization [10]. ACO is an arbitrary search algorithm, which uses positive feedbacks and follows the actual ant colony behaviors.

This algorithm can be used for allocating optimal sources for tasks scheduling with minimal execution time in a dynamic cloud computing environment [11]. The further hybrid scheduling algorithm is proposed using ant colony optimization (ACO) and genetic algorithm (GA). The benefits of both algorithms are used for solving TS problems. The proposed hybrid algorithm utilizes the global search engine present in the genetic algorithm for reaching the optimized solutions at a faster rate. Also, initial values for pheromones are utilized in the ACO algorithm [12].

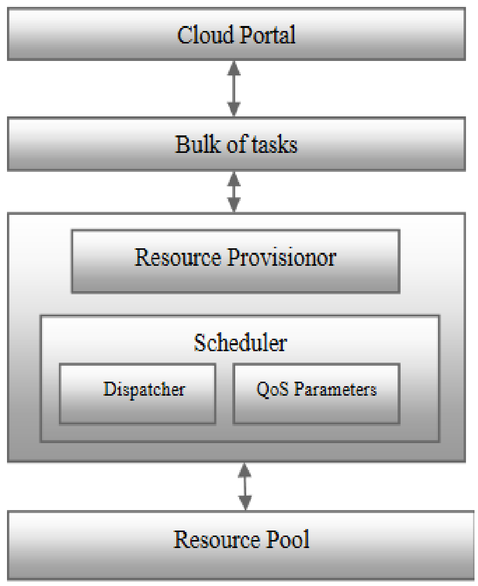

Fig. 5: Flowchart of Cloud task scheduling

In Fig. 5, the working of cloud task scheduling is shown. The client submits the tasks through its cloud portal. Resources are not directly allocated to the client, instead of before provisioning the resources scheduler firstly checks the resources database based upon the quality of service parameters. After verifying the QoS parameters and resource availability, the scheduler allocates the resources from the resource pool. After completion of tasks, resources are returned back to the resource pool. The scheduler used several strategies to schedule the tasks i.e. static as well as dynamic [13].

Next, we have done the comparative analysis of the scheduling algorithms in below Table 1

Table 1: Comparative Analysis of Task Scheduling Algorithms

Sr No.	**Paper Title**	**Journal/ Conference Name**	**Author (Year)**	**Finding**
(1)	Optimal Scheduling of Tasks in Cloud Computing Using Hybrid Firefly-Genetic Algorithm [17]	Advances in decision sciences, image processing, security and computer vision	Rajagopalan et al. (2020)	It is found that GA needs fewer knowledge about the situation, but it may be challenging to build an appropriate function and correct the interpretation and operators. GA is cost computerised, i.e. time consuming.
(2)	Multi-Objective Task Scheduling Using Hybrid Whale Genetic Optimization Algorithm in Heterogeneous Computing Environment [35]	Wireless Personal Communications	Gopalakrishnan N atesan, · Arun Chokkalinga m(2020)	This paper reveals that Initially, genetic algorithms choose a random group regardless of the optimal answer. So, time is highly complicated.

(3)	A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments [36]	Cluster Computing	Laith Abualigah and Ali Diabat (2020)	It is found that the algorithm of Ant Lion Optimization has several handicops, including long-term instability and prematurity with many.
(4)	A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system [37]	Cluster Computing	Jun-qing Li Yun-qi Han (2019)	Artificial Bee Colony algorithm is a swarm-based algorithm which is utilised inappropriately in the resolution of complex problems.
(5)	Multi-Objective Task Scheduling Using Hybrid Genetic-Ant Colony Optimization Algorithm in Cloud Environment [38]	Wireless Personal Communications	A. M. Senthil Ku mar and M. Venkatesan (2019)	It states that the theoretical study of the Ant colony is complicated and the distribution of possibilities is modified through iterations.
(6)	A novel water pressure change optimization technique for solving scheduling problem in cloud computing [39]	Cluster Computing	Nasr et al. (2019)	At the low time difficulty, preparation is accomplished through the optimization strategy of water pressure transition. It is motivated by the phenomena of changes in water intensity when strain is small regardless of shifts in water's physical properties. It also increases the schedule duration, load balance, resource use, and memory utilization and cloud server capacity.
(7)	A Multi-objective Optimal Task Scheduling in Cloud Environment Using Cuckoo Particle Swarm Optimization [40]	Wireless Personal Communications	T. Prem Jacob and K. Pradeep (2019)	This paper found that the PSO algorithm normally suffers from local convergence rate.
(8)	An EDA-GA Hybrid Algorithm for Multi-Objective Task Scheduling in Cloud Computing [41]	IEEE Access	Pang et al. (2019)	The EDA-GA hybrid scheduling algorithm is based on EDA (distribution algorithm estimate) and GA (genetic algorithm) was modelled by the researcher. The first thing to do is to create a certain amount of viable solutions using the EDA likelihood model and process. Secondly, GA's searche solutions are extended with the crossover and mutation activities. The optimum planing technique is eventually applied to delegate activities to virtual machines. This algorithm profits from fast convergence and a good capacity to search.
(9)	Task scheduling for cloud computing using multi-objective hybrid bacteria foraging Algorithm [33]	Future Computing and Informatics Journal	Srichandan et al. (2018)	Researchers also hybridised multi-target forging of bacteria and genetic algorithm. First of all, the algorithm of schedulation minimises the make-up and secondly, reducing both economic and ecological energy usage. Experimental findings indicate that the efficiency in consistency, reliability and richness of solutions of the proposed algorithm is superior to other algorithms.
(10)	A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments [42]	ICT Express	Negar Dordaie, Nima Jafari Navimipour (2018)	The PSO is easily subjected to partial optimism, resulting in less accurate pace and direction management.

4 Open Research Challenges

Based on the study and analysis, we found the following research challenges on which further work is possible.

1. In the GA algorithm, the parents are selected randomly to generating new offsprings to determine the optimal solution. However, if the selected parents are inappropriate then determining the optimal solution is not possible. To overcome this limitation, one appropriate solution is to select parents optimally then the GA algorithm is applied for task scheduling.

2. The PSO algorithm provides a global solution as compared to the genetic algorithm but the limitation of the PSO algorithm is that it falls into the local optimum solution. To overcome this issue, one solution is to hybrid the PSO algorithm with other algorithms to update the position of particles.

3. The firefly algorithm provides high complexity and a low convergence rate as compared to the other algorithm. One solution is to explore the other algorithm that provides lesser complexity and high convergence rate. The tripartite cloud development model, on the one side, allows for sustainable, secure, high-quality production resources and on the other side, poses a range of problems and challenges for cloud planning. The main challenge is to ensure that the members in and the group (e.g. the supplier, user, and consumer) are continuously involved. Since any user is an independent agency with its own target and desires, their own needs should be equilibrated and their priorities fulfilled in order to ensure that they continue to be active in cloud development. However, there is often disagreement between the goals and priorities of various people. A significant question in this situation is how to align your desire and achieve your goals. They will withdraw from the scheme if they do not meet their goals or ambitions [34].

4. Overall, the research by Zuo et al. [10] proposed an enhanced optimization algorithm for ant colonies in order to solve the multifactor optimization problem. With two efficiency and budget, limitations functions the enhanced ant colony optimization algorithm will analyse and change the solution output in time. The enhanced ant colony algorithm then addresses the dilemma that the initial ant colony algorithm is only optimally local [10]. But there are challenges if this algorithm can be merged with any other algorithm to improve the performance of the cloud computing task scheduling. As already stated, a variety of algorithms for this issue have been suggested, there are still some opportunities for developing a more effective algorithm. The suggested method should be optimal, it takes less memory, it should use less resources and have a rapid response time, a lower waiting period, a lower downtime, and a lower rotation period. Also, an algorithm for all requests can execute as few migrations as necessary. Both of these problems can then be considered to incorporate the latest algorithm for the cloud service schedule.

5 Conclusion

The scalability, expense adaptability, simplicity, flexibility, and on-demand pay-as-you-go applications separate cloud computing from standard computing paradigms. Since cloud storage concurrently supports millions of customers, all customer requirements for high efficiency and quality of service assurances (QoS) have to be fulfilled. Therefore, we must incorporate a reasonable algorithm to schedule assignments in order to address these demands equally and effectively. The challenge of task preparation is one of the most significant challenges in the cloud world since Cloud success is largely based upon it. Task scheduling in the cloud setting is one of the major functions. The primary goal of scheduling is to optimize resource utilization and reduce the workload. The activities to ensure service quality, productivity and justice can be balanced by a task strategist. In the cloud computing environment, task scheduling is very important to reduce the overall cost and to use the resources effectively. In this paper, various meta-heuristic techniques of task scheduling in cloud computing were discussed along with some hybrid techniques that overcome the shortcomings of the original techniques. It is found that using the meta-heuristic algorithms, the task scheduling in a cloud computing environment can be enhanced and improved. In the future, this research aims to find more hybrid meta-heuristic techniques that can further show the improvements.

References

[1] Aladwani T. Types of Task Scheduling Algorithms in Cloud Computing Environment. Book Chapter, April 2020.
[2] Al-maamari A, Omara F. (2015). Task scheduling using the PSO algorithm in cloud computing environments. International Journal of Grid Distribution Computing, 8(5), 245-256.
[3] Endo P, Rodrigues M, Gonçalves G, Kelner J, Sadok D, Curescu C. (2016). High availability in clouds: Systematic review and research challenges. Journal of Cloud Computing Advances, Systems, and Applications.
[4] Banga, P., and S. Rana, 2017. Heuristic-based independent task scheduling techniques in cloud computing: A review. Int. J. Comput. Applic., 166: 27-32.
[5] Dhaenens, C., and L. Jourdan, 2016. Metaheuristics – A Short Introduction. In: Metaheuristics for Big Data, Dhaenens, C.C. and L. Jourdan (Eds.), John Wiley and Sons, Inc, Hoboken, USA, pp: 23-52.
[6] Kalra, M. and S. Singh, 2015. A review of metaheuristic scheduling techniques in cloud computing. Egyptian Inform. J., 16: 275-295.
[7] Alba, E., G. Luque and S. Nesmachnow, 2013. Parallel metaheuristics: Recent advances and new trends. Int. Trans. Op. Res., 20: 1-48.
[8] Huang J, "The Workflow Task Scheduling Algorithm Based on the GA Model in the Cloud Computing Environment." Journal of Software, Vol. 9, No 4, PP. 873-880, April 2014.
[9] Raghavan S, Marimuthu, C, Sarwesh, P, & Chandrasekaran K., "Bat algorithm for scheduling workflow applications in cloud", International Conference Electronic Design, Computer Networks & Automated Verification (EDCAV), IEEE, pp. 139–44, 2015.
[10] Zuo L, Shu L, Dong S, Zhu C, "A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing", Article in IEEE access 3:1-1·December 2015.
[11] Rani T. S, Kannan S, "Task Scheduling on Virtual Machines using BAT Strategy for Efficient Utilization of Resources in Cloud Environment," International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6663-6669.
[12] Gitanjali J. et. al., "Scheduling Tasks Using Bat's Algorithm in Cloud Sim", International Journal of Pure and Applied Mathematics, Vol. 118, No. 18, 3919-3928, 2018.
[13] Sagnika S. et. al., "Workflow Scheduling in Cloud Computing Environment using Bat Algorithm," Proceedings of First International Conference on Smart System, Innovations and Computing", DOI:10.1007/978-981-10-5828-8_15, Jan 2018.
[14] Yang X-S, He X (2013) Firefly algorithm: recent advances and applications. Int J Swarm Intell 1:36–50
[15] S. Lukasik and S. Zak, "Firefly Algorithm for continuous constrained optimization Tasks", Lecture Notes in Computer Science, 5796, 2009, pp. 97-106.
[16] Faten A. Saif, M N Derahman, Ali A. Alwan, Rohaya Latip, "Performance Evaluation of Task Scheduling using Hybrid Meta-heuristic in Heterogeneous Cloud Environment," International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.6, pp.3249-3257, 2019.
[17] Rajagopalan, Aravind & Modale, Devesh & Senthilkumar, Radha. (2020). Optimal Scheduling of Tasks in Cloud Computing Using Hybrid Firefly-Genetic Algorithm. 10.1007/978-3-030-24318-0_77.
[18] Zebari, Amar & Almufti, Saman & Abdulrahman, Chyavan. (2020). Bat algorithm (BA): review, applications and modifications. International Journal of Scientific World. 8. 1. 10.14419/ijsw.v8i1.30120.
[19] Yang, X.S., Gandomi, A.H. (2012) Bat Algorithm: A Novel Approach for Global Engineering Optimization. Eng. Comput. 2012, 29, 464–483.
[20] Sathya, M.R., Ansari, M.M.T. (2015) Load frequency control using Bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system. Int. J. Electr. Power Energy Syst., 64, 365–374.
[21] Shameer, A. P., & Subhajini, A. C. (2016). Study on Different Scheduling Algorithm for Cloud Computing. International Journal of Advanced Research in Computer Science and Software Engineering, 6(5), 895-900.
[22] Joundy, M., Sarhan, S., & Elmougy, S. (2015). SCHEDULING ALGORITHMS IN CLOUD COMPUTING: A COMPARATIVE STUDY. International Journal of Intelligent Computing And Information Science, 15(4), 1-10.
[23] Awad, A. I., El-Hefnawy, N. A., & Abdel_kader, H. M. (2015). Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Computer Science, 65, 920-929.
[24] Al-Maamari, A., & Omara, F. A. (2015). Task scheduling using PSO algorithm in cloud computing environments. International Journal of Grid and Distributed Computing, 8(5), 245-256.
[25] Huang, X., Li, C., Chen, H., & An, D. (2019). Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies. Cluster Computing, 1-11.
[26] Tawfeek, M. A., El-Sisi, A., Keshk, A. E., & Torkey, F. A. (2013, November). Cloud task scheduling based on ant colony optimization. In 2013 8th international conference on computer engineering & systems (ICCES) (pp. 64-69). IEEE.
[27] Guo, Q. (2017, April). Task scheduling based on ant colony optimization in cloud environment. In AIP Conference Proceedings (Vol. 1834, No. 1, p. 040039). AIP Publishing LLC.
[28] Jang, S. H., Kim, T. Y., Kim, J. K., & Lee, J. S. (2012). The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, 5(4), 157-162.
[29] Yin, S., Ke, P., & Tao, L. (2018, May). An improved genetic algorithm for task scheduling in cloud computing. In 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 526-530). IEEE.
[30] Taj, N., & Basu, A. (2019). Hybridization of Genetic and Group Search Optimization Algorithm for Deadline-Constrained Task Scheduling Approach. Journal of Intelligent Systems, 28(1), 153-171.
[31] Natesan, G., & Chokkalingam, A. (2020). Multi-Objective Task Scheduling Using Hybrid Whale Genetic Optimization Algorithm in Heterogeneous Computing Environment. Wireless Personal Communications, 110(4), 1887-1913.
[32] Pradeep, K., & Jacob, T. P. (2018). CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment. Information Security Journal: A Global Perspective, 27(2), 77-91.
[33] Srichandan, S., Kumar, T. A., & Bibhudatta, S. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. Future Computing and Informatics Journal, 3(2), 210-230.
[34] Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, *57*(15-16), 4854-4879.
[35] Natesan, G., & Chokkalingam, A. (2020). Multi-Objective Task Scheduling Using Hybrid Whale Genetic Optimization Algorithm in Heterogeneous Computing Environment. Wireless Personal Communications, 110(4), 1887-1913.
[36] Abualigah, L., & Diabat, A. (2020). A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Computing, 1-19.
[37] Li, J. Q., & Han, Y. Q. (2019). A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system. Cluster Computing, 1-17.
[38] Kumar, A. S., & Venkatesan, M. (2019). Multi-Objective Task Scheduling Using Hybrid Genetic-Ant Colony Optimization Algorithm in Cloud Environment. Wireless Personal Communications, 107(4), 1835-1848.

[39] Nasr, A. A., Chronopoulos, A. T., El-Bahnasawy, N. A., Attiya, G., & El-Sayed, A. (2019). A novel water pressure change optimization technique for solving scheduling problem in cloud computing. Cluster Computing, 22(2), 601-617.
[40] Jacob, T. P., & Pradeep, K. (2019). A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. Wireless Personal Communications, 109(1), 315-331.
[41] Pang, S., Li, W., He, H., Shan, Z., & Wang, X. (2019). An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing. IEEE Access, 7, 146379-146389.
[42] Dordaie, N., & Navimipour, N. J. (2018). A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. ICT Express, 4(4), 199-202.

Proceedings DOI: 10.21467/proceedings.114

Series: AIJR Proceedings
ISBN: 978-81-947843-8-8
ISSN: 2582-3922