A Review on Meta-heuristic Independent Task Scheduling Algorithms in Cloud Computing

Anup Gade, M. Nirupama Bhat, and Nita Thakare

1 Introduction

Cloud computing has proven itself as one of the most demanded, acclaimed and user friendly technology in recent time. Cloud computing unifies distributed computing, high performance computing, soft computing, parallel and grid computing. This gained huge amount of popularity due its ease of use model, versatility for both service provider and end users. The vibrant use, popularity and challenges of this concept have attracted the attention of scientific community. Virtualization is a key concept responsible for its growth and success. By implementing the concept of virtualization even a single computer can have more than one operating system which can share its resources like memory, storage and CPU portioning. All mentioned parameters made the concept of virtualization materialized. Mapping physical infrastructure to dynamically changing organizational requirement is a key success of virtualization. While discussing cloud computing there are numerous issues which needs serious attention and research as a matter of consideration reduction in the power consumption of data centre, efficient use of network bandwidth, improving quality of service, better resource management, security issues, better implementation of multi-tenancy approach and the most important effective implementation of task scheduling mechanism which is ultimately responsible for processor utilization [1].

As cloud computing is mainly designed for multiple users and on-demand service mode is a basic feature hence good scheduling algorithm played a vital role in the successful implementation of system. A good scheduling algorithm has objectives

A. Gade (✉) · M. Nirupama Bhat
VFSTR Deemed to Be University, Guntur, Andhra Pradesh, India

N. Thakare
Priyadarshini College of Engineering, Nagpur, Maharashtra, India

S. Smys et al. (eds.), *New Trends in Computational Vision and Bio-inspired Computing*, https://doi.org/10.1007/978-3-030-41862-5_118

of efficient resource utilization, minimization of task completion time, reduced waiting time, improved quality of service and profit maximization. Task scheduling in cloud computing is treated as a one of the most tedious task and hence the best solution is difficult to propose. Researchers have proposed optimal solutions for task scheduling as it is difficult to solve it in a deterministic time. It is generally referred as NP-hard problem whose solution is achieved in polynomial time. Task scheduling in cloud computing can be roughly categorized into two types first is dependant task scheduling or workflow scheduling in which tasks are divided into sub-tasks. The sequence of execution of sub-task is a matter of concern here. The other is independent task scheduling in which all tasks are independent of each other and there is nothing like sequencing or precedence that needs to take care of. Task scheduling is responsible for providing better system performance and it is possible through implementation of appropriate and suitable scheduling strategy. Meta-heuristic algorithms have gained popularity in recent years to solve scheduling problems [2]. The most common approaches are Genetic Algorithms (GA) in which authors are correlated biological genetic algorithm to computing machines, Ant Colony Optimization (ACO) applied the analogies of real ants who searches shortest path between their colonies and victuals, Particle Swarm Optimization (PSO) inspired by communal behaviour of particles, BAT algorithm motivated from echolocation behaviour of bats, League Championship Algorithm (LCA) taken from sports league, Pareto Optimization, Cuckoo search algorithm and Lion optimization algorithm (Fig. 1).

The next sections of paper describes accordingly, Sect. 2 focuses on escalation parameters, Sects. 3–5 surveys based on comparison of existing literature in GA, ACO, PSO respectively, Sect. 6 discussed about LCA and Pareto optimization. The

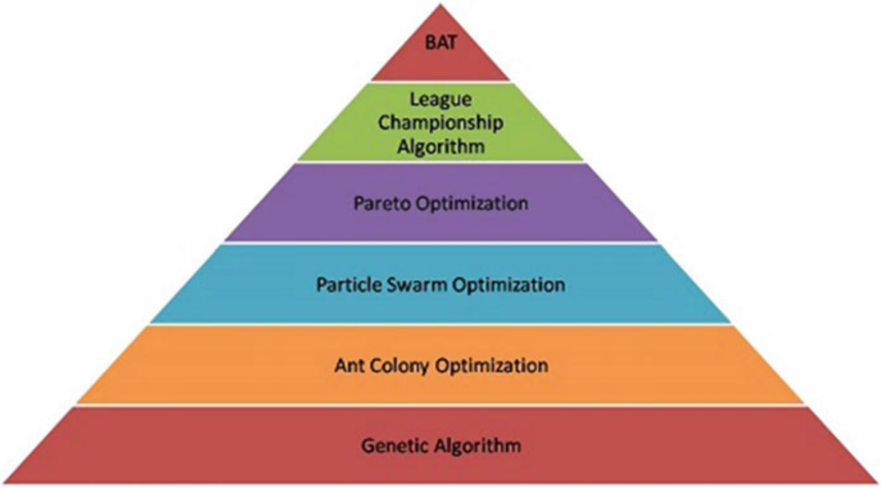

Fig. 1 Types of meta-heuristic task scheduling algorithms in cloud computing

observations are presented in Sect. 7, and conclusion is given in Sect. 8 of this article.

2 Escalation (Optimization) Parameters

Task scheduling in cloud computing environment has primarily two main concerns while focusing on it deeply. The user's view and service provider's view [2]. The factors of users concerned are minimization of makespan, reduction in completion time, minimum response time, meeting deadline, security issues, economic services and reasonable CPU utilization and cloud service providers are generally worried about how to increase resource utilization? which in turn increase profit maximization, increased throughput, minimization in power consumption, load balancing and maintaining quality of service. Figure 2 summarizes the motives of scheduling in the view of users and service providers.

2.1 User's View

(a) **Makespan:** Makespan indicates the time at which last task finishes it execution. Reducing makespan is primary responsibility of any task scheduling algorithm.

$$Makespan = \max\ \{F_i\}\ where\ F_i\ denotes\ the\ finishing\ time\ of\ task\ i \tag{1}$$

Table 1 shows the example calculating the makespan in case of scheduling of ten independent tasks among two resources.

Here in this example makespan is 25 s as last task completed execution at 25 s.

(b) **Completion time:** For an individual task completion time is addition of execution time and waiting time of a task.

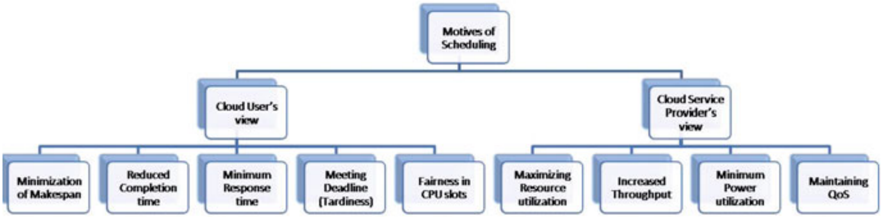

Fig. 2 Motives of scheduling in cloud computing

Table 1 A scheduling snapshot

Resource 1	T1	T3	T4	T6	T8	T10
Time	0–4	9	13	19	21	25
Resource 2	T2	T5	T7	T9		
Time	0–5	9	12	16		

$$Completion\ time\ (C_i) = E_i + W_i\ Where, \tag{2}$$

E_i is execution time of task i and
W_i is waiting time of task i

Here Completion time for task T5 is 9 s by assuming that all tasks are arrived at 0 s. A good scheduling algorithm should have minimum task completion time.

(c) **Response time:** It is the difference between the submission time of task and first response received by a task. Response time should also be as less as possible.

(d) **Deadline:** For a task scheduling algorithms deadline for a task should tend to zero as it is the difference between expected finishing time of a task and actual time when task has finished off.

(e) **Fairness:** To avoid starvation in scheduling it is necessary that each task must get fair share of CPU. Round Robin is an example in which fair CPU slots can ensured. SJF and Priority are the strategies where fairness in CPU allocation is usually bypassed.

2.2 Service Provider's View

(a) **Resource utilization:** One of the major issues for service provider is how to keep all resource as busy as possible as it directly proportionate the profit. A competent task scheduling algorithm uses all the resources to its fullest extent. Resource utilization is time taken by resources to finish all tasks by makespan of the task and number of resources involved. According to our example in Fig. 3 Resource utilization = (25 + 16)/50 = 0.82

(b) **Throughput:** It is number of tasks completed execution per time unit. Maximizing throughput is an essential criterion for efficient resource utilization and hence it carries reasonable importance in the view of service providers.

(c) **Energy efficiency:** A study reveals that power consumption in data centres is one of the most serious and costly affair in service providers point of view. Lot of research work is going on to minimize power consumption. Development of green cloud and thereby reducing the cost applied on it is main motive.

(d) **Quality of Service (QoS):** Maintaining the QoS is the most important responsibility of service provider and fulfilling all the clauses of Service Level Agreement is a key responsibility. Service provider always strived hard to have an effective scheduling algorithm which met all its QoS requirements.

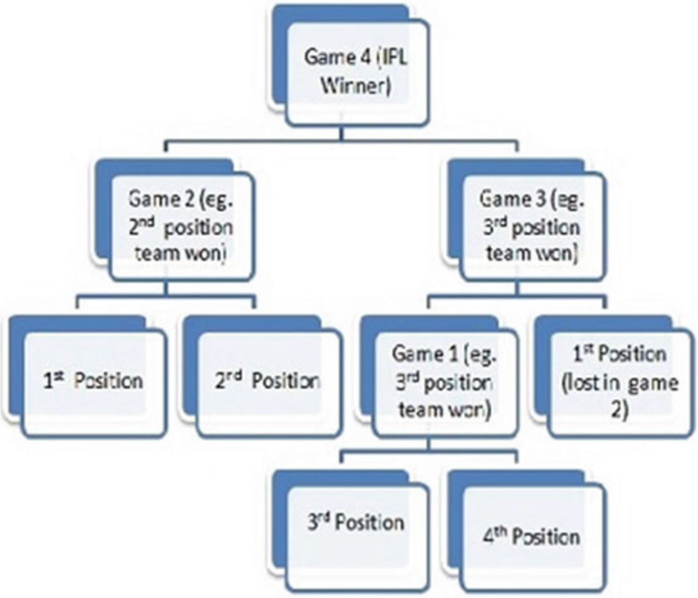
Game 4 (IPL Winner)
Game 2 (eg. 2nd position team won)
Game 3 (eg. 3rd position team won)
1st Position
2nd Position
Game 1 (eg. 3rd position team won)
1st Position (lost in game 2)
3rd Position
4th Position

Fig. 3 Example of LCA showing IPL taxonomy

All above discussed scheduling issues are achieved through effective implementation of various meta-heuristic algorithms as GA, ACO, PSO, BAT, LCA, etc.

3 Genetic Algorithm (GA) for Task Scheduling

GA was applied on the principles of Darwin popularly known as fittest of the survival. The heuristics used to generate optimal solution to computing problem. All possible solutions to a problem are represented as chromosomes in GA. At the initial state of the algorithm these chromosomes are selected as random solution as each chromosome represents the potential solution out of which chromosomes having highest fitness value will be selected for further operations like crossover and mutation [3]. The algorithm will have repeated iterations unless best possible solution is not generated out of the existing chromosomes.

Jena and Mohanty [4] applied GA to get the best task-VM pair so that makespan can be reduced and throughput can be increased. Scheduling is performed using shortest job first policy instead of routine FCFS. Article applied the concept of Advance Reservation (AR) and Best Effort (BE) through which resources can be reserved. AR category will have high priority than BE tasks. AR tasks are of non-pre-emptive nature whereas BE tasks may have to stop its execution if any AR task is arrived i.e. BE tasks are pre-emptive in nature. Task with high priority should be of type AR to stick the deadline of a task. Algorithm focuses on reducing waiting time as its increment will drop-down the customer satisfaction ratio.

Kaur and Verma [5] used GA for minimizing execution time and cost of a task. Authors have concentrated over usually neglected parameters including computational intrication, length of task, energy required, cost factors included for computation. Authors implemented modified form of genetic algorithm (MGA) by using Longest Task to Fastest Processor (LTFP) and Shortest Task to Fastest Processor (STFP) for creation of initial population of their MGA. Algorithm proved its betterment over regular genetic algorithm on the other hand it's a kind limitation as result is compared with only one approach.

Ge and Wei [6] implemented task scheduling for Hadoop MapReduce, for getting improved results they have implemented genetic algorithm to achieve enhanced load balancing in available nodes. The notable contribution is predictable execution time of task assigned to some of the processing element but on the contrary the success of implementation of GA using this method depends greatly on preciseness of predicated execution time of task. The matter of concerned here is time needed for GA to complete its effective calculations.

4 Ant Colony Optimization (ACO) for Task Scheduling

Algorithm is inspired by the behaviour of real ant species. These species are living in a group it has named as 'ant colony' while an individual ant start moving for searching its food leaves kind of chemical referred as pheromone on its entire path which other ants can be used as a path of guidance to reach the food source. Every ant is following the same process and the path which is actually shortest will have higher amount of pheromone as it was followed maximum number of ants. It was originally developed by Dorigo as his doctoral thesis in 1992 and named it as any system. Method has features of solving optimization problems like travelling salesman problem, knapsack problem, job shop problem to name a few. In the same way this approach is used to solve schedule issue in cloud and grid computing as it has nature of optimization only and mentioning precisely scheduling is used for optimizing multiple factors in cloud system. Initially path created by every individual ant is a candidate solution to the problem and the value set for pheromone goes on updating as more number of ants started using particular path. Path having best/highest pheromone value will be treated as shortest path.

Tawfeek and El-sisi [7] used ant colony optimization algorithm in order to achieve significant minimization in makespan as compared to traditional FCFS and RR algorithm. Authors have came-up with modification in initiation of pheromone, method to choose VM for next task and technique to update pheromone so as to achieve best probable solution. Factors like load balancing and workflow dependencies need to take into account for making it implementation ready.

Lu and Gu [8], proposed ACO algorithm for resolving scheduling issues in self-adaptive dynamic cloud environment. Authors primarily focused on solving the unpredictability of load occurrence in cloud environment. Cluster monitors will activate the hot spot if it reaches the threshold value set for CPU usage, memory and bandwidth resources. The major concerned authors have given towards is scheduling adaptive cloud using ACO algorithm and its ability to pass on the message of best probable solution by means of feedback and its ability to work on in simultaneous mode.

Mathiyalaga et al. [9] have modified ant colony algorithm to achieve better results than that of existing ACO algorithm. They have proposed modified form of updating the pheromone through which the algorithm achieved better results in term of time taken of execution. Minor issues have given fair amount of emphasis while designing algorithm like evaporation rate of pheromone, trail intensity of edge, etc. Algorithm is compared with existing ant colony optimization algorithm only to prove the novelty and superiority of the algorithm it requires the comparison of its results with two or three different heuristic algorithm.

5 Particle Swarm Optimization (PSO) for Task Scheduling

PSO is one of the popular meta-heuristic techniques used for finding optimality in distributed atmosphere. It was designed and developed by Kennedy and Eberhart [10] Method has gained popularity due to its simplicity and effectively in getting solution. It considers all possible solutions as a particles, each particle is assigned with position and velocity. Particles move through its available search space in order to get best possible solution, each particle will have local best solution ($L_{b).}$ From all available particles, particle having best position and velocity will be referred as global best solution ($G_{b).}$ In all iterations every individual particle will try to get its best position and velocity and compared with the values of G_b, if the value of global best is smaller than local best then this local best will be the new global best solution. PSO has made it easier and economic because of which it has been applied in variety of applications. Technique uses local search method with global search so that it would be able to get the best particle of the globe.

Ramezani et al. [11] found PSO as a suitable method for load balancing in cloud environment typically for task-oriented approach. In their article they have given the concept of migrating overloaded task instead of completely overloaded VM. Proposed model has used PSO algorithm to migrate overloaded task so that it can reduced the time of balancing the load. It avoids transferring the load from one physical machine to another thereby it reduces the effective transfer time and optimization algorithm lowered down the execution time.

Pooranian et al. [12] proposed solution by clubbing PSO algorithm with gravitational emulation local search approach for solving task scheduling issue of independent tasks. According to authors PSO has done exceptional in global search but when it comes to local optima, PSO needs some external support, GELS has an ability cross local optima. PSO has been used to search an entire space and GELS has combined to get an improved upcoming population. Algorithm achieved minimized makespan and number of those tasks which are not able to maintain their calculated time of completion which is usually equals to makespan. Authors have not focused on other factors of scheduling like load balancing, computational cost.

Sindhu et al. [13] talked about PSO based load rebalancing within available resources. Authors applied PSO with smallest position value method for matching the task. The primary goal of designing algorithm is to handle heterogeneous task and the machines available to deal with it. Proposed algorithms was proved to be reliable than PSO-LR approach. Authors have not compared it with other heuristics like ACO, GA or ABC algorithms and parameters like response time, economy and security also need attention for running system smoothly.

6 League Championship and Pareto Optimization for Task Scheduling

League Championship Algorithm is a kind of technique used for sports tournament to find out the best playing team. It correlates with finding out best solution available out of the existing one. All competing teams are having equal probability of winning the tournament initially it can be termed as candidate solution for the optimization problem. It works on by competing two teams with each other initially which has resemblance of cross over operation in GA. The fittest one in competition will win and move ahead for next stage the process continues till best one will not found out. It depends upon individuals how to form a league sequence. The famous Indian Premier League (IPL) has a following combination to get the winner shown in Fig. 3 from the figure it is clear that out of the existing teams top four teams are shortlisted for final knockout round and they are shown as first position, second position, third position and fourth position team. The knockout round is performed as given below:

Step I: First game will be played between position 3 and position 4 of general round teams. Here for understanding purpose let's assumed that team 3 won the match. Team 3 proved to be the fittest one here in this case.

Step II: Second game will be played between position 1 and position 2 teams of general round. Four our example assumed that team 2 has won the game. Team 2 is proved to be fittest in this game.

Step III: Third game is played between winner of step I and looser of step II (top 2 teams of general round got two opportunity to prove its fitness). The winning team will contest to grab the trophy. In this case winner of round I and looser of round II have a chance to keep in the race of optimal solution by winning round III. Here in example assumed that team 3 won the game.

Step IV: Final i.e. fourth game is played between team 2 and team 3 for proving it be the fittest one and making it best optimal solution available.

Above explained IPL algorithm is an example of LCA, many researcher have applied LCA for minimizing makespan and proved it be better than FCFS and RR algorithm.

Pareto Optimization is another method used quite frequently and popularly for solving multi-objective problems. Whenever multiple solutions are existing in that case solution S will be chosen only if no other solution meet the criterion and better than the solution S. If another solution S′ is existing and it is better than any one objective still in that case solution S will be chosen stating that it is better in other objectives than S′. Along with this many other techniques are available which many not be of meta-heuristic type but performed well in heterogeneous independent task scheduling in cloud computing one of them is dynamic heterogeneous shortest job first scheduling depicted in [1]. Authors have focused on heterogeneous parameters in cloud such as variations in VMs available, different capacity PMs and its CPU involved in computation and distribution of workload on these resources for applying scheduling algorithm effectively. Dynamic heterogeneity achieved reduced makespan, utilization of power in economic mode and bringing up the performance of the system. Comparative analysis of various meta-heuristic tasks scheduling algorithm for independent tasks are given in Table 2.

7 Observations

According to the survey performed in subsequent sections and comparisons made in Table 1 following observations can be depicted

(a) Methodology for finding out fittest chromosome in GA, optimal path in ACO, best local value in PSO and critical method to get winning team in LCA is most important aspect which requires top priority attention as it has proved in many articles that task scheduling is a NP-hard problem.

(b) Hybridization while implementation of solution can help in improving the quality of solution. Advantage of being hybrid approach is limitation of one can be covered up by another approach which will lead proposed hybrid approach to better position.

(c) Majority of the research community focused on optimizing makespan in user's view category whereas in service provider's view it has been observed that researchers primarily targeting on load balancing. Figs. 4 and 5 shows the optimization parameters focused in user's view and service provider's view respectively as per our comparative analysis Table 2.

(d) Green cloud development is a need of time and energy consumed by data centers is quite huge. Increasing number of data centres need an energy efficient scheduling algorithm to reduce power usage and thereby contributing towards building green cloud.

(e) Quality of Service (QoS) is a soul of cloud computing meeting Service Level Agreement (SLA) to the fullest extent is a fundamental 'mantra' of running

Table 2 Comparative analysis of meta-heuristic independent task scheduling algorithms in cloud computing

			Escalation parameters											
			Users view							Service providers view				
Name of meta-heuristic algorithm	Reference	Implementation technique/simulation software used	Makespan	Completion time	Response time	Deadline	Fairness	Security issues	Economy	Resource utilization	Throughput	Energy efficiency	Quality of services	Load balancing
Genetic algorithm	[4]	Matlab R2014a	✓	–	–	–	–	–	–	–	–	–	✓	–
	[5]	JGAP	–	✓	–	–	–	–	✓	–	–	–	–	–
	[6]	Mapreduce	✓	–	–	–	–	–	–	–	–	–	–	✓
	[14]	Matlab	✓	–	–	–	–	–	–	–	–	–	–	✓
Ant colony optimization algorithm	[15]	CloudAnalyst	–	–	–	–	–	–	–	–	–	–	✓	✓
	[9]	GRIDSim	✓	✓	–	–	–	–	–	–	–	–	–	–
	[16]	Not mentioned	–	–	✓	–	–	–	–	–	–	–	–	✓
	[17]	Java	–	–	–	–	–	–	–	–	–	–	–	✓
	[7]	CloudSim	✓	–	–	–	–	–	–	–	–	–	–	–
	[18]	CloudSim	–	–	✓	–	–	–	–	–	✓	✓	✓	✓
	[19]	Grid simulation environment	✓	–	–	–	–	–	–	–	–	–	–	✓
	[20]	Grid simulation environment (Benchmark simulation)	✓	–	–	–	–	–	–	–	–	–	–	✓
	[21]	Matlab 7.9	✓	–	✓	–	–	–	–	–	–	–	✓	–
	[8]	Xen	–	–	–	–	–	–	–	✓	–	–	–	✓

Particle swarm optimization algorithm	[22]	Simulated grid environment	–	✓	–	–	–	–	–	✓	–	–	–	–
	[23]	CloudSim	–	–	✓	–	–	–	–	–	✓	✓	–	–
	[24]	CloudSim	✓	✓	–	–	–	–	–	✓	–	–	–	–
	[25]	VC++	✓	✓	–	–	–	–	–	–	–	–	–	–
	[26]	CloudSim	✓	–	–	–	–	–	–	✓	–	–	–	–
	[27]	Eucalyptus Cloud	✓	✓	–	–	–	–	✓	–	–	–	–	–
	[13]	Not mentioned (Heterogeneous environment)	✓	✓	–	–	–	–	–	✓	–	–	–	–
	[11]	CloudSim	–	✓	–	–	–	–	✓	–	–	✓	✓	✓
	[12]	Java	✓	–	–	✓	–	–	–	–	–	–	–	–
	[28]	GridSim	✓	✓	–	–	–	✓	✓	–	–	–	–	–
	[29]	CloudSim	–	✓	–	–	–	–	–	✓	–	✓	–	–

(continued)

Table 2 (continued)

Name of meta-heuristic algorithm	Reference	Implementation technique/simulation software used	Escalation parameters											
			Users view							Service providers view				
			Makespan	Completion time	Response time	Deadline	Fairness	Security issues	Economy	Resource utilization	Throughput	Energy efficiency	Quality of services	Load balancing
Artificial bee colony algorithm	[30]	Matlab R2013b	✓	✓	✓	–	–	–	–	–	–	–	–	–
	[31]	Matlab	–	–	✓	–	–	–	✓	✓	–	–	✓	✓
	[32]	Matlab	✓	✓	✓	–	–	–	–	✓	✓	–	–	✓
	[33]	CloudSim	✓	✓	✓	–	–	–	–	✓	–	–	–	✓
	[34]	C#	✓	✓	–	–	–	–	–	✓	–	–	–	–
	[35]	CloudSim	✓	✓	–	–	–	–	–	–	–	–	–	✓
	[36]	Not mentioned	✓	–	–	–	–	–	–	–	–	–	–	–
	[37]	CloudSim	–	–	–	–	–	–	✓	–	–	–	–	–
	[38]	Not mentioned	✓	–	–	–	–	–	–	–	–	–	✓	✓
Bat algorithm	[30]	CloudSim	✓	✓	–	✓	–	–	–	–	–	✓	–	–
	[31]	Not mentioned	✓	✓	–	–	–	–	–	–	–	–	–	–
Imperialist competitive algorithm	[39]	Not mentioned	✓	–	✓	–	–	–	–	–	–	–	–	–
	[40]	CloudSim	✓	–	–	–	✓	–	–	–	–	–	✓	–
	[41]	CloudSim	–	–	–	–	–	–	–	–	–	✓	✓	✓
	[42]	WorkflowSim	–	–	–	✓	–	–	✓	–	–	–	–	–
	[43]	C#.Net	–	–	✓	–	–	–	✓	–	–	–	✓	–

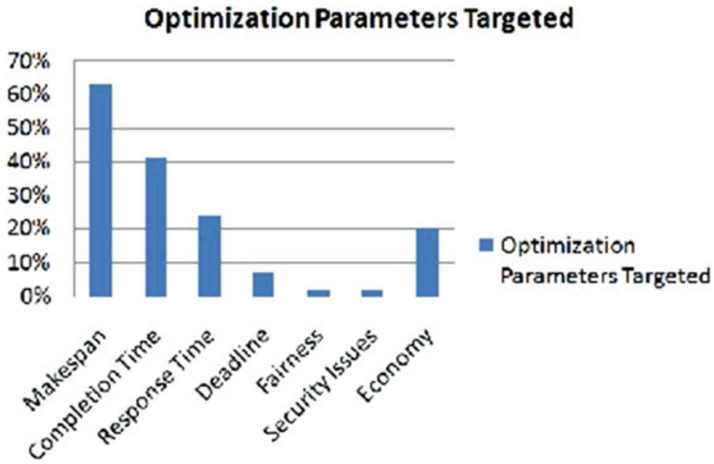

Fig. 4 Analysis of optimizations parameters targeted in user's view

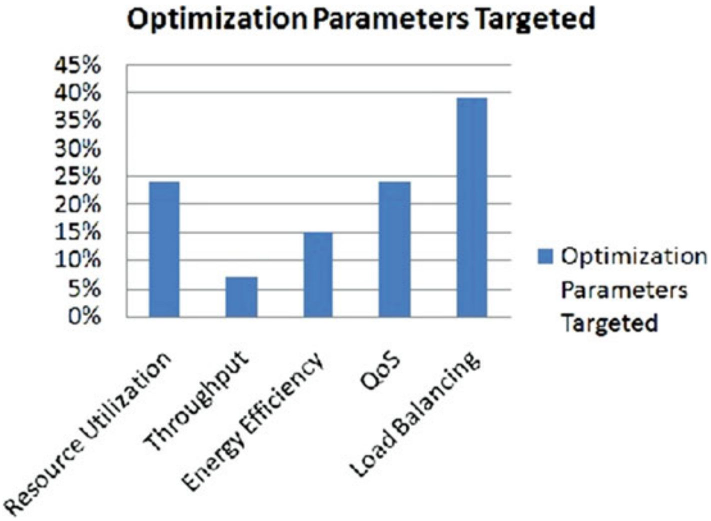

Fig. 5 Analysis of optimizations parameters targeted in service provider's view

cloud successfully. Customer satisfaction directly proportional to the QoS hence more attention is required, can be achieved through appropriately designed scheduling algorithm.

(f) Security is an open challenge for all the internet based application and particularly cloud computing is prone to internet threats. Though scheduling is not directly concerned with security but focusing towards security issues is a need of an hour.

8 Conclusion

This study describes the decisive relative analysis of diversified task scheduling algorithms in cloud/grid computing specially of meta-heuristic nature. The paper focuses particularly on tasks of independent nature while comparing. It has been observed that there are around nine factors of interest while scheduling task but only very few objectives are targeted by researchers' community. Other objectives of scheduling are equally important and more attention on these objectives need to be emphasized. Some noteworthy scale of enrichment in the performance of scheduling algorithms is expected as meta-heuristics algorithms are providing only optimal or sub-optimal solutions. Multiple observations are given in this paper and analyses of existing algorithms are also presented here. This article will be supportive for auxiliary study in the spectrum of independent task scheduling in cloud computing specially of meta-heuristics type.

References

1. Sonam Seth and Nipur Singh (2018) Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems, Int. J. Inf. Tecnol, Springer. https://doi.org/10.1007/s41870-018-0156-6
2. Poonam Singh, Maitreyee Dutaa and Naveen Agrawal (2017) A review of task scheduling based on meta-heuristics approach in cloud computing, Knowl Inf Syst (2017) Springer, https://doi.org/10.1007/s10115-017-1044-2
3. Mala Kalra and Sarbjeet Singh (2015) A review of metaheuristic scheduling techniques in cloud computing, Egyptian Informatics Journal, https://doi.org/10.1016/j.eij.2015.07.001
4. Tamanna Jena and J.R. Mohanty (2017) GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing, Arab J Sci. Eng. https://doi.org/10.1007/s13369-017-2766-x
5. Shaminder Kaur, Amandeep Verma (2012) An efficient approach to genetic algorithm for task scheduling in cloud computing environment, I.J. Information Technology and Computer Science, 2012, 10, 74-79, https://doi.org/10.5815/ijitcs.2012.10.09
6. Ge Y, Wei G (2010) GA-based task scheduler for the cloud computing systems. In: International conference web information system Mining, WISM 2010. pp 181–186
7. Tawfeek MA, El-sisi A (2013) Cloud task scheduling based on ant colony optimization. In: 8th International conference on computing engineering systems, pp 64–69
8. Lu X, Gu Z (2011) A load-adaptive cloud resource scheduling model based on ant colony algorithm. In: IEEE international conference cloud computing intelligence system, pp 296–300
9. Mathiyalagan P, Suriya S, Sivanandam SN (2010) Modified ant colony algorithm for grid scheduling. Int J Comput Sci Eng 2:132–139.
10. Kennedy J, Eberhart R (1995) Particle swarm optimization pron int conf neural networks, vol 4, IEEE, 1995,p, 1942-8
11. Ramezani F, Lu J, Hussain FK (2014) Task-based system load balancing in cloud computing using particle swarm optimization. Int J Parallel Program 42:739–754. https://doi.org/10.1007/s10766-013-0275-4
12. Zara Pooranian et al. (2015) An efficient meta-heuristic algorithm for grid computing. J Comb Optim (2015), New York, https://doi.org/10.1007/s10878-013-9644-6

13. Sidhu MS, Thulasiraman P, Thulasiram RK (2013) A load-rebalance PSO heuristic for task matching in heterogeneous computing systems. In: Swarm intelligence (SIS), 2013 IEEE Symposium, pp 180–187
14. Wang T, Liu Z, Chen Y, et al (2014) Load balancing task scheduling based on genetic algorithm in cloud computing. In: IEEE 12th international conference on dependable, autonomic security computing, pp 146–152
15. Dam S, Mandal G, Dasgupta K, Dutta P (2014) An ant colony based load balancing strategy in cloud computing. Adv Comput Netw Inform 2:403–413. https://doi.org/10.1007/978-3-319-07350-7
16. Yeboah T, Odabi OI (2015) Hybrid bee ant colony algorithm for effective load balancing and job scheduling in cloud computing. West African J Ind Acad Res 13:54–59
17. Zhang Z, Zhang X (2010) A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. In: 2nd International conference on industrial mechatronics and automation, pp 240–243
18. Khan S, Sharama N (2014) Effective scheduling algorithm for load balancing (SALB) using Ant colony optimization in cloud computing. Int J Adv Res Comput Sci Softw Eng 4:966–973
19. Liu A, Wang Z (2008) Grid task scheduling based on adaptive ant colony algorithm. In: International conference on management e-commerce e-government grid. pp 415–418
20. Bagherzadeh J, MadadyarAdeh M (2009) An improved ant algorithm for grid scheduling problem. In: 14th International CSI computing conference, pp 323–328
21. Hongyan Cui, Yang Li et al. (2017) Cloud service reliability modelling and optimal task scheduling, IET Communication, pp-161-167, https://doi.org/10.1049/iet-com.2016.0417
22. Zhang L, Chen Y, Sun R (2008) A task scheduling algorithm based on PSO for grid computing. Int J Comput Intell Res 4:37–43. https://doi.org/10.1109/ISDA.2006.253921
23. Pacini E, Mateos C, Garc C (2014) Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments. CLEI Electron J 14:1–14
24. Priyadarsini RJ, Arockiam L (2015) PBCOPSO: A parallel optimization algorithm for task scheduling in cloud environment. Indian J Sci Technol 8:6–10. https://doi.org/10.17485/ijst/2015/v8i
25. Izakian H, Ladani BT, Zamanifar K, Abraham A (2009) A novel particle swarm optimization approach for grid job scheduling. Inf Syst Technol Manag 31:100–109. https://doi.org/10.1007/978-3-642-00405-6_14
26. Al-Maamari A, Omara Fa (2015) Task scheduling using hybrid algorithm in cloud computing environments. IOSR J Comput Eng 17:2278–2661. https://doi.org/10.9790/0661-173696106
27. Abdi S, Motamedi SA, Sharifian S (2014) Task scheduling using modified PSO algorithm in cloud computing environment. In: International conference on machine learning, electrical and mechanical engineering, pp 37–41
28. Aron R, Chana I, Abraham (2015) Ahyper-heuristic approach for resource provisioning-based scheduling in grid environment. J Supercomput 71:1427–1450. https://doi.org/10.1007/s11227-014-1373-9
29. HE Hua, XU Guangquan et al. (2016) AMTS: Adaptive multi-objective task scheduling strategy in cloud computing, China Communications
30. Navimipour NJ (2015) Task scheduling in the cloud environments based on an artificial bee colony algorithm. In: International conference on image processing production computer science, Istanbul (Turkey), pp 38–44
31. Soni A (2015) A bee colony based multi-objective load balancing technique for cloud computing environment. Int J Comput Appl 114:19–25
32. Singh R (2015) Analysis of enhanced TDB based parallel scheduling algorithm using artificial bee colony. In: International Conference on Modeling and Simulation Analysis UKSIM-AMSS. IEEE, pp 470–475
33. Kruekaew B, Kimpan W (2014) Virtual machine scheduling management on cloud computing using artificial bee colony. In: International multi conference engineers and computer scientists, pp 1–5

34. Hashemi SM, Hanani A (2013) Solving the scheduling problem in computational grid using artificial bee colony algorithm. Adv Comput Sci Int J 2:37–41
35. Kumar RS (2014) Improving task scheduling in large scale cloud computing environment using artificial bee colony algorithm. Int J Comput Appl 103:29–32
36. Kashani MH (2011) Utilizing bee colony to solve task scheduling problem in distributed systems. In: International conference on computational intelligence on communication system networks, pp 298–303
37. Garg A, Krishna CR (2014) An improved honey bees life scheduling algorithm for a public cloud. In: International conference on contemporary computing and informatics, pp 1140–1147
38. K R Ramesh Babu and Philip Samuel (2016) Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud, Journal of network and innovative computing, pp-153-142
39. Abarghoei A, Mahdipour E, Askarzadeh M (2015) Cloud computing resource planning based on imperialist competitive algorithm. Cumhur Sci J 36:1312–1324
40. Fayazi M (2016) Resource allocation in cloud computing using imperialist competitive algorithm with reliability approach. Int J Adv Comput Sci Appl 7:323–331
41. Yakhchi S, Ghafari SM, YakhchiM et al (2015) ICA-MMT: a load balancing method in cloud computing environment. In: 2nd World symposium web application networks IEEE, pp 1–7
42. Arshad R, RafehR(2015) Deadline-constrained workflow scheduling using imperialist competitive algorithm on infrastructure as a service clouds. In: International conference on knowledge-based engineering innovation, pp 835–842
43. Amin J, Zalinda O et al (2013) A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition, IEEE, doi: 978-1-4673-5891-0/13