

Chapter 35

Task Scheduling in Cloud Computing Using Hybrid Meta-Heuristic: A Review



Sandeep Kumar Patel and Avtar Singh

1 Introduction

Cloud computing is an on-demand availability of shared resources, i.e., storage, computation power, network, software, and other services to fulfill client requests in small time and cost over the internet. The advantages include resource-transparency, reliability, affordability, flexibility, location—independence and a high availability of services [1]. To achieve these functionalities, a proper task scheduling is required so that it can provide a good performance in a swift manner. Moreover, cloud computing aims to satisfy the customer requirements in view of the Service Level Agreement (SLA) and the Quality of Service (QoS) [2]. There exist basically three service models viz.; 1. Platform as a service (PaaS), 2. Infrastructure as a service (IaaS) and 3. Software as a service (SaaS) which can be deployed on various deployment models like Private Clouds, Public Clouds and Hybrid Clouds [3].

Virtualization allows sharing a single instance of a resource among multiple people, e.g., server, network, desktop, operating system. It is used to display the hallucination, rather than actual, of many isolated virtual machines. Each VM runs many guest operating systems to ensure the heterogeneity of application. In this scenario, Hypervisor plays a major role as it assists the interaction between guest OS and physical hardware [4].

A key concept in cloud computing is the Resource Management which is implemented in two stages. The first stage—Resource Provisioning, provides means for

S. K. Patel (✉) · A. Singh
Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of
Technology, Jalandhar, Punjab 144011, India
e-mail: sandeep.pmg61@gmail.com

A. Singh
e-mail: avtars@nitj.ac.in

the selection, deployment and management of software from task submission to task execution as requested by an application. The second stage—Task Scheduling, is the process of mapping of various incoming tasks to existing resources to achieve an optimal execution time and an efficient resource utilization [5]. The total completion cost of any task is the summation of the communication cost and execution cost of that task. The Data transfer cost may also be considered for large data transfers. To minimize this cost, resources are equally distributed among the tasks.

In this research area, numerous studies have been done over the years with meta-heuristic techniques being the most prevalent in the literature. Singh et al. [6] present a summarized study of various meta-heuristic optimization techniques employed in the cloud computing environment for task scheduling. Our study is the only one that focuses on hybrid techniques. The primary objective of the study is to conduct a proper systematic comparative analysis of various hybrid distinctions based on the metrics like makespan, cost, throughput and energy consumption. Aiming to infer intrinsic behavioral properties to these algorithms and assist in the appropriate and efficient hybridization. To build a roadmap for future studies is the ultimate outcome of this research.

The organization for the rest of the paper is as follows: Sect. 2 presents a brief description of the task scheduling in the cloud environment. In Sect. 3, various optimization techniques are discussed. In Sect. 4, a literature review of hybrid meta-heuristic techniques for scheduling is presented. Sections 5 and 6 gives a tabular summary of the related works and comparison of performance metrics respectively and the conclusion and the future work are presented in Sect. 7.

2 Task Scheduling in Cloud

The task scheduling in cloud environment is an NP-complete problem so it is hard to find an optimal solution in polynomial time. The scheduling in cloud improves the resource utilization and reduces the overall completion time. There does not exist a standard task scheduling technique that could be extended to a large-scale environment. The main job of task scheduler is to distribute customer requests to all the present resources to execute them. Task scheduling becomes very important from the user's point of view as they have to pay based on usage of resources based upon time. There are different effective resource scheduling criteria which reduces execution cost, time, energy and increases CPU utilization and productivity. A broad classification can be done into the following categories: static, dynamic, preemptive, non-preemptive, centralized and decentralized scheduling [7]. The major performance metrics used in the literature are as follows [8–9]:

- The *Makespan* is the maximum finishing among all the received tasks.

$$\text{Makespan} = \text{Max} \{ \text{FT}_i \mid \forall_i \in I \} \quad (1)$$

- The *Throughput* is the number of tasks completed with respect to deadline of each job.

$$\text{Throughput} = \sum_{i \in I} X_i \quad (2)$$

- The *Response Time* is the time at which task arrives in the system to the time task is scheduled first time for execution.

$$\text{Response Time} = T_{\text{first execution}} - T_{\text{arrival}} \quad (3)$$

- The *Transmission Time* is time required to transfer a task from queue to a specific VM.
- The *waiting time* is defined as the time consumed in the waiting queue before the start of execution of particular task.
- The *Total Cost* depends on transfer of file and processing time.

$$\text{TotalCost} = P_i \times P_c + \left\{ \sum_{f \in \text{FIN}_i} \text{Size}(f) + \sum_{f \in \text{FOUT}_i} \text{Size}(f) \right\} \times \text{PTPB} \quad (4)$$

where P_C processing cost, f is file, PTPB processing time per bytes.

3 Optimization Techniques

The performance of a system is directly influenced by the efficiency of task execution schedule. To achieve this, a number of optimization algorithms for allocating and scheduling the resources proficiently in the cloud have been proposed over the years. A comparative study of different meta-heuristic techniques is presented here that perform efficient task scheduling is given below:

3.1 Genetic Algorithm (GA)

GA is inspired from the biological idea of creating a new generation population. Like in the Darwin's theory of Natural Selection, the term "Survival of the fittest" is employed as the strategy method for task scheduling as the tasks are assigned to resources according to the value of fitness function. The basic terminologies of the GA are: *Initial Population* (all solution), *Fitness Function* (measures of fitness of all solution) *Selection* (choose fittest solution for next generation) *Crossover* (Intermixing of parts of two parents) *Mutation* (produce genetic diversity) [10–11].

3.2 *Harmony Search Algorithm (HS)*

HS is a meta-heuristic search algorithm inspired by the process of musicians searching for a perfect harmony [12]. The musician can create a new harmony using three rules: Playing exactly the same from memory; almost similar to known one after pitch adjustment; totally compose a new one. The major principles in the HS are: *Initialization* (parameters like Harmony Memory Size (HMS), Pitch Adjusting Rate (PAR) and Harmony Memory Considering Rate (HMCR), *Creation of Harmony Memory* (2-D matrix containing a set of possible solutions), *Improvise a new harmony* (create a new solution), *Randomization* (diversity of solution), *updating* (to get better harmony).

3.3 *Tabu Search (TS)*

TS is a meta-heuristic optimization search algorithm which uses a memory like HS. Tabu search was proposed by Glover [13]. It basically begins with a single random solution and is updated by one of the neighboring solutions. This process continues until the most optimal solution is found. It generates a neighborhood solution from the current solution and accepts a solution as the best solution if it is not improving the previous solution. This method can form a cycle by regenerating a previous solution again. Hence to avoid this cycle, TS discards the previous visited solution using memory called Tabu list.

3.4 *Particle Swarm Optimization (PSO)*

PSO has recently become an important heuristic approach and has been applied to various computationally hard and complex problems, such as task scheduling problem, extraction of knowledge in data mining, electrical power systems, etc. It draws inspiration from the social behavior of organisms like a bird flock or fish schooling. The main steps of the PSO are defined as [14]: Initial population (all possible solution, i.e., particles), Fitness Function, Selection (choosing best among two parameters personal best, i.e., p-best and global best, i.e., g-best), Updating (updates velocity and position of a particle).

3.5 *Cuckoo Optimization Algorithm (COA)*

COA is inspired from obligate brood parasitism of cuckoo which lays eggs in the host nest having Lévy flight behavior of the birds [15]. The main terminologies of

this algorithm are [12]: *Initialization* (population of solutions), *New Cuckoo Generation* (cuckoos, i.e., solutions are generated using levy flights), *Fitness Evaluation*, *Updating* and *Selection/Rejection*.

3.6 *Artificial Bee Colony (ABC)*

ABC is a swarm-based meta-heuristic optimization technique, inspiring from foraging conduct of honey bee colonies. ABC algorithm classifies the bees into three types: employed bees, scout bees and onlooker bees. The employed bees search for the food around the food source in their memory and this info of food sources is passed onto the onlooker bees. The onlooker bees do the selection procedure from the food sources found by the employed bees. The probability of selection of a food source by the onlooker bees is determined by its quality. The scout bees induct the diversity by abandoning their food sources and getting along in search of new ones. The total number of employed bees or the onlooker bees is the total number of solutions in the swarm [16]. The main phases of ABC Algorithm are: *Initialization* (all possible solutions), *Phase Employed Bee Phase* (determines the neighborhood food source), *Onlooker Bee Phase* (evaluate effectiveness of all food sources), *Scout Bee Phase* (new solutions are randomly discovered) and *Fitness value*.

3.7 *Ant Colony Optimization (ACO)*

ACO is a meta-heuristic method, introduced by Dorigo in 1992, inspired from food searching method of the ants. The ants share the food source information through pheromone path. An ant solves a problem by using a construction graph where edges are the possible partial solution that the ant can take according to a probabilistic state transition rule. After the selection of either a partial or a complete solution, the pheromone updating begins to start. This rule gives a mechanism for speeding up convergence and also prevents premature solution stagnation [17, 18].

3.8 *Simulated Annealing (SA)*

SA is an iterative meta-heuristic random search optimization technique for solving several nonlinear optimization problems. The name and motivation originate from annealing in metallurgy, describes a process of heating and controlled cooling of a material to improve the dimension of its crystals and diminish their defects. It was proposed as the metropolis algorithm and after that many variations were introduced later on. Simulated annealing is widely being used in task scheduling in cloud environment, machine -scheduling and vehicle routing etc. [19].

3.9 Bacteria Foraging Optimization Algorithm (BFO)

BFO includes three basic mechanisms: chemotaxis, reproduction, and elimination-dispersal. Chemotaxis helps the motion of E-coli cell by swaying and plunging with help of flagella. Reproduction: Only half part of population survives, and that bacterium degenerates into two identical ones, which are then positioned at the same location leaving the total bacteria population unaffected. Elimination and Dispersal: The chemotaxis is considered for local search and it increases rate of the convergence. Since bacteria can get stuck in local minima hence, the diversity of BFO is changed to disregard the chances of getting stuck in the local minima. The event of dispersion occurs after a particular number of reproduction processes. So, some bacteria are taken with probability P , to be killed and shifted to a different location within the environment [20].

3.10 Gravitational Search Algorithm (GSA)

GSA is an optimization technique method based on “Gravitational Law” [21]. This is basically a population-based multi-dimensional optimization algorithm where agents are called as objects and their performance can be calculated by their masses. The masses are the way of communication as the agents move toward heavier masses by gravitational force. The heavy masses correspond to good solutions and move slowly than lighter ones. Each agent (mass) has four characteristics: position, Active Gravitational Mass (AGM), inertial mass (IM), and Passive Gravitational Mass (PGM). The solution of the problem can be obtained by position, and its inertial masses and gravitational can be calculated using a fitness function.

3.11 Lion Optimization Algorithm (LOA)

LOA is a meta-heuristic algorithm taken inspiration from the lifestyle of the lion. The lion has two types of social organization: resident and nomad. Residents live in groups, called pride that includes one or more than one adult males, around five females and their cubs. The nomads move about sporadically either single or in pairs. A lion can switch lifestyle means nomads may become residents and vice-versa [22].

Hybrid Meta-heuristic approaches Every meta-heuristic algorithm comes with its share of pros and cons. Clubbing together a selected set of them to harness the advantages of each one can improve the efficiency. Several such hybrid approaches have been proposed in the literature, which have been discussed below.

3.12 The Harmony Tabu Search (THTS)

In this proposed method, TS and HS are combined to improve the results. TS is applied as the first step followed by the HS. At the beginning of the algorithm, TS is initialized with a tabu list that contains all the candidate solutions and generates initial solutions which are compared with the best candidate solution in the tabu list. Its better quality guarantees its inclusion into the tabu list. After this, HS is applied with the initialization of Harmony memory (HM) with the tabu list. A new solution is obtained from HM by improvising each component of the solution with harmony memory considering rate (HMCR) parameter and mutation of the solution by pitch adjusting rate (PAR) [23].

3.13 Cuckoo Harmony Search Algorithm (CHSA)

The CS is very efficient for local search with a single parameter. But it has a limitation that it takes a huge amount of time to obtain an optimal solution. Similarly, HS has a limitation too, its search execution completely depends upon the parameter values adjustments. When hybridization is applied, it is seen that it removes those limitations which affect the performance of CS and HS individually [12].

3.14 Harmony-Inspired Genetic Algorithm (HIGA)

This Hybrid algorithm is composed of the HS and GA to detect both local optima as well as global optima when scheduling is being done. The HIGA provides better results when a scenario arises where the best individual remains in the same state either in local optimal state or global optimal state after many generations with the help of HS and updates the current population in the GA. If HS failed to find it in many iterations, it simply means the best solution might be in the global optimal state. As a result, the process can halt. So, in spite of the halting process, the HIGA algorithm reduces the number of iterations and senses local or global optimal state every time. In this, GA is considered as a primary optimization algorithm and when local optimal solution is found by any individual then HS is used to evaluate global optimal solution [24].

3.15 Genetic Algorithm-Particle Swarm Optimization (GA-PSO)

Here GA is applied first and a random population is generated. Then fitness function is applied to obtain elites which are divided into two equal halves. First half part is employed by GA and rest of the half by PSO. In GA, the best elites are given to crossover operator and mutation operator, while in PSO p-best and g-best is calculated for each elite. The position and velocity of elites is calculated and updated in the each iteration [25].

3.16 Multi-objective Hybrid Bacteria Foraging Algorithm (MHBFA)

This algorithm produces a solution with a better local and global search capability and a greater convergence time. Since, Bacteria Foraging (BF) has a great local search capability and unluckily has a poor global search. GA overcomes this limitation, hence, the MHBFA inherits swarming, elimination and dispersal from BF and these are measures which are critical in global search procedure [26].

3.17 Simulated Annealing Based Symbiotic Organisms Search (SASOS)

This Hybrid algorithm is comprised of the SA and Symbiotic Organism Search (SOS) for achieving the improved convergence rate and improved quality of the solution. The SOS algorithm includes phases like mutualism, commensalism, and parasitism. The SA has a systematic ability to get better local search solutions using the policy of commensalism and mutualism phases of the SOS. The parasitism phase remains unaffected because it deletes the passive solutions and injects the active ones in the solution space which could help the search process out of the local optimal region [27].

3.18 The Technique for Order of Preference by Similarity to Ideal Solution-Particle Swarm Optimization (TOPSIS-PSO)

In this hybridization technique, PSO is combined with TOPSIS algorithm to find an optimal solution by taking into account criteria like, transmission time, execution

time, and cost, which is carried out in two phases. In the first phase, TOPSIS is applied in order to achieve the relative proximity of the jobs. In the second phase, PSO is applied for all tasks to compute closeness in these three criteria in all virtual machines (VM). The fitness function of PSO is formulated using TOPSIS which gives an optimal solution in minimum time [28].

3.19 Artificial Bee Colony Simulated Annealing (ABC-SA)

This Hybrid algorithm is comprised of ABC and Simulated SA for the efficient task scheduling depending upon their sizes, priority of request came etc. [29].

3.20 Genetic Algorithm Artificial Bee Colony (GA-ABC)

This Hybrid algorithm combines the features of GA and ABC with the facility of dynamic voltage and frequency scaling (DVFS) to achieve efficient task scheduling. In this algorithm, GA is used as the first step for starting the allocation process of tasks to VM and obtained the new individuals until the termination condition of GA occurs. The output of GA is fed as the input to the ABC. Then, ABC provides the optimal distance between tasks and VMs [30].

3.21 Cuckoo Gravitational Search Algorithm (CGSA)

This hybrid CGSA composed of CS and GSA. The major demerit of CS algorithm is that it takes maximum time in order to find the optimal solution and the disadvantage of GSA is that it does not converge well for local optimal solution. The CGSA uses the advantages of CS and GSA It conquers the weaknesses and provides the efficient solution in a shorter computational time [31].

3.22 Oppositional Lion Optimization Algorithm (OLOA)

This hybrid OLOA uses the benefits of Lion optimization algorithm (LOA) and oppositional based learning (OBL). In this hybrid approach, OBL is nested within the LOA [32].

3.23 *Fuzzy System—Modified Particle Swarm Optimization (FMPSO)*

PSO uses the Shortest Job to Fastest Processor (SJFP) technique to initiate the initial population, location matrix of particle and velocity matrix. The roulette wheel selection, crossover operator and mutation operator are considered to conquer the demerits of PSO like the local optima. The Hierarchical fuzzy system is used for the evaluation purpose of fitness value of each particle [33].

4 Literature Review

The related studies on this research area have been discussed in Table 1. Alazzam et al. [23] proposed a hybrid task scheduling algorithm which includes Tabu- Harmony search algorithm (THTS). The algorithm performs better in respect of makespan and cost compared to TS, HS, round-robin individually. Pradeep et al. [12] presented hybrid Cuckoo Harmony Search Algorithm (CHSA) for task scheduling to improve the energy consumption, memory usage, credit, cost, fitness function and penalty and it was observed that the performance of this proposed algorithm is comparatively better than individual CS and HS algorithm, and hybrid CSGA. Sharma and Garg [24] focused on a Harmony-Inspired Genetic Algorithm (HIGA) for energy-efficient task scheduling to improve energy efficiency and performance. The results describe that the presented algorithm improved efficiency and performance. Senthil Kumar et al. [25] discussed a hybrid Genetic Algorithm Particle Swarm Optimization (GA-PSO) to minimize the total execution cost. GA-PSO helped to obtain the result better than various existing algorithms like GA, Max-Min, Min-Min.

Srichandan et al. [26] discussed a Hybrid Bacteria Foraging Algorithm (HBFA) for task scheduling which inherits the desirable characteristics of GA and Bacteria foraging (BF) in cloud to minimize the makespan and reduce energy consumption economically as well as ecologically. The results show that HBFA outperforms than GA, PSO, BF when applied alone. Abdullahi and Ngadi [27] put forth a hybrid algorithm to optimize the task scheduling based on SA and SOS for improving convergence speed, response time, degree of imbalance and makespan. The results show that SASOS performs better than SOS. Panwar et al. [28] proposed a new hybrid algorithm based on TOPSIS and PSO to solve multiple objective such as transmission time, resource utilization, execution time, and cost. The achievement of TOPSIS-PSO has been compared with ABC, PSO, dynamic PSO (DPSO), FUGE and IABC algorithm in terms of transmission time, makespan, resource utilization and total cost.

Muthulakshmi and Somasundaram [29] proposed a hybrid algorithm which combines the advantages of ABC and SA to improve the makespan. The result obtained by using this algorithm outperforms than MFCFS, Shortest Job First (SJF), LJF, hybrid ABC-LJF and hybrid ABC-SJF. Kumar and Kalra [30] has presented a

Table 1 Literature survey summary

Authors	Strategy	Description	Performance metrics used	Achievements	Tools
Alazzam et al. [23]	THTS	Tabu search is applied as first step after that Harmony search is applied until optimal solution obtained	<ul style="list-style-type: none"> • Makespan • Throughput • Total cost 	<ul style="list-style-type: none"> • Least makespan • Least cost • Nearly the same throughput 	CloudSim
Sharma and Garg [24]	HIGA	Current generation is evolved by GA results in local optima and it is given to HS through which global optima is achieved	<ul style="list-style-type: none"> • Makespan • Energy consumption • Execution time 	<ul style="list-style-type: none"> • Makespan improved by 47% • Energy saving 33% • Less execution time by 39% 	MATLAB 2013b
Senthil Kumar et al. [25]	GA-PSO	The population is randomly generated. The half population is evaluated GA and rest half is by PSO and combine both result to get optimal solution	<ul style="list-style-type: none"> • Response time 	<ul style="list-style-type: none"> • Response time is lowered by 1678, 1393, and 1000 ms with Min-Min, Max-Min and GA 	CloudSim
Al-Arasi and Saif [34]	HTSCC	GA is employed to randomly generated solution with tournament selection operator, If optimal solution is find then it is given to PSO to find best solution	<ul style="list-style-type: none"> • Makespan • Resourceutilization 	<ul style="list-style-type: none"> • Makespan 31.32% and 22.36% • Resource utilization 23.17% and 19.6% better than GA and PSO 	CloudSim

(continued)

Table 1 (continued)

Authors	Strategy	Description	Performance metrics used	Achievements	Tools
Jana and Poray [36]	Enhanced GA -PSO	Implemented with GA and obtained result is given to PSO to achieve best result	<ul style="list-style-type: none"> • Response time • Waiting time 	<ul style="list-style-type: none"> • Minimum waiting time • Minimum response time 	CloudSim
Pradeep and Jacob [31]	CHSA	Local optimal solution with cuckoo search and then it is given to Harmony Search	<ul style="list-style-type: none"> • Memory usage • Cost • Energy consumption • Fitness • Penalty • Credit 	<ul style="list-style-type: none"> • Memory usage 156 • Cost of 0.0098\$ • Energy consumption is 0.23 • High Fitness profit • Penalty is 0.276 • Credit is 0.724 	CloudSim
Kousalya and Radhakrishnan [35]	Improved GA-PSO	Group of PSO is ordered and coefficient of every constraint of optimal solution of GA is evaluated. Best solution is obtained if anyone meets termination criteria	<ul style="list-style-type: none"> • Cost • Execution time 	<ul style="list-style-type: none"> • Least cost • Least execution time 	CloudSim
Srichandan et al. [26]	MHBFA	Initialized with BF and Hybrid chemotaxis and Hybrid reproduction is employed by PSO to obtain optimal solution	<ul style="list-style-type: none"> • Makespan • Energy consumption • Convergence • Stability • Solution diversity 	<ul style="list-style-type: none"> • Makespan decreases • Minimum energy consumption than GA, PSO and BFA • Scalable • High coverage ratio 	MATLAB 2013b
Abdullahi and Ngadi [27]	SASOS	SOS is initialized first and SA techniques is applied during mutualism and commensalism phase of SOS	<ul style="list-style-type: none"> • Convergence speed • Response time • Makespan 	<ul style="list-style-type: none"> • Faster convergence for 500 tasks • Improved response time • Least makespan 	CloudSim

(continued)

Table 1 (continued)

Authors	Strategy	Description	Performance metrics used	Achievements	Tools
Panwar et al. [28]	TOPSIS –PSO	Initialized with PSO in which fitness is calculated using TOPSIS	<ul style="list-style-type: none"> • Makespan • Transmission time • Cost • Resource utilization 	<ul style="list-style-type: none"> • Least makespan • Minimum transmission time • Cost • Improved resource utilization 	CloudSim
Kumar and Kalra [30]	GA-ABC	Started with GA and when termination occurred then result is given as input ABC followed by initialization of DVFS	<ul style="list-style-type: none"> • Makespan • Energy consumption 	<ul style="list-style-type: none"> • Less makespan by 75.5% for 40 tasks • Less energy consumption by 84.14% for 40 tasks than Modified GA 	CloudSim
Muthalakshmi et al. (2017)	ABC-SA	Initialized with ABC having random selection capability of SA to increase efficiency	<ul style="list-style-type: none"> • Task size • Priority of the request 	<ul style="list-style-type: none"> • Improved task size and priority of request 	CloudSim
Pradeep and Jacob [31]	OCSA	Initialized with OBL and fittest solution is selected and updating is done by CS	<ul style="list-style-type: none"> • Makespan • Cost 	<ul style="list-style-type: none"> • Makespan is 141.5 for 500 tasks • cost is 110.3 for 500 tasks 	CloudSim
Gan et al. [37]	GSAA	Population employed with GA. After mutation phase the result is given to SA	<ul style="list-style-type: none"> • Bandwidth • Completion time • Cost • Reliability 	<ul style="list-style-type: none"> • Converges in 743 gen • Function-value 0.91174 • Optimal Scheduling 	MapReduce

(continued)

Table 1 (continued)

Authors	Strategy	Description	Performance metrics used	Achievements	Tools
Jiang et al. [38]	HSSA	Employed SA to population generated using partial HS	<ul style="list-style-type: none"> • Completion time 	<ul style="list-style-type: none"> • Least Completion time 	VC++6.0
Tawfeek and Elhady [39]	Hybrid swarm intelligence techniques	ABC is first initialized and the solution is handled by different suitable module like Bees(), Ants(), particles () etc.	<ul style="list-style-type: none"> • Makespan 	<ul style="list-style-type: none"> • Least makespan than ABC, PSO, ACO 	CloudSim
Mansouri et al. [40]	FMPSO	Initialized with PSO in which fitness of solution is calculated using fuzzy inference system	<ul style="list-style-type: none"> • Makespan • Improvement ratio • Efficiency • Execution Time (ET) 	<ul style="list-style-type: none"> • Makespan by 13% in comparison with FUGE • Reduce ET by 8% and 16% • Average efficiency 3.36 	CloudSim
Azad and Navimipour [41]	HCACO	Initialization is done with ACO and the obtained local result is given to cultural Algorithm	<ul style="list-style-type: none"> • Makespan • Energy conservation 	<ul style="list-style-type: none"> • Completion time 106.48 • Energy consumption 0.204 	C# in cloud azure
Li and Han [42]	ABC-HFS	Employed with ABC having 2 kinds; 1.HFS with identical parallel machines and 2.HFS with unrelated machines	<ul style="list-style-type: none"> • Completion time 	<ul style="list-style-type: none"> • Reduced completion time 	C++
Pradeep and Jacob [32]	OLOA	LOA is implemented with initialization of population based on OBL	<ul style="list-style-type: none"> • Makespan • Cost 	<ul style="list-style-type: none"> • Makespan 95.2 s • Cost is 65.2 s 	CloudSim
Mamikandan and Pravin [43]	LGSA	Implemented with LOA and fitness function is evaluated using GSA	<ul style="list-style-type: none"> • Profit • Cost • Energy 	<ul style="list-style-type: none"> • Max profit 0.8 • Min cost 0.011\$ • Min energy 0.039 	CloudSim

(continued)

Table 1 (continued)

Authors	Strategy	Description	Performance metrics used	Achievements	Tools
Alla et al. [33]	Dynamic Queue Meta-heuristic Algorithm	The FL-PSO algorithm and SA-PSO are applied to get the optimal solution	<ul style="list-style-type: none"> • Waiting time • Makespan • Cost • Resource utilization 	FL-PSO gives the improved results of <ul style="list-style-type: none"> • Waiting time • Makespan • Cost • Resource utilization 	CloudSim
Gabi [44]	CSM-SA	CSM is used as a first step then SA is implemented as second step	<ul style="list-style-type: none"> • Time • Cost 	Improved <ul style="list-style-type: none"> • Time • Cost 	CloudSim

hybrid algorithm GA-ABC to make improvement in makespan and energy consumption using DVFS. DVFS model is used for the calculation of power consumption. The results show better results than Modified Genetic Algorithm (MGA). Pradeep and Jacob [31] discussed a hybrid algorithm which inherits the benefits of both Cuckoo Search (CS) and Gravitational Search (GS) to execute the tasks with low cost, less usage of resources, and minimum energy-consumption. The results show that CGSA perform better than CS, GSA, GA, PSO. Krishnadoss and Jacob [32] presented a hybrid algorithm that uses LOA and Oppositional Based Learning (OBL) to improve makespan and cost. The OLOA performs better than PSO and GA. Alla et al. [33] proposed two hybrid algorithms using Fuzzy Logic with PSO and SA with PSO for optimization of makespan, waiting time, cost, resource utilization, degree of imbalance and queue length of the tasks in cloud environment. The hybrid algorithm outstrips the individual SA and PSO in their performance.

Al-Arasi and Saif [34] presented hybrid algorithm that inherits the advantages of GA with Tournament selection and PSO. The GA-PSO provides better results by reducing makespan and increasing the resource utilization. Kousalya and Radhakrishnan [35] implemented a hybrid algorithm that uses improved GA including divisible task scheduling into the foreground and background process and PSO. The GA-PSO performs better in terms of execution time and resource utilization. Jana and Poray [36] presented a hybrid GA-PSO algorithm to provide comparatively better response time and minimize the waiting time. The results show that this cost-effective GA-PSO achieves better response time, and minimizes the waiting time. Gan et al. [37] discussed about hybrid algorithm using GA and SA which considers the Quality of Service (QOS) requirements for many types of tasks, that correspond to the user's tasks-characteristics in cloud—computing environment. Jiang et al. [38] focused on hybridization using merits of HS and SA which provides global search and faster convergence speed and local minima escaping to get the better solutions. Tawfeek and Elhady [39] proposed a hybrid swarm intelligence technique which involves ABC, PSO, ACO. The algorithm performs better than existing algorithms.

Mansouri et al. [40] presented a hybrid algorithm FMPSO to determine the execution time, makespan, imbalance degree, improvement ratio and efficiency. The results show that it does better than other strategies like FUGE, SGA, MGA etc. Azad and Navimipour [41] discussed a hybrid algorithm based on Cultural Algorithm which considers acceptance and influence as major operators and the Ant Colony Optimization Algorithm minimizes the makespan and energy consumption. The results show that it performs better than HEFT and ACO. Li and Han [42] focused on a hybrid task scheduling technique with ABC algorithm with flow shop scheduling for improvement of convergence rate. Manikandan and Pravin [43] proposed a hybrid algorithm uses the benefits of LOA and GSA for the multi-objective task scheduling and uses profit, cost, and energy as the performance metrics. The LGSA perform better than the others. Gabi [44] presented a hybrid multi-objective algorithm comprised of Cat Swarm optimization (CSO) and SA for task scheduling. The algorithm outperformed its constituents by resulting in minimum execution time, cost and a greater scalability which provides global search and faster convergence speed and local minima escaping to get the better solutions.

5 Comparison of Performance Metrics

The selection of appropriate performance evaluation metrics is also important in determining the efficiency of a scheduling algorithm. There have been numerous metrics devised over the years to capture the overall efficiency of the algorithm. Achieving that with a single metric is not possible, making the use of multiple metrics for the evaluation of an algorithm a common trend in the literature. Figure 1 is the graphical depiction of Table 1, i.e., the number of metrics used by several authors in the literature. The most commonly used metric in the literature is the makespan which can be seen in the Fig. 2.

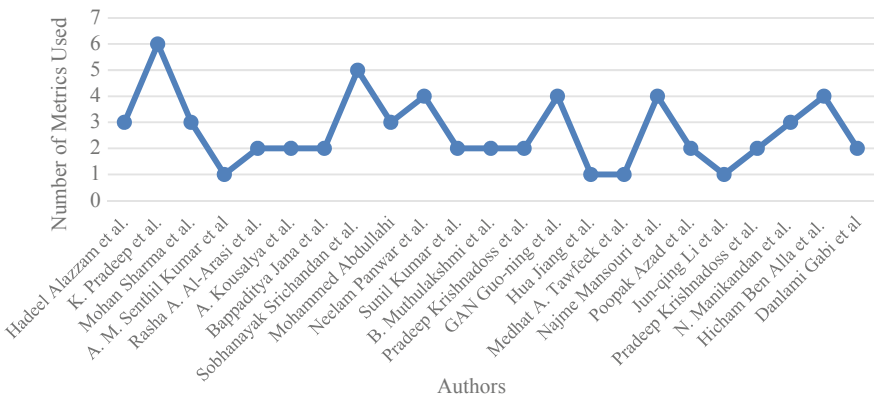


Fig. 1 Comparison on the basis of metrics used

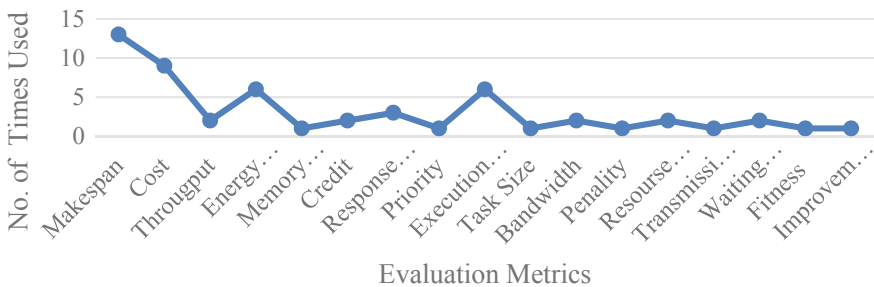


Fig. 2 Comparison of use of different evaluation metrics

6 Conclusion

The applications of the cloud computing environment have been spiking up since the past couple of decades. With more and more services and applications being shifting to the cloud, the requirement of developing more efficient and faster-driving algorithms viz. task scheduling, resource scheduling algorithms is also growing. Finding an appropriate cost-effective, efficient and competent scheduling algorithm is a tedious task. The scheduling algorithms used in conventional computing systems fail to perform well in a more constrained cloud environment. Relatively new techniques like LOA and ACO in hybrid form have shown promising results by outperforming the others. The performance evaluation metrics do not capture the comprehensive efficiency of the scheduling algorithm. The most widely used metric is the makespan but lately, there has been a shift toward energy-efficient algorithms increasing the use of energy efficiency metric for performance evaluation. All the studies in the literature have used the basic versions of the individual algorithms in the process of hybridization. In the future, the hybridization can be done with the improved variants of these algorithms like improved harmony search, modified PSO, etc. to eliminate the implicit limitations of the basic variants.

Though there are numerous standard data sets available that replicate the active cloud scenario but the research needs to be extended to the dynamic scheduling techniques, making it an open research field for the researchers in the future. So far, Meta-heuristics have been performing altogether quite efficiently but as they draw inspiration from many natural or man-made phenomenon making it susceptible to diverging away from the scientific consistency.

References

1. Porres I, Mikkonen T, Ashraf A (2013) Developing cloud software: algorithms, applications, and tools. Turku Centre for Computer Science, Finland
2. Buyya R (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener Comput Syst* 25(6):599–616
3. Kamboj S, Ghumman NS (2016) A survey on cloud computing and its types. In: 3rd International conference on computing for sustainable global development (INDIACom), IEEE, pp 2971–2974
4. Kumar R, Sahoo G (2014) Cloud computing simulation using CloudSim. arXiv preprint [arXiv:1403.3253](https://arxiv.org/abs/1403.3253). <https://doi.org/10.14445/22315381/IJETT-V8P216>
5. Singh S, Chana I (2016) A survey on resource scheduling in cloud computing: issues and challenges. *J Grid Comput* 14(2):217–264
6. Singh P, Dutta M, Aggarwal N (2017) A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowl Inf Syst* 52(1):1–51
7. Thomas A, Krishnalal G, Raj VPJ (2015) Credit based scheduling algorithm in cloud computing environment. *Procedia Comput Sci* 46:913–920
8. Mohialdeen IA (2013) Comparative study of scheduling algorithms in cloud computing environment. *J Comput Sci* 9(2):252–263

9. Raju R, Babukarthik RG, Chandramohan D, Dhavachelvan P, Vengattaraman T (2013) Minimizing the makespan using hybrid algorithm for cloud computing. In: 3rd IEEE International Advance Computing Conference (IACC), IEEE, pp 957–962
10. Ge Y, Wei G (2010) GA-based task scheduler for the cloud computing systems. In: International conference on web information systems and mining, vol 2, IEEE, pp 181–186
11. Moraga RJ, DePuy GW, Whitehouse GE (2006) Meta heuristics: a solution methodology for optimization problems. In: Handbook of industrial and systems engineering, CRC Press, Florida
12. Pradeep K, Jacob TP (2018) A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Pers Commun* 101(4):2287–2311
13. Glover F (1997) Tabu search and adaptive memory programming—advances, applications and challenges. In: Interfaces in computer science and operations research. Springer, Boston, pp 1–75
14. Bilgaiyan S, Sagnika S, Das M (2014) An analysis of task scheduling in cloud computing using evolutionary and swarm-based algorithms. *Int J Comput Appl* 89(2):11–18
15. Yang X-S, Deb S (2009) Cuckoo search via Levy flights. In: Proceedings of the world congress on nature and biologically inspired computing, pp 210–214
16. TSai PW, Pan JS, Liao BY, Chu SC (2009) Enhanced artificial bee colony optimization. *Int J Innovative Comput Inf Control* 5:1–12
17. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
18. Chiang CW, Lee YC, Lee CN, Chou TY (2006) Ant colony optimization for task matching and scheduling. *IEEE Proc—Comput Digital Tech* 153(6):373–380
19. Liu X, Liu J (2016) A task scheduling based on simulated annealing algorithm in cloud computing. *Int J Hybrid Inf Technol* 9(6):403–412
20. Chen H, Zhu Y, Hu K (2011) Adaptive bacterial foraging optimization. *Abstr Appl Anal* 2011 (Hindawi)
21. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
22. Yazdani M, Jolai F (2015) Lion optimization algorithm. *J Comput Des Eng*
23. Alazzam H, Alhenawi E, Al-Sayyed R (2019) A hybrid job scheduling algorithm based on Tabu and Harmony search algorithms. *J Supercomputing* 75(12):7994–8011
24. Sharma M, Garg R (2019) HIGA: harmony-inspired genetic algorithm for rack-aware energy-efficient task scheduling in cloud data centers. *Eng Sci Technol Int J*
25. Senthil Kumar AM, Parthiban K, Siva Shankar S (2019) An efficient task scheduling in a cloud computing environment using hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) algorithm. In: International Conference on Intelligent Sustainable Systems (ICISS), IEEE
26. Srichandan S, Kumar TA, Bibhudatta S (2018) Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Comput Inform J* 3(2):210–230
27. Abdullahi M, Ngadi MA (2016) Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. *PloS ONE* 11(6):e0158229
28. Panwar N, Negi S, Rauthan MMS, Vaisla KS (2019) TOPSIS-PSO inspired non-preemptive tasks scheduling algorithm in cloud environment. *Cluster Comput* 22(4):1379–1396
29. Muthulakshmi B, Somasundaram K (2019) A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment. *Cluster Comput* 22(5):10769–10777
30. Kumar S, Kalra M (2019) A hybrid approach for energy-efficient task scheduling in cloud. In: Proceedings of 2nd international conference on communication, computing and networking. Springer, Singapore
31. Pradeep K, Jacob TP (2018) CGSA scheduler: a multi-objective-based hybrid approach for task scheduling in cloud environment. *Inf Secur J—A Global Perspect* 27(2):77–91
32. Krishnadoss P, Jacob P (2019) OLOA: based task scheduling in heterogeneous clouds. *Int J Intell Eng Syst* 12.1
33. Alla HB, Alla SB, Touhafi A, Ezzati A (2018) A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment. *Cluster Comput* 21(4):1797–1820

34. Al-Arasi RA, Saif A (2018) HTSCC a hybrid task scheduling algorithm in cloud computing environment. *Int J Comput Technol* 17(2)
35. Kousalya A, Radhakrishnan R (2017) Hybrid algorithm based on genetic algorithm and PSO for task scheduling in cloud computing environment. *Int J Networking Virtual Organ* 17(2–3):149–157
36. Jana B, Poray J (2018) A hybrid task scheduling approach based on genetic algorithm and particle swarm optimization technique in cloud environment. In: *Intelligent engineering informatics*. Springer, Singapore, pp 607–614
37. Gan G, Huang T, Gao S (2010) Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. In: *International conference on intelligent computing and integrated systems*. IEEE
38. Jiang H, Bao Y, Zheng L, Liu Y (2012) A hybrid algorithm of harmony search and simulated annealing for multiprocessor task scheduling. In: *International Conference on Systems and Informatics (ICSAI)*, IEEE, pp 718–720
39. Tawfeek MA, Elhady GF (2016) Hybrid algorithm based on swarm intelligence techniques for dynamic tasks scheduling in cloud computing. *Int J Intell Syst Appl* 8(11):61
40. Mansouri N, Zade BMH, Javidi MM (2019) Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. *Comput Ind Eng* 130:597–633
41. Azad P, Navimipour NJ (2017) An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm. *Int J Cloud Appl Comput* 7(4):20–40
42. 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 Comput* 1–17
43. Manikandan N, Pravin A (2019) LGSA: hybrid task scheduling in multi objective functionality in cloud computing environment. *3D Res* 10(2):12
44. Gabi D (2020) Hybrid cat swarm optimization and simulated annealing for dynamic task scheduling on cloud computing environment. *J Inf Commun Technol* 17(3):435–467