IJICTR
International Journal of Information & Communication Technology Research

Volume 13- Number 4 – 2021 (28 -35)

A Survey on Task Scheduling Algorithms in Cloud Computing for Fast Big Data Processing

Zahra Jalalian
School of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
z_jalalian@comp.iust.ac.ir

Mohsen Sharifi*
School of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
msharifi@iust.ac.ir

Received: 3 August 2021 - Accepted: 17 October 2021

Abstract—**The recent explosion of data of all kinds (persistent and short-lived) have imposed processing speed constraints on big data processing systems (BDPSs). One such constraint on running these systems in Cloud computing environments is to utilize as many parallel processors as required to process data fast. Consequently, the nodes in a Cloud environment encounter highly crowded clusters of computational units. To properly cater for high degree of parallelism to process data fast, efficient task and resource allocation schemes are required. These schemes must distribute tasks on the nodes in a way to yield highest resource utilization as possible. Such scheduling has proved even more complex in the case of processing of short-lived data. Task scheduling is vital not only to handle big data but also to provide fast processing of data to satisfy modern time data processing constraints. To this end, this paper reviews the most recently published (2020-2021) task scheduling schemes and their deployed algorithms from the fast data processing perspective.**

Keywords: Fast Data Processing; Cloud Computing; Task Scheduling; Resource Utilization; Scheduling Algorithms.

Article type: Research Article

[DOI: 10.52547/itrc.13.4.28]

I. Introduction

Nowadays, owing to enormous number of numerical tasks due to data explosion, partly because of Internet of Things (IOT) device usage extension among users, the single core processors which leveraged from higher processing frequencies as well as more parallelism in early paradigms of computing methods, is not able to operate appropriately as a result of chip integration saturation. This, in fact, makes the power consumption and heat dissipation as the most challenging obstacles in modern digital ecosystems.

* Corresponding Author

The ongoing controversy is addressed properly by advent of multi-core processers that have been visibly widespread nowadays. In this context, Heterogeneous Multi-Core Processors (HMCPs) are the most promising and practical solutions due to their lower consumption of power as well as higher ability of parallelism in comparison to the homogeneous multi-core processors [1]. It is worth mentioning that HMCPs are required an appropriate software to provide the best performance, making Task Scheduling (TS) crucial to improve their benefits. That is to say, the TS between sub-processors of HTSTC and the algorithms with the capability of using out-of-processor resources have been gained conspicuous popularity among scholars.

On the other hand, Cloud computing that is conventionally accompanied by distributed, Grid, and parallel computing, is another ecosystem that could leverage from TS due to its heterogeneous nature as well [2-8]. It should be noted that due to its popularity, the Cloud systems are employed by users to utilize resources or storage as needed, or to define countless applications that could be hosted on the Cloud by stockholders. Therefore, the LB[1] and CT[2] are usually investigated for any Cloud system, but owing to the expansion of cluster size, the power is another factor that has got extensive attention in new systems, leading to the emergence of low power algorithms for gaining maximum power loss. This fact has efficiently decreased the concerns about the costs such as hardware maintenance or operating costs, encouraging the participants to use or create more and more applications [2].

In addition, the expulsion of data has been imposing some processing speed constraints on BDPSs[3], making them utilize more parallel processors to accomplish fast data processing. The nodes of the whole system or CC[4] in the more advanced cases, then, must have crowded clusters of computational terminals. To solve this problem, Parallel Processing, PP, is one persuasive solution which is carried out by defining tasks, dividing application programs, and distributing tasks among Cloud Nodes (CNs). Parallelism has led to low processing time. In addition, enormous number of CNs and architectures with vast variety of resource types can be integrated together to compose a Cloud ecosystem for accomplishing parallel tasks called Cloud of Parallel Task Processing, CPTP. Task scheduler is vital in this huge system to share and allocate the tasks with computation resources and nodes, maximizing the resource utilization rates and makespan optimization, or optimizing the time that is required for accomplishment of a task set. To optimize the performance in terms of power consumption of the system, tasks are being mapped among processors of the target platform in Multiprocessor Task Scheduling Problem (MTSP).

Directed Acyclic Graphs (DAGs) are usually used to illustrate the relationship between tasks [8]. To visualize this situation DAG graph consists of weights, and actually the weights are representing the data communication between successive tasks which can be shown in Fig. 1 as an example of a DAG graph [9]. In this graph, 8 tasks and their execution cost are illustrated as an example. The weights show the number of tasks and cost in terms of millions of processing instructions [10].

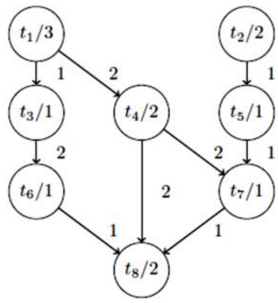

Fig. 1. An 8-task double processor MTSP [9]

Finding optimized solutions for MTSP requires high computational complexity which could be solved by discovering set of solutions, resulting in different outputs, ranging from a global optimization to an approximation of the problem's solution. Due to the flexibility, evolutionary algorithms or EAs could be promising candidates [7], [10-12].

Owing to the recent advances in Cloud technology, and the unique advantages that it provides for service customers, the number of tasks that must be handled by Cloud is dramatically increasing. Users are expecting on-time solutions more and more these days regardless of task numbers, further highlighting the significance of TS. The task scheduling must have low latency; moreover, it has to be resource usage efficient. In fact, this issue is of great importance in whole system performance.

To improve the effectiveness of TS, ample researches have been carried out such as Min-Min or Min-Max techniques, MET[5], FCFS[6], and so on. Unfortunately, it is established based on recent researches that these classic methods are not of the sufficient efficiency for large-scale Cloud environments. The Metaheuristic Algorithms (MAs) on the other hand, have shown superior quality to overcome the obstacles of efficient system set-up and reaching near optimal solutions with lower scheduling time [13-16]. In this review paper, the most important research works published during 2020-2021 in this field are summed up to get insight regarding fast processing provided by efficient TS in Cloud environments. Abbreviations are listed in the Appendix.

II. CLOUD COMPUTING AND TASK SCHEDULING

The CC system is composed of three main layers based on service mode. The first one is Interface as a service or IaaS, the second one is called Platform as a service or PaaS, and the third one is SaaS or Software as a service. TS algorithms are fundamental for all three

[1] Load Balancing
[2] Completion Time
[3] Big Data Processing Systems
[4] Cloud Computing
[5] Minimum Execution Time
[6] First Come First Serve

[DOI: 10.52547/itrc.13.4.28]

[DOI: 10.52547/itrc.13.4.28]

layers, specially, they are more important in the IaaS layer duo to its role as a computing resource provider. Notably, TS is important while the applications are used more than once and so many times, the situation that is usual in current computation paradigm [2].

The way that Cloud systems plan, break down, and allocate countless tasks to lots of VMs while performing tasks with short run time is task Schelling [17]. Assigned by a machine, each user task could be carried out, and by increasing the user tasks, the user service process is involved in an agreement with the service (or Cloud) provider called SLA or service level agreement, which is actually a contract in terms of expected distinctly-defined QoS including time of task completion, the required cost, and the level of security. In fact, the number of VMs is a part of contract that Cloud provider ensures to support based on the user requirement even before any real service provided based on the predicted resources needed to do user request [18]. This has led to an increased importance for TS as if it does not work properly, it will cause additional cost for users. Moreover, the optimal VM placement is another important issue to be considered for increasing the resource usage in CC platform. These two problems of CC, TS and optimal VM locating, are actually mutually coupled, and should be considered together which can be shown in Fig. 2.

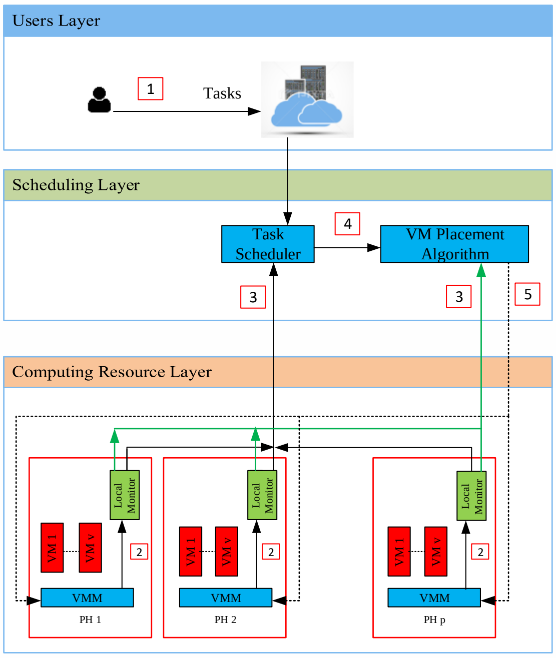

Fig. 2. TS and VM placement [13]

TS in CC is a bin-packing NP-Hard[7] problem which is become more hard-to-solve for complex CC platforms. In addition, VM placement has similar situation with regard to possible semi-optimal solutions. Consequently, in [13] these two problems are integrated to be solved as a c-optimization method using Metaheuristic Optimization Algorithm (MOA).

The time required for CC system to accomplish all tasks, or briefly, overall run time is called makespan [13]. But this definition is somehow theoretic and there is other practical definition of makespan which is the time period from the beginning position of a sequence of task to the final position. That is to say, makespan is an indicator of task scheduling effectiveness, meaning that lower value of makespan is associated with the fact that the scheduler is doing well and assigning tasks to devices almost perfectly, or at least, appropriately (for moderately low values of makespan); on the other hand, the high value of that is indicating the poor quality of scheduler planning process [17]. Therefore, with the mentioned definition in mind, the TS problem is studied in [17] using Antlion Optimizer Algorithm, AOA, by dividing the TS problem into two subparts A and B:

A. *How to reach efficient task distribution to HVMs in CC leveraging low-cost AOA?*

B. *How to reduce the time complexity of AOA for practical applications?*

In the context of CC and about the TS in CC, it is established that two effective algorithms with lower LB between clusters are Min-Min and Max-Min algorithms [2]. Actually, the former is able to decrease the completion time, but the main drawback of this algorithm is much load differences that it causes between machines and clusters. Therefore, the more efficient method in terms of LB, the later one was proposed to improve the Min-Min algorithm. Unfortunately, both of them perform weakly regarding long-standing islanding issues which is improved by KSF-Min algorithm in [2]. The abovementioned algorithms have overlooked the power consumption of the Cloud clusters during the TS process. With the expansion of the Cloud dimensions that it seems inevitable these days, the cost of energy consumed in Cloud clusters will experience a dramatic increase even by neglecting the user cost regarding hardware maintenance [19]. It might conversely affect the user satisfaction which is undesirable in a commercial point of view. That is to say, the following issues should be addressed by any newly-proposed algorithm:

1- *How to decrease average completion time by lowering the costs and energy consumption?*

2- *How to decrease average completion time while sustaining the efficiency?*

The clusters energy consumption quantification using the average completion time to establish a model relating these to parameters, and the analysis of rule comparisons are the solutions have been proposed in [2] to address two questions mentioned above, respectively. That is to say, the clusters' machines are investigated in terms of loss, reducing considerably the CC cost usage for users as well as the power issues by a low-power TS proposed in [2].

CC is essential concept for long cycle fast-response BD processing leading to long lasting repair routines.

[7] Non-Deterministic Polynomial-Time Hard

On the other hand, the other technology called edge computing or in a more sophisticated style named Fog computing is other way to provide demanded customer services rapidly (about 10ms and lower [20]) but locally. These two systems are suitable for highly concurrent applications such as local trade markets or any other kind of job that requires local network services. For those services that needed rapid response, the processing should be distributed in Fog computing nodes, that is to say, the providing the whole services only by Cloud platform is not able to meet the speed constraints, decreasing the user satisfaction of service. In this context, the task scheduling procedure should be transferred from Cloud platform to the Edge or Fog computing nodes as well. This shift regarding the processing method selection is accomplished by paying close and meticulous attention to power consumption and delay of service considerations. Therefore, the main research question of [20] was dedicated to possible ways for TS in large Cloud platforms including several Fog nodes [21].

III. Review of Algorithms For TS

The flexibility that CC provides in devices or resources makes it as effective as possible virtual apparatus in recent days to deal with growing computational requirements, notably for massive scale applications which include enormous number of tasks.

TS in CC can be categorized as the following:

1- heuristic algorithms, HA

2- metaheuristic algorithms, MHA

3- hybrid task scheduling algorithms, HTSA

TS using HAs (for example RR and SJF) assign tasks conveniently to reach high quality solutions, but there is no guarantee that the solutions are the best possible ones, and they sometimes come out with partial selections. On the other hand, MHAs which are the evolution of HAs, exploit erratic as well as local search algorithms capable of dealing with big data, developing the search area, and providing learning-based environment, tools, and strategies (such as GA, PSO, ACO) to find the optimum solutions [14], [22-25].

However, the solutions are sometimes so beforehand or are not infinitely optimal. To achieve faster convergence, the intersection and mutation operates are planned to cooperate in [20] to free the GA from local optimum solutions.

Noorian et al., presented a new task prioritization strategy and the application of task copy methods in order to solve the problem of scheduling dependent tasks in heterogeneous Cloud computing systems. The result of this research is a new list scheduling algorithm using methods for replicating related tasks. In the proposed algorithm, downward Optimistic Cost Table (OCTd) and upward Optimistic Cost Table (OCTu) methods are used to prioritize tasks in an efficient sorted list. In addition, the authors have used the fastest heterogeneous completion time method for duplicating tasks, which in return has reduced makespan effectively and efficiently [26].

The nature-inspired methods like MHA and ML have been used widely as energy efficient methods in recent years. But, if the main issue is not the energy efficiency, they will fail considerably to operate for fast data analysis [27]. These methods employed a kind of artificial intelligence, but this smartness has nothing to do with computation fastness, leading to higher execution time compared to deterministic methods [28-30]. The main issues regarding these methods are shown in Table 1.

The main drawback of neural networks is that if there are a lot of constraints for the algorithm, the architecture of ANN and its input and output layers are incapable of being expanded sufficiently. To overcome the obstacles mentioned above, in [28] ANN is employed to help scheduler independently. To this goal, algorithm do not bear any limitations from domain on its architecture, and, it examines both the size of received tasks and the current state of the Cloud structure, then allocates the best computing resources to the incoming size of tasks.

Table 1 The main issues of MHAs

MHAs, the main issues	
1-	MHAs are prone to higher execution time as a result of higher complicacy
2-	MHAs result in different solutions for the same problem, and in some cases, with longer execution time
3-	MHAs are not dynamic, which means they do not take into account the existing situation of Cloud and they are ignorant about natural alterations [31]

IV. Review of the Methods

The most important algorithms focused in this review are as follows:

Table 2 The major TS algorithms in this review

The Major TS algorithms of this review		
1	Genetic Algorithm (GA)	[24] [32] [33]
2	Particle Swarm Optimization (PSO)	[34] [35] [36]
3	Whale Optimization Algorithm (WOA)	[33]
4	Harris Hawks Optimization (HHO)	[37]

As the proposed method for TS in [38] is based on iHadoop parallelism of the computation, it could be applicable in BDPS. To find semi-optimal solutions the historical information of system nodes as well as the predicted resource requirements can be combined by the knowledge of system about capacities of the nodes, effectively increasing the pace of data processing. Allocating resources in an appropriate manner by virtual machines (VMs) is crucial to reduce computation overhead in Cloud environment, therefore,

[DOI: 10.52547/itrc.13.4.28]

in [33], a method using Map-Reduce framework and exploiting whale optimization algorithm is proposed to improve the quality of scheduling process, dividing the total tasks to set of subtasks using MRQFLDA [8] algorithm. This technique is of superior performance compared to other methods listed in [33].

MOTS [9] [11-12], [17-18], [38-39] method in conjunction with K-means algorithm ([40-41]) and LBS[10] are employed in [37] to produce the clusters of the task as an initial population for cluster optimization algorithm called DEA[11], decreasing makespan as well as LB. Actually, DEA is able to find globally-optimized solutions using much more potent strategy of linear methods which are combined by reinforced learning approach in [38] to reach less complex and more efficient solution in terms of parallelism and overall latency.

Since CC is assumed to be the main concept regarding distributed processing to execute SWFs [12] which need optimal usage of resources, each new proposal for the TS structures must bear excellent task allocation strategy [24]. A well-designed workflow management system, WMS, shown in Fig. 3, maps and manages dependent or independent tasks by taking into account the existing constraints with regard to shared resources. [32].

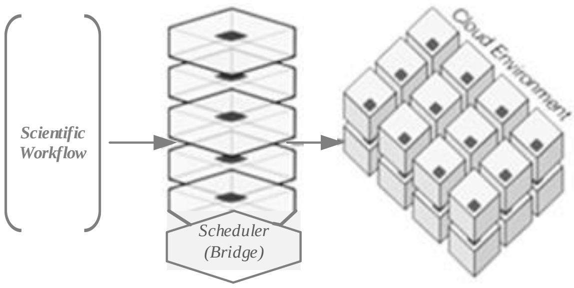

Fig. 3. SWF of execution architecture in CC [32]

Makespan in task scheduling is conversely affected by the higher communication overhead more than the execution time, therefore, the task clustering to integrate similar tasks meeting one task cluster's features is used in hybrid task scheduling algorithm proposed in [42] named HTSTC, considerably decreasing the TS makespan. This has led to decrease the concerns about the costs such as hardware maintenance or operating costs, encouraging the participants to use or create more and more applications [2] in Cloud environment. Consequently, the heterogeneous CC has been getting more attention owing to tons of physical hosts existing in large areas with vast variety of applications and performances including the different types of tasks and the different capacities regarding the storage or consumption power [2], [31].

The CC as a new concept was first introduced by Google and published in three fundamental papers[13][3]. Later, the academic as well as industrial researchers[14] have focused on this new revolutionary concept. The power for computation and storage are delivered over internet by a CC platform with an on-demand basis, executing user tasks satisfying QoS requirements by means of integrated VMs [36]. This decreases effectively the hardware physical components of a DC, and higher resource availability for participants using several VM hosted on CC [13], fulfilling the requirements of both users and providers.

Exploiting Catastrophic Genetic Algorithm (CGA), authors in reference [20] investigated the low delay method executing tasks on edge devices to reach an overall optimum performance. In this paper, fitness function is used to task run time quantification. Moreover, operators of mutation and crossover are optimized by improved roulette selection strategy.

By taking advantage of effective TS in distributed computational systems such as Cloud or Fog computing, the number of applications can be accomplished by CC systems, decreasing the computational overhead for the service customers [43-44]. This makes TS more popular, and on the other hand, more complex owing to the huge amount of the tasks and resources that involved as well as the constraints that must be considered for any proper design [33].

The method called ACO-CLA, combination of ant colony optimization and cellular learning automata, is introduced in [45] for TS in Mesh-Topology cluster computing. On the other hand, the low-latency optimal TS is examined thoroughly in [46] and [47] for Cloud-Fog environment.

To get high quality solutions in WOAs, Modified Henry Gas Solubility Optimization, HGSO, is employed in [29] in conjunction with comprehensive opposition-based learning, COBL, to gain optimum TS process.

ML-based optimization for Cost Effective Resource Scheduling (CERS) is studied in [14]. Using this strategy, the computation process is transferred to the edge of the network, and the computing infrastructure is also distributed in edge nodes, effectively decreasing the computation time [20], [25], [41]. This lower latency as well as improved task scheduling scheme is proven to be considerably effective for medical, smart home, intelligent traffic and transportation, environmental-friendly green applications that have distributed or local nature [48].

Owing to the rapid growth of the IOT devices, classic CC systems are not able to provide sufficient quickness in data transmission and the final efficiency of Edge and Fog is superior in comparison to classic CC in terms of latency, security issues, and communication bandwidth [49]. In this context, Cloud-Fog environments are able to improve the service quality by transforming some Cloud services from heavily loaded nodes to the edge of the network. This distribution of

[8] Maximized Raleigh Quotients Fisher's LDA
[9] Multi-Objective Task Scheduling
[10] Load Balancing Strategy
[11] Differential Evolutionary algorithm
[12] Scientific Workflows
[13] Google file system, Bigtable and MapReduce
[14] Amazon, Microsoft and the Apache Foundation

[DOI: 10.52547/itrc.13.4.28]

workload is beneficial for service quality but it has its own challenges regarding the selection of optimal nodes [46].

New load balancing algorithms have been proposed in order to efficiently allocate resources and schedule tasks, in addition, reduce response time in big data processing applications. Moreover, methods have been recommended to restrict the search area, resulting in the decrease of load balancing complexity in Cloud computing systems. Two mathematical optimization models for the dynamic allocation of resources to virtual machines and task scheduling are also introduced. Attempts have been made to minimize the execution time of tasks by reducing the idle time of the nodes [50 – 51].

Network failure in heterogeneous computing systems is inevitable due to the unprecedented growth of these systems in the last decade. Amini et al. have stated that network failure in Cloud systems will reduce system reliability. Hence, a reliability-aware task scheduling algorithm (RATSA) was presented to reduce failure rates. In RATSA, task scheduling is performed on directed acyclic diagrams using these evolutionary algorithms; Frog Jump and Genetic algorithm. In addition, the proposed RATSA algorithm uses a new technique for mapping tasks to the virtual machine in order to reduce failure rates. Experimental results show that this algorithm reduces the overall failure rate up to 43% compared to some of the current task scheduling algorithms [52].

V. Conclusion

In this paper, task scheduling algorithms are reviewed in recent works during 2020 – 2021 to find out the pros and cons of different methods with regard to fast data processing. The intricate nature of the issue requires comprehensive review/criticism of each method, that is to say, the dependency of different parameters together and the constraints that parameters must satisfy makes the evaluation more complex. In order to thoroughly analyze the methods, the main parameters and limitations of each method which is used recently for fast data processing are studied simultaneously to provide a realistic overview. The following methods for task scheduling are the most important ways to fast data processing proposed in the papers of this review's time frame (2020-2021):

1- The Combination of historical information of system nodes and the predicted resource requirements with the knowledge of system about capacities of the nodes

2- Dividing the total tasks to a set of subtasks using MRQFLDA exploiting map-reduce framework and whale optimization algorithm

3- MOTS method in conjunction with K-means algorithm accompanied by load balancing strategy to produce the clusters of tasks as an initial population for cluster optimization algorithm called Differential Evolutionary algorithm

4- Hybrid task scheduling algorithm, HTSTC, using task clustering to integrate similar tasks meeting one task cluster's features

5- Executing tasks on edge devices to reach an overall optimum performance exploiting Catastrophic Genetic Algorithm

6- Utilizing fitness function and optimized operators of mutation and crossover by roulette selection strategy to task run time quantification

7- Modified Henry Gas Solubility Optimization in conjunction with comprehensive opposition-based learning

8- Cost effective resource scheduling Machine Learning based optimization using computation process transfer and computing infrastructure distribution in edge nodes

References

[1] Abdel-Basset, Mohamed, et al. "EA-MSCA: An effective energy-aware multi-objective modified sine-cosine algorithm for real-time task scheduling in multiprocessor systems: Methods and analysis." Expert systems with applications 173 2021: pp. 114699.

[2] Liang, Bin, et al. "A low-power task scheduling algorithm for heterogeneous Cloud computing." The Journal of Supercomputing 2020: pp. 1-25.

[3] Kalia, Khushboo, and Neeraj Gupta. "Analysis of hadoop MapReduce scheduling in heterogeneous environment." Ain Shams Engineering Journal 12.1 2021: pp. 1101-1110.

[4] Hussain, Mehboob, et al. "Energy and performance-efficient task scheduling in heterogeneous virtualized Cloud computing." Sustainable Computing: Informatics and Systems 30 2021: pp. 100517.

[5] Sulaiman, Muhammad, et al. "A hybrid list-based task scheduling scheme for heterogeneous computing." The Journal of Supercomputing 2021: pp. 1-37.

[6] Wang, Bo, et al. "Security-aware task scheduling with deadline constraints on heterogeneous hybrid Clouds." Journal of Parallel and Distributed Computing 153 2021: pp. 15-28.

[7] Sulaiman, Muhammad, et al. "An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment." Journal of Grid Computing 19.1 2021: pp. 1-31.

[8] Deng, Zexi, et al. "Task scheduling on heterogeneous multiprocessor systems through coherent data allocation." Concurrency and Computation: Practice and Experience 33.10 2021: e6183.

[9] Silva, Eduardo Cassiano da. "Representações de Algoritmos Genéticos para o Problema de Escalonamento Estático de Tarefas em Multiprocessadores." 2020.

[10] da Silva, Eduardo C., and Paulo HR Gabriel. "A comprehensive review of evolutionary algorithms for multiprocessor DAG scheduling." Computation 8.2 2020: pp. 26.

[11] Ismayilov, Goshgar, and Haluk Rahmi Topcuoglu. "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in Cloud computing." Future Generation computer systems 102 2020: pp. 307-322.

[12] Abdel-Basset, Mohamed, et al. "EA-MSCA: An effective energy-aware multi-objective modified sine-cosine algorithm for real-time task scheduling in multiprocessor systems: Methods and analysis." Expert systems with applications 173 2021: pp. 114699.

[13] Alboaneen, Dabiah, et al. "A metaheuristic method for joint task scheduling and virtual machine placement in Cloud data centers." Future Generation Computer Systems 115 2021: pp. 201-212.

[14] Lohi, Shantanu A., et al. "Analysis and review of effectiveness of metaheuristics in task scheduling process with delineating machine learning as suitable alternative." 2020 International Conference on Innovative Trends in Information Technology (ICITIIT). IEEE, 2020.

[DOI: 10.52547/itrc.13.4.28]

[15] Chhabra, Amit, Gurvinder Singh, and Karanjeet Singh Kahlon. "Multi-criteria HPC task scheduling on IaaS Cloud infrastructures using meta-heuristics." Cluster Computing 24.2 2021: pp. 885-918.

[16] Houssein, Essam H., et al. "Task scheduling in Cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends." Swarm and Evolutionary Computation 2021: pp. 100841.

[17] Abualigah, Laith, and Ali Diabat. "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in Cloud computing environments." Cluster Computing 24.1 2021: pp. 205-223.

[18] Lavanya, M., B. Shanthi, and S. Saravanan. "Multi objective task scheduling algorithm based on SLA and processing time suitable for Cloud environment." Computer Communications 151, 2020: pp. 183-195.

[19] Medara, Rambabu, and Ravi Shankar Singh. "Energy efficient and reliability aware workflow task scheduling in Cloud environment." Wireless Personal Communications 2021: pp. 1-20.

[20] Wang, Shudong, et al. "A task scheduling strategy in edge-Cloud collaborative scenario based on deadline." Scientific Programming 2020

[21] Baniata, H., Anaqreh, A., & Kertesz, A. “PF-BTS: A Privacy-Aware Fog-enhanced Blockchain-assisted task scheduling” Information Processing & Management, vol. 58, no. 1, 2021

[22] Narayanan, Deepak, et al. "Heterogeneity-aware cluster scheduling policies for deep learning workloads." 14th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 20). 2020.

[23] Li, Jingbo, et al. "OKCM: improving parallel task scheduling in high-performance computing systems using online learning." The Journal of Supercomputing 77.6 2021: pp. 5960-5983.

[24] Asghari, Ali, Mohammad Karim Sohrabi, and Farzin Yaghmaee. "Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm." The Journal of Supercomputing 77.3 2021: pp. 2800-2828.

[25] Sheng, Shuran, et al. "Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing." Sensors 21.5 2021: 1666.

[26] Noorian, Hosseini and Motameni, “A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous Cloud computing platforms”, Journal of King Saud University- Computer and Information Sciences, doi.org/10.1016/j.jksuci.2021.05.011.

[27] Hajisami, Abolfazl, et al. "Elastic resource provisioning for increased energy efficiency and resource utilization in Cloud-RANs." Computer Networks 172, 2020: pp. 107170.

[28] Sharma, Mohan, and Ritu Garg. "An artificial neural network based approach for energy efficient task scheduling in Cloud data centers." Sustainable Computing: Informatics and Systems 26, 2020: 100373.

[29] Abd Elaziz, Mohamed, and Ibrahim Attiya. "An improved Henry gas solubility optimization algorithm for task scheduling in Cloud computing." Artificial Intelligence Review 54.5, 2021: pp. 3599-3637.

[30] Shneiderman, Ben. "Human-centered artificial intelligence: Reliable, safe & trustworthy." International Journal of Human–Computer Interaction 36.6, 2020: pp. 495-504.

[31] Praveenchandar, J., and A. Tamilarasi. "Dynamic resource allocation with optimized task scheduling and improved power management in Cloud computing." Journal of Ambient Intelligence and Humanized Computing 12.3, 2021: pp. 4147-4159.

[32] Aziza, Hatem, and Saoussen Krichen. "A hybrid genetic algorithm for scientific workflow scheduling in Cloud environment." Neural Computing & Applications 32.18 2020

[33] Sanaj, M. S., and PM Joe Prathap. "An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in Cloud computing environment." Materials Today: Proceedings 37, 2021: pp. 3199-3208.

[34] Alsaidy, Seema A., Amenah D. Abbood, and Mouayad A. Sahib. "Heuristic initialization of PSO task scheduling algorithm in Cloud computing." Journal of King Saud University-Computer and Information Sciences, 2020

[35] Balusamy, Jeevanantham, and Manivannan Karunakaran. "Hybridization of immune with particle swarm optimization in task scheduling on smart devices." Distributed and Parallel Databases, 2021: pp. 1-23.

[36] Jing, Weipeng, et al. "QoS-DPSO: QoS-aware Task Scheduling for Cloud Computing System." Journal of Network and Systems Management 29.1, 2021: pp. 1-29.

[37] Attiya, Ibrahim, Mohamed Abd Elaziz, and Shengwu Xiong. "Job scheduling in Cloud computing using a modified harris hawks optimization and simulated annealing algorithm." Computational intelligence and neuroscience 2020

[38] Jalalian, Zahra, and Mohsen Sharifi. "A hierarchical multi-objective task scheduling approach for fast big data processing." The Journal of Supercomputing, 2021: pp. 1-30.

[39] Jalalian, Zahra, and Mohsen Sharifi. "Autonomous Task Scheduling for Fast Big Data Processing." Big Data and HPC: Ecosystem and Convergence. IOS Press, 2018. Pp. 137-154.

[40] Mostafa, Samih M., and Hirofumi Amano. "Dynamic round robin CPU scheduling algorithm based on K-means clustering technique." Applied Sciences 10.15, 2020: 5134.

[41] Ullah, Ihsan, and Hee Yong Youn. "Task classification and scheduling based on K-means clustering for edge computing." Wireless Personal Communications 113.4, 2020: pp. 2611-2624.

[42] Tian, Qiao, et al. "A hybrid task scheduling algorithm based on task clustering." Mobile Networks and Applications 25.4, 2020: pp. 1518-1527.

[43] Li, Chunlin, et al. "Adaptive priority-based data placement and multi-task scheduling in geo-distributed Cloud systems." Knowledge-Based Systems 224, 2021: 107050.

[44] Javanmardi, Saeed, et al. "FUPE: A security driven task scheduling approach for SDN-based IoT–Fog networks." Journal of Information Security and Applications 60, 2021: 102853.

[45] Boveiri, Hamid Reza, Reza Javidan, and Raouf Khayami. "An intelligent hybrid approach for task scheduling in cluster computing environments as an infrastructure for biomedical applications." Expert Systems 38.1, 2021: e12536.

[46] Tsai, Jung-Fa, Chun-Hua Huang, and Ming-Hua Lin. "An Optimal Task Assignment Strategy in Cloud-Fog Computing Environment." Applied Sciences 11.4, 2021: 1909.

[47] Guevara, Judy C., and Nelson LS da Fonseca. "Task scheduling in Cloud-Fog computing systems." Peer-to-Peer Networking and Applications 14.2, 2021: pp. 962-977.

[48] Yuan, Haitao, et al. "Biobjective task scheduling for distributed green data centers." IEEE Transactions on Automation Science and Engineering 18.2, 2020: pp. 731-742.

[49] Ibrahim, Ibrahim Mahmood. "Task scheduling algorithms in Cloud computing: A review." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.4, 2021: pp. 1041-1053.

[50] Mirtaheri and Grandinetti,”Optimized load balancing in high-performance computing in big data analytics”, Concurrency Computat Pract Exper, doi: 10.1002/cpe.6265, 2021.

[51] Aghdashi and Mirtaheri, “Novel dynamic load balancing algorithm for Cloud-based big data analytics”, The Journal of Supercomputing, doi.org/10.1007/s11227-021-04024-8,2021.

[52] Amini, Movaghar and Rahmani, “ A new reliability-based task scheduling algorithm in Cloud computing”,International Journal of Communication Systems,doi.org/10.1002/dac.5022,2021.

APPENDIX I

List of Abbreviations

Abbreviation	Expansion
ACO	Ant Colony Optimization
ANN	Artificial Neural Networks
AOA	Antlion Optimizer Algorithm
BDPS	Big Data Processing Systems
CC	Cloud Computing
CERS	Cost Effective Resource Scheduling
CLA	Cellular Learning Automata
CN	Computing Node
CPTP	Cloud of Parallel Task Processing
CT	Completion Time
DAG	Directed Acyclic Graph
DC	Data Center
DEA	Differential Evolutionary Algorithm
DP	Data Processing
EA	Evolutionary Algorithm
ET	Execution Time
HMCP	Heterogeneous Multi-Core Processors
HTSTC	Hybrid Task Scheduling Algorithm Based on Task Clustering
LBS	Load Balancing Strategy
MA	Metaheuristic Algorithms
ML	Machine Learning
MOA	Metaheuristic Optimization Algorithm
MOTS	Multi-Objective Task Scheduling
MRQFLDA	Maximized Raleigh Quotients Fisher's LDA
MTSP	Multiprocessor Task Scheduling Problem
PP	Parallel Processing
QoS	Quality of Service
SLA	Service Level Agreement
SWF	Scientific Work Flows
TS	Task Scheduling
VM	Virtual Machines
WMS	Workflow Management System

Zahra Jalalian received her B.Sc. degree in Computer Engineering from Iran University of Science and Technology (IUST), and M.Sc. degree in Computer Science from Concordia University, Canada. She is currently an instructor at Kharazmi University, Iran since 2006. Presently, she is a Ph.D. candidate in Computer Engineering at IUST. Her research interests include Big Data, Distributed Systems, Programming Languages, and Information Retrieval.

Mohsen Sharifi is a Professor of System Software Engineering currently at the School of Computer Engineering of Iran University of Science and Technology. He received his B.Sc., M.Sc., and Ph.D. in Computer Science from the University of Manchester in the United Kingdom. He directs a Distributed Systems research group and laboratory. His main interest is in the Development of Distributed Systems, Solutions, and Applications, Particularly for use in Various Fields of Science.

[DOI: 10.52547/itrc.13.4.28]