ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 10(26), DOI: 10.17485/ijst/2017/v10i26/97031, July 2017 ISSN (Online) : 0974-5645

A Comparative Study of Metaheuristics based Task Scheduling in Distributed Environment

Sunil Kumar[1*], Sumit Mittal[2] and Manpreet Singh[3]

[1]Department of Computer Science, Guru Nanak College for Girls, Sri Muktsar Sahib –152026, Punjab, India; ph.d.sunilarora@gmail.com
[2]Maharishi Markandeshwar Institute of Computer Technology and Business Management, Maharishi Markandeshwar University, Mullana, Ambala – 133203, Haryana, India; sumit.mittal@mmumullana.org
[3]Department of Computer Science and Engineering, Maharishi Markandeshwar University, Sadopur, Near Omaxe Flats, Ambala-Chandigarh Highway, Ambala – 133001, Haryana, India; dr.manpreet.singh.in@gmail.com

Abstract

Objectives: To make an extensive survey on various meta-heuristic and hybrid task scheduling along with their classification patterns and to find the scope of improvement in these techniques. **Method:** This paper carries to the deep study of 99 reputed research papers from Springer, IEEE, Elsevier, Scopus indexed; SCI indexed of well-known renowned journals. These research papers are selected by taking into consideration of relevance to research area. These scheduling algorithms are compared in terms of their performance metrics, environments and results. **Findings:** This paper described that there are various renowned researchers who have proposed various meta-heuristic task scheduling techniques to achieve the optimum results but after the extensive survey of various scheduling techniques based on genetic, Simulated Annealing (SA), ACO, PSO and hybrid reveals that a lot of dimensions are yet to be explored in terms of datacenter cost, virtual machine migration, energy consumption and Service-Level Agreement etc. **Application:** It discusses numerous meta-heuristic based task scheduling algorithms with their classification patterns so as to find the gap in the already proposed algorithm and suggest the untouched areas for the further research.

Keywords: Cloud Computing, Distributive Environment, Metaheuristics, NP Hard Problems, Task Scheduling

1. Introduction

Distributed computing emerged from decade to decade in the form of cluster, grid and cloud computing, has gained more popularity due to its capability to share the resources with low cost and more reliability[1,2]. Cluster computing provides access to powerful computers connected by high speed networks for fast and reliable execution of compute intensive jobs. On the other hand, grid computing can be considered as a distributed system with non-interactive jobs that engage a large set of files. Grid computing [3–5] provides seamless access to resources that spans across many virtual organizations. Grids are also one of the forms of distributed computing, where virtual network of super computers composed with a loosely coupled computers stand-in together to perform lengthy tasks. Cloud computing is a developing technology that provides different kind of services such as infrastructure, software and different applications through network[6,7]. Cloud delivers infrastructure, platform and software as an on-demand as a pay-per-use service[8]. The motto of all distributed systems is to enhance the throughput and to serve large scale computationally intensive applications by efficient utilization of distributed resources. For this purpose, there is a need to sequence the activities. The process of managing the resources and task is known as resource scheduling and task scheduling[9,10] which is a crucial issue in these environments. The research done in this field can be broadly classified into independent task scheduling[11,12] and workflow scheduling[13].

Independent task scheduling means tasks have no precedence relations with one other, so the tasks can

* Author for correspondence

be assigned as per priority list when they become free and there is no need to analyze a project digraph before allocating the specific task to processors. On the other hand, workflows are used to represent the applications comprising of various tasks connected on the basis of their data dependencies[14]. The aim of workflow system is to support the automated processing of complex and large-scale applications by utilizing the storage and compute power of underlying distributed infrastructure. Workflows are usually modeled as Directed Acyclic Graphs (DAG), where nodes correspond to tasks and edges represents the data dependency along with data transfer cost[15,16].

The task scheduling approaches can be further classified as heuristic[17], metaheuristics[18] and hybrid task scheduling techniques as shown in Figure 1. Heuristic task scheduling algorithms provide ease to schedule the task and deliver the best possible solutions, but it doesn't assure that the result is optimal[19,20]. However, these methods can be used to speed up the process of generating satisfactory results and are suitable to solve the simple problems. Metaheuristics are capable of handling enormous search space to locate optimal solution for task scheduling problem within polynomial time[21]. These algorithms are generally used for the complex problems and provide both general structure and guidelines for developing a heuristic for solving computational problems. Hybrid techniques add up the feature of both the heuristic and met heuristic and can be further classified as multi-criteria based optimization techniques[22].

Figure 1. Classifications of meta-heuristics.

The structure of the paper is organized as follows: Taxonomy of met heuristics is described in Section 2. A critical analysis of research work pertaining to scheduling of tasks based on Genetic Algorithm (GA) Simulated Annealing (SA), particle swam optimization and ant colony optimization etc. is carried out in Section 3. Section 4 concludes the paper and it also provides a roadmap for future work.

2. Taxonomy of Metaheuristics

Metaheuristics are the search based strategies to find the near-optimal solution without getting caught in cramped areas of the search space. Depending upon the procedure adopted for the construction of solution; metaheursitics can be classified as local search, constructive, population-based and hybrid metaheuristics as shown in Figure 1. Local search metaheuristics usually start with a feasible solution and try to improve the quality of the solution with each iteration[23]. The search terminates as soon as a local minimum is attained. Constructive metaheuristics[24] builds solutions from their constituting elements by adding the best possible element at each iteration. In order to generate better solutions, a local search phase is taken into account after the construction phase. Population-based metaheuristics provides a convenient way for finding the near optimal solution by incorporating search processes which describe the evolution of a set of points in the search space. Hybrid metaheuristics are capable of yielding better results for complex combinatorial optimization problems by combining the prominent features of metaheuristics of different classes[25].

3. Meta-Heuristic based Task Scheduling

The application of meta-heuristic to solve combinatorial optimization problems including task scheduling is gaining lot of importance. The main goal is to allow compounded moves or to generate next solution for local search in an efficient manner. Many researchers are actively addressing the metaheuristics based scheduling schemes; however, the review in this section is structured around the four commonly used metaheuristics as outlined below:

3.1 Genetic Algorithm based Scheduling

GA has proved to be a useful meta-heuristics for generating high eminence solutions for solving combinatorial optimization problems including task scheduling[26]. GA uses the terminology of real life genetic system of human beings. The high level description of GA is as follows:

1. Create a population of initial solutions
2. Find the fitness value of each solution
3. **While** (*Termination condition is not satisfied*)
    Select individuals from the population
    Apply Crossover to these individuals
    Apply Mutation to few individuals elements
    Replace the population with new individuals

End

4. Output the best solution

In conventional GA, initial population is generated randomly. To obtain best possible results and to increase the convergence pace of the GA, some heuristic approaches can be integrated to generate the initial population[27]. In[28], authors used Longest Job to Fastest Processor (LJFP) and Smallest Job to Fastest Processor (SJFP) as heuristics for generation of initial population. In[29], Max-Min heuristic has been used to generate initial population in[30], Best-Fit and Round-Robin methods are used to select good candidate resources for tasks. The role of fitness function is to determine the suitability of chromosomes and it can be evaluated on make span, energy consumption or execution cost. The selection operators like Roulette Wheel Strategy, Binary Tournament Selection, Elitism, and Rank selection operators have been used by researchers to select chromosomes for implication of crossover[31]. The crossover is used to create offspring by interchanging the genes between chromosomes. On the other hand, the mutation process will change the value of randomly selected gene to get the modified gene. Various crossover operators (uniform, one-point, two-point) and mutation operators like Simple Swap[32], Swap and Move[30] have been proposed to create offspring and to get the mutant.

In[11], authors proposed two GA based algorithms as Critical Path Genetic Algorithm (CPGA) and Task Duplication Genetic Algorithm (TDGA). CPGA performs the rescheduling of critical path nodes to reduce the idle time of processors and to manage the load between the processors. It also handles the situation when two or more scheduling solutions are of same length. On the other hand, TDGA is used to overcome the communication overhead with the help of task duplication techniques. A Modified Genetic Algorithm (MGA)[33] which produces the initial population with enhanced Max-Min technique is presented to get the optimum result in term of make span. As compared to standard Max-Min algorithm, the average execution time is used for the selection of tasks in enhanced Max-Min. The performance exhibited by various GA based task scheduling algorithms is presented in Table 1[34–40].

3.2 Simulated Annealing based Scheduling

It is applied to solve optimization problems and is typically based on thermodynamic mechanism. In SA, the objective function is used to compare the current solution with the random neighboring solution[41,42]. If there is improvement in solution then it is accepted and sometimes a fraction of inferior solutions are also accepted to escape local maxima while searching for global optima. The probabilistic selection of accepting inferior solutions depends on temperature value, which is reduced gradually at each iteration of the algorithm[43,44]. With time, this technique gets the popularity and lot of work has been done with this algorithm to schedule the tasks. The high level description of SA algorithm is as follows:

1. Generate an initial solution and also set the initial temperature.
2. **While** (*Termination condition is not satisfied*)

Generate another random solution

Evaluate the fitness value of both solutions in terms of energy

If (difference of fitness values is less than or equal to zero)

Then consider the new solution for next iteration

Else consider the new solution for next generation with probability based on current temperature and fitness value difference

Update temperature value

End

3. Output the best solution

Table 1. Performance comparison of task scheduling algorithms based on GA

Sr. No	Technique	Performance Matrix	Environment/ Simulator	Results
1.	[34]A variant of GA based on load priority	Execution time	CloudSim	Significant reduction in execution time as compared to standard GA
2.	[35]Priority based GA to optimize the total cost of workflow	Normalized schedule length	Java	Proposed algorithm results in lesser schedules length in comparison to standard GA for all synthetic workflows (Montage, Epigemonics, SIPHT, LIGO, CyberShake)
3.	[36]Improved Adaptive heuristic algorithm (IAHA) based on tasks prioritization	Makespan, Load balancing, Failure rate of tasks	CloudSim	IAHA gives better response for all performance metrics in comparison to other traditional GA approaches
4.	[33]MGA based on Max-Min for initial population generation	Makespan	CloudSim	The performance in terms of makespan is exhibited as: MGA < GA-LCFP < Enhanced Max-Min < IGA (Improved GA)< Improved Max-Min
5.	[37]GA based on inter-nodes load balancing and task spanning time	Makespan, Load balancing, Running time of jobs	Matlab	Better performance in term of total running time of jobs and load variance w.r.t Adaptive Genetic Algorithm Convergence speed of proposed algorithm is not better as compared to Adaptive GA due to its consideration for other parameters
6.	[38]GA based decentralized model for DAG scheduling	Schedule length	Cluster of 11 nodes	Proposed approach results in generation of best schedule length instead of Centralized GA and Decentralized non-cooperative GA
7.	[39]Priority in terms of expected completion time is integrated into fitness function of proposed GA.	Convergence speed	Cloudsim	For the chosen simulation setup, proposed algorithm generated optimum computation cost in almost half of iterations as compared to Adaptive GA
8.	[40]GA based scheduler for task handling in Hadoop Map Reduce	Makespan, Resource utilization	Java	Proposed approach improves average resource utilization w.r.t. FIFO and delay scheduling policy by 33% and 18% In terms of makespan, it outperforms FIFO and delay scheduling policy by 29% and 15%

In[45], authors proposed SA based solution for scheduling applications in a dynamic multi-cloud system. The objective of proposed algorithm is to optimize both performance and cost, while taking into account the heterogeneity of the virtual machines. A genetic SA algorithm for task scheduling in cloud environment is presented. In this algorithm, the features of SA with GAs are merged to give due consideration to QoS parameters for efficient resource allocation and utilization in the cloud. The result of GA is made input to SA to get the optimum result for resource allocation. Table 2 exhibits the performance of various SA based task scheduling algorithms[46–53].

3.3 Particle Swarm Optimization based Scheduling

Particle Swarm Optimization (PSO) is a population based optimization to search for optimal value of given a given problem[54–56]. PSO is initialized with random solutions (group of particles) and then searches for better solutions by updating generations. In every iteration, the motion of each particle is tracked to determine the best position of each particle and for entire swarm. These best position values are used to control the movements of the particles in next iteration[57]. The iteration is repeated until a near-optimal solution is discovered. The high level description of PSO algorithm is as follows:

1. Generate an initial population of particles

2.While (*Termination condition is not satisfied*)

Apply fitness function to evaluate each particle's position

Find the best solution of each particle *(if current position of particle is*

better then update the best solution)

Find the best solution of all particles(*choose the particle according to*

previous best position)

Make appropriate updates in velocity of each particle

Make appropriate updates in position of each particle

End

3. Output the best solution

In[58], authors proposed a PSO based mathematical model taking into account make span, communication cost and load for scheduling and allocation of tasks among cloud resources. The proposed algorithm can improve the reliability of system by rescheduling unmapped tasks on other available resources. APSO based resource allocation and scheduling technique for scientific workflows on Infrastructure as a Service (IaaS) clouds is presented[59] considering both the execution cost and deadline constraints. In[60], authors presented a variant of PSO with Adaptive Weighted Sum (AWS) method for reducing the make span and flow time of tasks in heterogeneous environment. The introduction of an acceleration factor in proposed algorithm enhances the search capability at global level to overcome the local optima. Table 3 presents the performance exhibited by various PSO based task scheduling algorithms[61–69].

Table 2. Performance comparison of task scheduling algorithms based on SA

Sr. No	Technique	Performance Matrix	Environment/ Simulator	Results
1.	[46]Mutation Based SA based on traditional SA, mutation with one change and a modified MCT heuristic	Makespan	Matlab	Reduction in Makespan of proposed technique is around 18 and 3 units in comparison to Min-Min and RGSGCS
2.	[47]SA based tasks scheduling for grid environment	Makespan	Java	Makespan reduction in proposed algorithm is about 34% in comparison to on-line mode
3.	[48]SA approach for scheduling customer's job in cloud satisfying various QoS parameters	Cost, Execution time	Not mentioned	Minimum makespan and cost in comparison to GA and a mapping algorithm
4.	[49]Incorporating SA with discrete PSO to improve quality of solutions for grid scheduling problem	Cost	Not mentioned	The impact of communication cost on total cost is minimum in proposed approach
5.	[50]Genetic SA algorithm for scheduling tasks in Cloud based on their QoS requirements	Convergence speed, Schedule length	Not mentioned Simulation setup consist of eight nodes with 20 tasks of different QoS requirements	The simulation results show that the algorithm efficiently completes resource search (in 743 iterations) and schedules tasks
6.	[51]SA based dynamic load balancing in grid	Execution Time	Java	Proposed algorithm yields near optimal solution in reasonable time
7.	[52]Resource prediction based SA for scheduling jobs on heterogeneous grids	Response Time, Cost (Computational & Communication)	SchedSim	Proposed approach generates low response time and cost as compared to Lowest[AES], Route[ARL], Round-Robin on 32-node cluster and 512-node grid
8.	[53]Proposed mimetic algorithm uses SA for local search while scheduling tasks in distributed environment	Makespan, Communication cost, Resource utilization	Not mentioned	Better results than GA and TS for performance metrics with varying tasks, population size, iterations

Table 3. Performance comparison of task scheduling algorithms based on PSO

Sr. No	Technique	Performance Matrix	Environment/ Simulator	Results
1.	[61]Discrete Symbolic Organism Search algorithm for task scheduling in cloud	Makespan	Cloudsim	Makespan improves in proposed algorithm with increase in search space Makespan reduces by 3.8 to 25.5% in comparison to combined SA and PSO approach
2.	[62]Load Balancing Mutation PSO for Task Scheduling in cloud	Makespan Roundtrip Time	CloudSim	Result of Round trip time is: Standard PSO > Mutation PSO > Random > Proposed > Longest Cloudlet Fastest Processor heuristics Result of execution time is: Standard PSO > Mutation PSO > Random > Longest Cloudlet Fastest Processor heuristics > Proposed
3.	[63]Integer PSO based task scheduling in cloud environment	Makespan	Not mentioned	With reference to No. of Tasks: Cost improvement 5% to 6.5% and Makespan Improvement 11% to 15 % With reference to No. of VMs: Cost improvement 5% to 31% and Makespan Improvement 10% to 35%
4.	[64]Workflow scheduling in cloud using improved PSO	Speedup ratio, Makespan, Load balancing rate	Matlab	In comparison to GA, the overall speed is improved by 3.8%. Makespan reduction is 6.25% and 4.18% in comparison to GA and standard PSO Results pertaining to load balancing rate is: GA< Standard PSO< Proposed PSO
5.	[65]Workflow scheduling in cloud using PSO considering both computation and communication cost	Cost	Not mentioned	Cost incurred in proposed algorithm is 3 times lower than 'Best Resource Selection' (BRS) algorithm while processing a data set of 1024MB The convergence for simulated applications in proposed PSO is achieved in 20 to 30 iterations
6.	[66]Improved Binary PSO for scheduling tasks in green cloud	Execution time	Java	Proposed algorithm outperforms Sequential Scheduling approach with varying tasks and VMs
7.	[67]PSO approach for optimizing task scheduling at user and system level in cloud	Execution time, Cost	CloudSim	Execution time and cost are better as compared to sequential algorithm
8.	[68] PSO with max-min ACO for optimizing schedules in cloud	Convergence speed	Not mentioned	Hybrid PSO converges quickly as compared to PSO
9.	[57]Task Scheduling problem is reduced to task-resource assignment graph and then implemented using PSO	Convergence speed, Makespan	C-Programming	Proposed algorithm reduces makespan by 9% as that of GA Faster than GA by 1.5 times Best solution is generated in almost half iterations (17) in comparison to GA (37)
10.	[69]Grid task scheduling based on advanced no velocity PSO	Makespan	GridSim	Proposed algorithm yields better result as compared to ACO

3.4 Ant Colony Optimization based Scheduling

In ACO algorithms, artificial ants move through a solution space by making decision based on the artificial pheromones and heuristic information[16,70]. With their movements, ants construct a solution to a problem which is later on evaluated using a fitness function. [71]The ants also update the pheromone trail which is further used by them in future to control the movement in search space. The high level description of ACO algorithm is as follows:

1.Assign the initial pheromone values and set best solution as null

2.While (*Termination condition is not satisfied*)

Each ant builds a probabilistic solution based on pheromone trails and heuristic information.

Evaluate the solution of each ant using a fitness function

Update the best solution, if the fitness value of any ant provides better solution than existing best solution

Update all pheromone values

End

3. Output the best solution

In[72], authors proposed an ACO algorithm for scheduling jobs in the grid taking into account both the make span and system load. In[73], authors proposed a solution for job scheduling problem in grid based on the concept of lazy ant. Lazy ants are the mutated version of active ants and persist till the fitter lazy ants are generated in the subsequent iterations. Not only, they reduces the time complexity of the algorithm but also generate the better solutions for given objectives. In[74], a cost effective and deadline constrained scheduling algorithm for enhancing their liability of workflow execution is proposed. An ACO system based on three heuristics (two for generating feasible schedule and third one for enhancing the reliability of the system) is developed to minimize the constraints violation and to improve the reliability of schedules. The performance exhibited by various ACO based task scheduling algorithms is presented in Table 4[75–83].

3.5 Hybrid Metaheuristics based Scheduling

Hybrid Metaheuristic cracks a constraint satisfaction problem with combination of heuristic or metaheuristic task scheduling algorithms[84,85]. Hybridization of metaheuristics can be implemented by including the components from one metaheuristic into another one[86]. On the other hand, it may be implemented in the form of various heuristics/metaheuristics exchanging information with one another. The research work of various authors in the field of task scheduling based on hybrid heuristics can be summarized as:

In[87], authors proposed an algorithm exploiting the merits of PSO and ACO. In this algorithm, the initial pheromone is generated through PSO and search for best solution is carried out using Max-Min Ant Colony Algorithm. The proposed algorithm resulted in better results in terms of cost management and load management. In[88], authors combined the merits of Cuckoo Search (to perform local search) and ACO to perform the scheduling of jobs in cloud environment, with an objective to reduce the total execution time. A hybrid algorithm based on HEFT and GA is proposed to yield better performance under dynamically changing heterogeneous computational environment for variable workload[89]. The performance exhibited by various hybrid task scheduling algorithms is presented in Table 5[90–98].

Table 4. Performance comparison of task scheduling algorithms based on ACO

S. No	Technique	Performance Matrix	Environment/ Simulator	Results
1.	[75]Load Sharing ACO for scheduling of meta-tasks in grid	Waiting time, Response time	Matlab	Proposed algorithm exhibits lesser waiting and response time as compared to min-min and max-min Results are not affected due to different job arrivals timings
2.	[76]ACO based task scheduling in cloud environment	Makespan	CloudSim	Scheduling policy based on ACO Better than Default Policy
3.	[77]Self Adaptive ACO for task scheduling in cloud	Makespan, Load balancing	CloudSim	Result related to makespan is: SAACO < PACO < min-min Result related to load balancing is: SAACO > PACO > min-min
4.	[78]Task distribution based on selection of best cloud in the federation using Load Balancing ACO	Makespan, Scheduling time	BioNimbuZ	Despite, taking more time in making scheduling decision it yields schedule with lesser makespan

5.	[79]Two-way ants mechanism for scheduling workflow in cloud computing	Scheduling Time	CloudSim	Proposed ACO has better scheduling time as compared to traditional ACO
6.	[80]ACO for cloud task scheduling considering load among the nodes	Makespan, Degree of imbalance	CloudSim	Under task variation from 100 to 500, proposed solution generates lesser values for makespan and degree of imbalance w.r.t. basic ACO and FCFS
7.	[81]ACO based cloud task scheduling	Makespan, Average degree of imbalance (DI)	Cloudsim	Result related to makespan is: ACO (700)< RR (1000) < FCFS (1050) Result related to DI is: FCFS (3.7) >RR (3.6) > ACO (2.7)
8.	[82]Assignment of tasks to grid resources using ACO	Load balancing	Java	ACO improvement over RR in terms of load deviation ranges from 4.83% to 73.5%.
9.	[83]Multi-objective optimization scheduling based on ACO in cloud	Makespan, Cost, Deadline violation rate	CloudSim	Proposed algorithm improves makespan by 56.6% w.r.t. FCFS in Cost reduced from 7% to 23% compared to other methods. Deadline violation rate is reduced by 34% compared to FCFS

Table 5. Performance comparison of hybrid task scheduling algorithms

Sr. No	Technique	Performance Matrix	Environment/ Simulator	Results
1.	[90]Optimal solution generated by GA is used as initial pheromone of ACO while solving task scheduling in cloud	Convergence speed, Task execution time	Cloud Sim	GA-ACO finds optimal solution faster (28 iterations, 98%) and better than GA (50 iterations, 63%) and ACO (50 iterations, 95%). Task execution time is: GA-ACO (70) < ACO (100) < GA (120)
2.	[91]GA with gravitational emulation local search (GELS) for job scheduling on grids	Makespan, Miss ratio	Java	Result pertaining to makespan and miss ratio is: Proposed < Global searching SA < GELS < GA
3.	[92]Hybrid of Min-Min and Max-Min for grid task scheduling	Makespan, Average resource utilization	Grid Sim	Selective performs better than Min-Min, Max-Min
4.	[93]Merger of Best-Fit and Round Robin methods with GA for workflow scheduling in cloud	Makespan, Load balancing, Speedup ratio	Not mentioned	Makespan reduction in proposed is by 19.2 % in LAGA and 34.4 % with NGA Load balancing in proposed is better by 19.2 % in LAGA and 34.4 % with NGA. Speed up ratio in proposed is better by 18.2 % in LAGA and 33.8 % with NGA
5.	[94]Merits of ACO and Artificial Bee Colony (ABC) algorithm are combined for task scheduling in cloud	Execution Time	Cloud Sim	Proposed hybrid algorithm shows execution time improvement by 19% over FCFS, 11% over ABC and 9% over ACO
6.	[95]Dynamic fusion of GA and ACO for cloud workflow scheduling	Energy Consumption, Makespan, Execute Generation		As compared to standard GA, proposed hybrid does a lot of energy saving with little increase in makespan
7.	[96]Proposed algorithm integrates ACO with GA to solve task scheduling problem with multi-QoS constraints. To generate the initial pheromone efficiently for ACO, GA is invoked	Load balancing, Execution time	Cloud Sim	Compared with GA and ACO, resource utilization of proposed algorithm is increased by 28% and 24.1% Reduction in execution time for tasks under proposed hybrid is significant as compared to GA or ACO

8.	[97]PSO in combination with gravitation search for workflow scheduling	Cost	Cloud Sim	Cost reduction in proposed hybrid as compared to others is presented as: non-heuristic by 70%, gravitational search by 50%, PSO by 30%
9.	[98]Modified GA with fuzzy based fitness function to perform job scheduling in cloud	Execution time, Execution cost	Cloud Sim	Proposed algorithm exhibits 45% reduction in execution cost and 50% reduction in execution time as compared to ACO

4. Summary and Conclusion

In this paper, an attempt has been made to highlight the importance of metaheuristics in solving the task scheduling problem for distributed environment. An outline of the most commonly used metaheuristics has been presented followed by a comparative analysis of each such metaheuristic related to task scheduling. The survey also includes the close examination of the performance of hybrid metaheuristics. An extensive review of recent proposals for scheduling techniques reveals that a lot more dimension are yet to be explored in terms of datacenter cost, virtual machine migration, energy consumption and Service-Level Agreement etc.

5. References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. In Association for Computing Machinery (ACM) Future Generation Computer Systems. 2009 Jun; 25(6):599–616.
2. Maheswaran M, Ali S, Siegal HJ, Hensgen D, Freund RF. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In the Proceedings of 8th Institute of Electrical and Electronics Engineers (IEEE) Heterogeneous Computing Workshop, USA; 1999 Apr 12. p. 30–44. Crossref
3. Buyya R, Ramamohanarao K, Yu J. Workflow scheduling algorithm for grid computing. Meta-Heuristics for Scheduling in Distributed Computing Environment, Studies in Computational Intelligence, Berlin, Heidelberg, Springer. 2008; 146:173–214.
4. Barrionuevo JJD, Fard HM, Prodan R, Fahringer T. A multi-objective approach for workflow scheduling in heterogeneous environment: cluster, cloud and grid computing. In the Proceedings of the 12th Institute of Electrical and Electronics Engineers (IEEE)/ Association for Computing Machinery (ACM) International Symposium on Cluster, Cloud and Grid Computing, Canada; 2012 May 13–16. p. 300–9.
5. de Mello RF, Senger LJ, Yang LT. A routing load balancing policy for grid computing environments. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 20th International Conference on Advanced Information Networking and Applications. 2006 Apr 18–20; 1:153–8. Crossref
6. Mell P, Grance T. The NIST definition of cloud computing. National Institute of Standards and Technology Special Publication. 2011 Sep; 800-146:1–7.
7. Kara N, Soualhia M, Belqasmi F, Azar C, Glitho R. Genetic-based algorithms for resource management in virtualized IVR applications. Journal of Cloud Computing: Advances. Systems and Applications, Springer; 2014 Dec. p. 3–15.
8. Mao Y, Chen X, Li X. Max–min task scheduling algorithm for load balance in cloud computing. In the Proceedings of International Conference on Computer Science and Information Technology, Advances in Intelligent Systems and Computing, Springer. 2014; 255:457–65. Crossref
9. Chang R–S, Chang J–S, Lin P–S. An ant algorithm for balanced job scheduling in grids. Future Generation Computer Systems, Elsevier, ScienceDirect. 2009 Jan; 25(1):20–7. Crossref
10. Yi P, Ding H, Ramamurthy B. A Tabu search based heuristic for optimized joint resource allocation and task scheduling in grid/clouds. Institute of Electrical and Electronics Engineers (IEEE) International Conference on Advanced Networks and Telecommunications Systems, Kattankulathur, India; 2013 Dec 15–18. p. 1–3. Crossref
11. Omara FA, Arafa MM. Genetic algorithms for task scheduling problem. Journal of Parallel and Distributed Computing, Elsevier, ScienceDirect. 2010 Jan; 70(1):13–22.
12. Dubey S, Jain V, Shrivastava S. An innovative approach for scheduling of tasks in cloud environment. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 4th International Conference on Computing, Communications and Networking Technologies, Tiruchengode, India; 2013 Jul 4–6. p. 1–8. Crossref
13. Chopra N, Singh S. HEFT based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 4th International Conference on Computing, Communications and Networking Technologies, Tiruchengode, India; 2013 Jul 4–6. p. 1–6. Crossref
14. Mamani-Aliaga AH, Goldman A, Ngoko Y. A comparative study on task dependent scheduling algorithms for grid computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 13th Symposium on

Computer System, Petropolis, Brazil; 2012 Oct 17–19. p. 202–9. Crossref
15. Chen W–N, Zhang J. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Systems. Man and Cybernetics Part C: Applications and Reviews. 2009 Jan; 39(1):29–43. Crossref
16. Elhady GF, Tawfeek MA. A comparative study into swarm intelligence algorithms for dynamic tasks scheduling in cloud computing. Institute of Electrical and Electronics Engineers (IEEE) 7th International Conference on Intelligent Computing and Information Systems, Cairo, Egypt. 2015 Dec 12–14. p. 362–9. Crossref
17. Anousha S, Ahmadi M. An improved min-min task scheduling algorithm in grid computing. International Conference on Grid and Pervasive Computing, Lecture Notes in Computer Science, Springer. 2013; 7861:103–13.
18. Bilgaiyan S, Sagnika S, Das M. A multi-objective cat swarm optimization algorithm for workflow scheduling in cloud computing environment. Intelligent Computing, Communication and Devices, Advances in Intelligent Systems and Computing, Springer. 2015; 308:73–84.
19. Liu G, Li J, Xu J. An improved min-min algorithm in cloud computing. In the Proceedings of the International Conference of Modern Computer Science and Applications, Advances in Intelligent Systems and Computing, Springer. 2012; 191:47–52.
20. Bajaj R, Agrawal DP. Improving scheduling of tasks in a heterogeneous environment. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Parallel and Distributed System. 2004 Feb; 15(2):107–18. Crossref
21. Zhao C, Zhang S, Liu Q, Xie J, Hu J. Independent tasks scheduling based on genetic algorithm in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Conference on Wireless Communications. Networking and Mobile Computing, Beijing, China; 2009 Sep 24–26. p. 1–4.
22. Tsai C–W, Rodrigues JJPC. Metaheuristic scheduling for cloud: a survey. Institute of Electrical and Electronics Engineers (IEEE) Systems Journal. 2014 Mar; 8(1):279–91. Crossref
23. Shanmugapriya R, Padmavathi S, Shalinie SM. Contention awareness in task scheduling using Tabu search. Institute of Electrical and Electronics Engineers (IEEE) International Advance Computing Conference, Patiala, India; 2009 Mar 6–7. p. 272–7. Crossref
24. Kennedy J, Eberhart R. Particle swarm optimization. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Neural Networks, Australia. 1995 Nov 27 – Dec 1; 4:1942–8. Crossref
25. Ma Y, Wang Y. Grid task scheduling based on chaotic ant colony optimization algorithm. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 2nd International Conference on Computer Science and Network Technology, Changchun, China; 2012 Dec 29–31. p. 469–72. Crossref
26. Zhu K, Song H, Liu L, Gao J, Cheng G. Hybrid genetic algorithm for cloud computing applications. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Asia- Pacific Service Computing Conference, Jeju Island, South Korea; 2011 Dec 12–15. p. 182–7. Crossref
27. Abdulal W, Jabas A, Ramachandram S, Jadaan OA. Rank based genetic scheduler for grid computing systems. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Computational Intelligence and Communication Networks (CICN), Bhopal, India; 2010 Nov 26–28. p. 644–9. Crossref
28. Carretero J, Xhafa F, Abraham A. Genetic algorithm based schedulers for grid computing systems. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Journal of Innovative Computing. Information and Control. 2007; 3(6):1–19.
29. Xiaoguang Y, Tingbin C, Qisong Z. Research on cloud computing schedule based on improved hybrid PSO. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 3rd International Conference on Computer Science and Network Technology, Dalian, China; 2013 Oct 12–13. p. 388–91. Crossref
30. Yu J, Buyya R. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Journal on Scientific Programming– Scientific Workflow, Association for Computing Machinery (ACM). 2006 Dec; 14(3,4):214–30.
31. Magalhaes-Mendes J. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. WSEAS Transactions on Computers. 2013 Apr; 12(4):164–73.
32. Kaur S, Verma A. An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science. 2012 Sep; 4(10):74–9. Crossref
33. Singh S, Kalra M. Scheduling of independent tasks in cloud computing using modified genetic algorithm. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Computational Intelligence and Communication Networks, Bhopal, India; 2014 Nov 14–16. p. 565–9. Crossref
34. Ma J, Li W, Fu T, Yan L, Hu G. A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing. Wireless Communications, Networking and Applications, Lecture Notes in Electrical Engineering, Springer. 2015 Oct 29; 348:829–35.
35. Verma A, Kaushal S. Budget constrained priority based genetic algorithm for workflow scheduling in cloud. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Conference on Advances in Recent Technologies in Communication and Computing, Bangalore, India; 2013 Sep 20–21. p. 216–22.
36. Zhang Y, Li Y. An improved adaptive workflow scheduling algorithm in cloud environments. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)

3rd International Conference on Advanced Cloud and Big Data, Yangzhou, China; 2015 Oct 30 – Nov 1. p. 112–6. Crossref
37. Wang T, Liu Z , Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 12th International Conference on Dependable, Autonomic and Secure Computing, Dalian, China; 2014 Aug 24–27. p. 146–52. Crossref
38. Pop F, Dobre C, Cristea V. Genetic algorithm for DAG scheduling in grid environments. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania; 2009 Aug 27–29. p. 299–305. Crossref
39. Hu B, Sun X, Li Y, Sun H. An improved adaptive genetic algorithm in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, Beijing, China; 2012 Dec 14–16. p. 294–7.
40. Ge Y, Wei G. GA-based task scheduler for the cloud computing systems. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Web Information Systems and Mining, Sanya, China. 2010 Oct 23–24; 2:181–6. Crossref
41. Raj R. Beyond simulated annealing in grid scheduling. International Journal on Computer Science and Engineering. 2011 Mar; 3(3):1312–8.
42. Nikolaev A, Jacobson S. Simulated annealing handbook of meta-heuristics. International Series in Operations Research and Management Science, Springer; 2010. p. 1–39.
43. Pandit D, Chattopadhyay S, Chattopadhyay M, Chaki N. Resource allocation in cloud using simulated annealing. In the Conference on Applications and Innovations in Mobile Computing; 2014 Feb. p. 21–7.
44. Portaluri G, Giordano S. Power efficient resource allocation in cloud computing data centers using multi-objective genetic algorithms and simulated annealing", 4th International Conference on Cloud Networking (CloudNet). IEEE 2015; 319 – 321.
45. Moschakis IA, Karatza HD. Multi-criteria scheduling of bag-of-tasks applications on heterogeneous interlinked clouds with simulated annealing. The Journal of Systems and Software, Elsevier, ScienceDirect. 2015 Mar; 101(c):1–14. Crossref
46. Abdulal W, Jabas A, Ramachandram S, Jadaan OA. Mutation based simulated annealing algorithm for minimizing makespan in grid computing systems. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 3rd International Conference on Electronics Computer Technology (ICECT), Kanyakumari, India. 2011 Apr 8–10; 6:90–4. Crossref
47. Fidanova S. Simulated annealing for grid scheduling problem. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) John Vincent Atanasoff International Symposium on Modern Computing, Sofia, Bulgaria; 2006 Oct 3–6. p. 41–5. Crossref
48. Abdullah M, Othman M. Simulated annealing approach to cost-based multi-quality of service job scheduling in cloud computing environment. American Journal of Applied Sciences. 2014; 11(6):872–7. Crossref
49. Chen R–M, Shiau D–F, Lo S–T. Combined discrete particle swarm optimization and simulated annealing for grid computing scheduling problem. International conference on Intelligent computing, Emerging Intelligent Computing Technology and Applications with Aspects of Artificial Intelligence, Lecture Notes in Computer Science, Springer. 2009; 5755:242–51. Crossref
50. Guo-ning Gan G, Huang T, Gao S. Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Intelligent Computing and Integrated Systems, Guilin, China; 2010 Oct 22–24. p. 60–3.
51. Paletta M, Herrero P. A simulated annealing method to cover dynamic load balancing in grid environment. International Symposium on Distributed Computing and Artificial Intelligence, Advances in Soft Computing, Springer. 2009; 50:1–10. Crossref
52. de Mello RF, Senger LJ. On simulated annealing for the scheduling of parallel applications. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 20th International Symposium on Computer Architecture and High Performance Computing, Campo Grande, Brazil; 2008 Oct 29 – Nov 1. p. 29–36. Crossref
53. Kashani MH, Jahanshahi M. Using simulated annealing for task scheduling in distributed systems. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Computational Intelligence, Modelling and Simulation, Brno, Czech Republic; 2009 Sep 7–9. p. 265–9. Crossref
54. Xue S, Shi W, Xu X. A heuristic scheduling algorithm based on PSO in the cloud computing environment. International Journal of u-and e-Service, Science and Technology. 2016; 9(1):349–62.
55. Masdari M, ValiKardan S, Shahi Z, Azar SI. Towards workflow scheduling in cloud computing: a comprehensive analysis. Journal of Network and Computer Applications, Elsevier, ScienceDirect. 2016 May; 66:64–82. Crossref
56. Shahverdy M, Mahdavi Z, Mahdavi AH, Mirzaee A, Asgari M. Review the use of improved scheduling PSO algorithm in Grid Computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 3rd International Conference on Communication Software and Networks (ICCSN), China. 2011 May 27–29. p. 86–9. Crossref
57. Chen T, Zhang B, Hao X, Dai Y. Task scheduling in grid based on particle swarm optimization. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Symposium on Parallel and Distributed Computing, Timisoara, Romania; 2006 Jul 6–9. p. 238–45. Crossref

58. Awad AI, El-Hefnawy NA, Abdel-Kader HM. Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Computer Science, Elsevier, ScienceDirect. 2015; 65:920–9. Crossref
59. Rodriguez MA, Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Cloud Computing. 2014 Apr–Jun; 2(2):222–35. Crossref
60. Vidya G, Sarathambekai S, Umamaheswari K, Yamunadevi SP. Task scheduling using adaptive weighted particle swarm optimization with adaptive weighted sum. Procedia Engineering, Elsevier, ScienceDirect. 2012; 38:3056–63. Crossref
61. Abdullahi M, Ngadi MA, Abdulhamid SM. Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Generation Computer Systems, Elsevier, ScienceDirect. 2016 Mar; 56:640–50.
62. Awad AI, El-Hefnawy NA, Abdel_kader HM. Enhanced particle swarm optimization for task scheduling in cloud computing environments. International Conference on Communication, Management and Information Technology, Procedia Computer Science, Elsevier, ScienceDirect. 2015; 65:920–9.
63. Beegom ASA, Rajasre MS. A particle swarm optimization based pareto optimal task scheduling in cloud computing. Advances in Swarm Intelligence, Lecture Notes in Computer Science, Springer. 2014; 8795:79–86.
64. Chitra S, Madhusudhanan B, Sakthidharan GR, Saravanan P. Local minima jump PSO for workflow scheduling in cloud computing environments. Advanced in Computer Science and Its Applications, Lecture Notes in Electrical Engineering, Springer. 2014; 279:1225–34.
65. Pandey S, Wu L, Guru SM, Buyya R. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 24th International Conference on Advanced Information Networking and Applications, Perth, Australia; 2010 Apr 20–23. p. 400–7. Crossref
66. Xu L, Wang K, Ouyang Z, Qi X. An improved binary PSO-based task scheduling algorithm in green cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 9th International Conference on Communications and Networking in China, Maoming, China; 2014 Aug 14–16. p. 126–31. Crossref
67. Saxena D, Saxena S. Highly advanced cloudlet scheduling algorithm based on particle swarm optimization. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 8th International Conference on Contemporary Computing (IC3), Noida, India; 2015 Aug 20–22. p. 111–6. Crossref
68. Xiaoguang Y, Tingbin C, Qisong Z. Research on cloud computing schedule based on improved hybrid PSO. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 3rd International Conference on Computer Science and Network Technology, Dalian, China; 2013 Oct 12–13. p. 388–91. Crossref
69. Wang M, Zeng W, Wu K. Grid task scheduling based on advanced no velocity PSO. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Internet Technology and Applications, Wuhan, China; 2010 Aug 20–22. p. 1–4.
70. Hu Y, Xing L, Zhang W, Xiao W, Tang D. A knowledge-based ant colony optimization for a grid workflow scheduling problem. International Conference in Advances in Swarm Intelligence, Lecture Notes in Computer Science, Springer. 2010; 6145:241–8. Crossref
71. Achary R, Vityanathan V, Raj P, Nagarajan S. Dynamic job scheduling using ant colony optimization for mobile cloud computing. Intelligent Distributed Computing, Advances in Intelligent Systems and Computing, Springer. 2015; 321:71–82. Crossref
72. Chang R–S, Chang J–S, Lin P–S. An ant algorithm for balanced job scheduling in grids. Future Generation Computer Systems, Elsevier, ScienceDirect. 2009 Jan; 25(1):20–7. Crossref
73. Tiwari PK, Vidyarthi DP. Improved auto control ant colony optimization using lazy ant approach for grid scheduling problem. Future Generation Computer Systems, Elsevier, ScienceDirect. 2016 Jul; 60:78–89. Crossref
74. Kianpisheh S, Charkari NM, Kargahi M. Reliability-driven scheduling of time/cost-constrained grid workflows. Future Generation Computer Systems, Elsevier, ScienceDirect. 2016 Feb; 55:1–16. Crossref
75. Kokilavani T, Amalarethinam DIG. An ant colony optimization based load sharing technique for meta task scheduling in grid computing. Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing, Springer. 2013; 177:395–404.
76. Wang L, Ai L. Task scheduling policy based on ant colony optimization in cloud computing environment. In the Proceedings of the 2nd International Conference on Logistics, Informatics, Service Science, Springer; 2013. p. 953–7. Crossref
77. Sun W, Ji Z, Sun J, Zhang N, Hu Y. SAACO: A Self Adaptive Ant Colony Optimization in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Conference on Big Data and Cloud Computing, Dalian, China; 2015 Aug 26–28. p. 148–53. Crossref
78. Oliveira G, Ribeiro E, Ferreira D, Araujo A. ACOsched: a scheduling algorithm in a federated cloud infrastructure for bioinformatics applications. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Bioinformatics and Biomedicine, Shanghai, China; 2013 Dec 18–21. p. 8–14. Crossref
79. Zhou Y, Huang X. Scheduling workflow in cloud computing based on ant colony optimization algorithm. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 6th International Conference on Business Intelligence and Financial Engineering, Hangzhou, China; 2013 Nov 14–16. p. 57–61. Crossref
80. Li K, Xu G, Zhao G, Dong Y, Wang D. Cloud task schedul-

ing based on load balancing ant colony optimization. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 6th Annual ChinaGrid Conference, Liaoning, China; 2011 Aug 22–23. p. 3–9. Crossref
81. Tawfeek MA, El-Sisi A, Keshk AE, Torkey FA. Cloud task scheduling based on ant colony optimization. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 8th International Conference on Computer Engineering and Systems, Cairo, Egypt; 2013 Nov 26–28. p. 64–9.
82. Michalas A, Louta M. Adaptive task scheduling in grid computing environments. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 4th International Workshop on Semantic Media Adaptation and Personalization, San Sebastian, Spain; 2009 Dec 14–15. p. 115–20. Crossref
83. Zuo L, Shu L, Dong S, Zhu C, Hara T. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. Institute of Electrical and Electronics Engineers (IEEE) Access: Big Data Services and Computational Intelligence for Industrial Systems. 2015 Dec 17; 3:2687–99.
84. Fan Y, Liang Q, Chen Y, Yan X, Hu C, Yao H, Liu C, Zeng D. Executing time and cost-aware task scheduling in hybrid cloud using a modified DE algorithm. International Symposium on Intelligence Computation and Applications, Computational Intelligence and Intelligent Systems, Communications in Computer and Information Science, Springer. 2016; 575:74–83.
85. Shojafar M, Javanmardi S, Abolfazli S, Cordeschi N. FUGE: a joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, Springer. 2015 Jun; 18(2):829–44.
86. Xian–Jia R. Research on hybrid task scheduling algorithm simulation of ant colony algorithm and simulated annealing algorithm in virtual environment. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 10th International Conference on Computer Science and Education, Cambridge, UK; 2015 Jul 22–24. p. 562–5. Crossref
87. Kalpana C, Kumar UK, Gogulan R. Max-min particle swarm optimization algorithm with load balancing for distributed task scheduling on the grid environment. International Journal of Computer Science Issues. 2012 May; 9(3):365–73.
88. Babukarthik RG, Raju R, Dhavachelvan P. Hybrid algorithm for job scheduling: combining the benefits of ACO and cuckoo search. Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing, Springer. 2013; 177: 479–90.
89. Nasonov D, Butakov N, Balakhontseva M , Knyazkov K, Boukhanovsky AV. Hybrid evolutionary workflow scheduling algorithm for dynamic heterogeneous distributed computational environment. International Joint Conference SOCO’14-CISIS’14-ICEUTE’14, Advances in Intelligent Systems and Computing, Springer. 2014; 299:83–92. Crossref
90. Liu C–H, Zou C–M, Wu P. A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, Xian Ning, China; 2014 Nov 24–27. p. 68–72. Crossref
91. Pooranian Z, Shojafar M, Tavoli R, Singhal M, Abraham A. A hybrid metaheuristic algorithm for job scheduling on computational grids. Informatica. 2013; 37(2):157–64.
92. Etminani K, Naghibzadeh M. A min-min max-min selective algorithm for grid task scheduling. In the Proceedings of the 3rd Institute of Electrical and Electronics Engineers (IEEE) /IFIP International Conference in Central Asia on Internet, Tashkent, Uzbekistan; 2007 Sep 26–28. p. 1–7.
93. Delavar AG, Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. Cluster Computing, Springer. 2014 Mar; 17(1):129–37. Crossref
94. Madivi R, Kamath SS. An hybrid bio-inspired task scheduling algorithm in cloud environment. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th International Conference on Computing, Communication and Networking Technologies, Hefei, China; 2014 Jul 11–13. p. 1–7. Crossref
95. Zhao J, Qiu H. Genetic algorithm and ant colony algorithm based energy-efficient task scheduling. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 3rd International Conference on Information Science and Technology, Yangzhou, China; 2013 Mar 23–25. p. 946–50. Crossref
96. Dai Y, Lou Y, Lu X. A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China. 2015 Aug 26–27; 2:428–31. Crossref
97. Mirzayi S, Rafe V. A hybrid heuristic workflow scheduling algorithm for cloud computing environments. Journal of Experimental and Theoretical Artificial Intelligence. 2015 Mar 16; 27(6):721–35. Crossref
98. Javanmardi S, Shojafar M, Amendola D, Cordeschi N, Liu H, Abraham A. Hybrid job scheduling algorithm for cloud computing environment. In the Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications, Advances in Intelligent Systems and Computing, Springer. 2014; 303:43–52. Crossref