Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

A Survey on Multiple Workflow Scheduling Algorithms in Cloud Environment

Zaid Hanoosh[1]

[1] *University of Kufa, Kufa, Iraq.* Zaidm.alrazaqhnoosh@uokufa.edu.iq

https://doi.org/10.46649/fjiece.v3.1.4a.13.4.2024

Abstract. *The workflow approach is a standard for displaying processes and their implementation process. With the advent of electrical sciences, more cumbersome workflows were created with more processing requirements. New distributed systems, such as grid systems and computing clouds, allow users to access heterogeneous resources that are geographically located at different points and execute their workflow tasks. Therefore, the simultaneous receipt and execution of several workflows is obvious. As a result of discussing scheduling algorithms, it is necessary to consider arrangements for implementing multiple workflows on a shared resource set. Improving the execution of multiple workflows can accelerate the process of obtaining results when sending processes to the cloud. In this paper, we first discuss the classification of multiple workflow scheduling algorithms, and then briefly describe the scheduling algorithms for the cloud environment, and finally, the algorithms of papers were compared with each other.*

Keywords: *Cloud Computing, Multiple Workflows, Scheduling, Grid Computing.*

1. INTRODUCTION

For providing the tasks set which are interrelated, also their distribution between various accessible resources to the optimal scheduling of resource, the workflow is being used [1]. Nowadays, the workflows are utilized frequently for shaping great scientific applications of scale distributed [2]. They make easy hybrid-stage task workloads statement in the behavior that is simple for realizing, retain, debug. With utilizing the scientific workflows hybrid investigators are able to collude on planning the unique application which is distributed. It is due to that the workflows are regulated as the directed acyclic graphs (DAGs), which every node is the standalone task, also the edges provide some dependencies among the tasks. Such dependencies are normally the files which are output or input and require to be transmitted among the tasks. By increasing the workflows popularity in a society which is scientific, allocated the systems of execution management have appeared for presenting the environment of dedicated execution. Like, Pegasus [3], which is utilized in scientific areas amount, for example bioinformatics, astronomy is the system which is able to perform the workflows on the clusters, desktops, clouds, also grids. This execution is the total tasks, particularly on the clouds, that a resource providing, de-providing, calculating the cost, resource setup should be considered. Several applications which are known, utilize workflow meaning, contain: Chimera, LIGO, Montage, CSTEM, AIRSN.

Against a lot of tasks which have been performed in the cases of scheduling of workflow, multiple workflows scheduling is the problem which is challenging, just main researches have been performed in the domain [4, 5]. Developing the multiple workflows execution is able to quicken achieving conclusions process while sending the progresses toward cloud.

Prevalent workflow scheduling algorithm aim is decreasing length of schedule (makespan). Length of schedule is straightly liable for workflow execution time. While the multiple workflows divide similar environment of execution, by the side of makespan minimization, here are some contrasts that should be watched, monitored for guaranteeing efficiency of system. Like, the way of scheduling workflows like this, no one will be specialized on resources by losses while comparing to another one in the cases of execution time. When here are the multiple workflows for being scheduled, also an algorithm takes into account just once, primary one took into account will be useful, so the makespan of it will be decreased with the larger number than the last workflow makespan for being scheduled. So, an algorithm which is scheduling must takes into account the goodness for dividing resources equal between arriving workflows [6].

2. CATEGORIZATION OF MULTIPLE WORKFLOW SCHEDULING ALGORITHMS

Various algorithms are presented for simultaneous scheduling of multiple workflows in distributed systems, including grid and cloud systems, and each of these algorithms considers different quality of service criteria. In addition to the quality of service parameters in grid systems (including reduced run-time, increased utilization, increased reliability, availability, and increased fairness), the cost parameter is also considered in cloud systems.

Bittencourt et al. [6] suggested three general approaches to the problem of scheduling multiple workflows:

- Schedule workflows independently.
- Schedule workflows interleaving.
- Merge of workflow graphs in a graph and Scheduling of the integrated graph.

For scheduling the multiple workflows, we suggest that, at the given time, we have some N workflows tasks for being scheduled. Mention that it does not essentially concept that we require to schedule each workflows tasks at scheduling task beginning time. While one/more workflows receive, we take into account whole of the workflows tasks which are non-executed which received before. When we just take into account the workflows which only received, we won't take advantage of connection times which are left by workflows recently scheduled, however, not performed so far. In another word, when we wait for the multiple workflows to receive, it might delay first workflow execution. Therefore, if the workflow receives, we take into account the workflow tasks which are non-executed which received before as the workflows for being re-scheduled. This procedure, novel workflow is combined by workflows recently scheduled for taking advantage of the connection times. That workflows, tasks would be taken into account for the re-scheduling is explained with middleware, which might determine this based on the how much time is left for every workflow for being complete, for example.

According to category provided by Bittencourt, his coworkers, to every category, they also define the algorithm to the grid systems. First, we explain such algorithms at every category, then we suggest the algorithms of multiple workflow scheduling to the cloud.

2.1. *Schedule workflows independently*

In this categorization, workflows are scheduled and run independently and in turn, this means that after completion of the current workflow scheduling (scheduling all the tasks of a workflow on resources), it turns into the next workflow.

Bittencourt et al. provided two sequential algorithms and Gap search in grid systems, both of which used a path clustering heuristic (PCH) method to independently schedule each workflow. With the difference that the second algorithm attempts to schedule a task from the workflow to use the gaps of empty in the resources (gaps created by scheduling the tasks of previous workflows).

Path Clustering Heuristic is the scheduling heuristic of DAG that utilizes procedure of clustering for producing tasks set (clusters), the list scheduling procedure for choosing processors, also tasks. Sets ways of PCH of DAG, schedules them originally on similar resource, by decreasing connection costs aim. A basis for selecting resource for every cluster is to decrease the EFT of that cluster.

Varalakshm et al. [7] by provisioning an Optimal Workflow based Scheduling (OWS) algorithm, an aim was to find the answer which faces the parameters of user-preferred Quality of Service (QoS). Work attracts the attentions on scheduling cloud workflows. Firstly, an algorithm of Resource detection, regulates whole resources, it aids in placing free resources. In this chapter utilizes suggested observing tree-based architecture to observe the resources of cloud. Every resource sends the information of itself to the urgent parent node of itself (the resources are arranged in the manner which is hierarchical). This procedure root node usually retains the whole resources list, so this is simple for polling root for asking to each information according to resources. Therefore, the query of client will be arrived with root node. A root node considers whether here is matching resource between the cluster nodes of itself. Then, root node transmits the ask of itself to just the head of corresponding cluster. This procedure they avoid the flooding of information. A root node remains whole updated compute nodes index. On the other hand, tree leaves remain whole "actual" resources which execute job essential execution, computation. Servers of cluster head as the middle node executing the root node, compute node functionalities. Nodes need special daemons for being run in them. One time observing information has been achieved, next stage is separating resources in order to different QoS (reliability, time, and cost) needs.

The computes scheduler needed some parameters such as earliest start time(est), earliest completion time (ect), favorite predecessor (fpred), favorite processor (FAP). According to such parameters, clusters of sub-task are produced of workflow. Then, from observing the information, the resources are separated according to parameter of client-desired QoS viz., cost, reliability, time. Then, the algorithm of OWS scheduling takes the client specialized parameter of QoS as the main agents. Given to the different dependencies of data, control between some of the sub-tasks, clusters are produced of this job. Then, match creating progress is performed. According to guarantee the goodness, they decrease the Processor Fairness Value (PFV) accompanied by every resource, so that previous tasks are not absorbed to similar potent resource. Utilizing heuristics of QoS (which protect different parameters of client-specified QoS) they determine sub-task cluster to optimal resource of itself. In term sufficient resources are not accessible for the job, then density is executed, i.e., whole empty spaces are gather to each other for making the virtual disk(s), then leaving the jobs are performed here. So, clusters are regulated in the CP ascending order that is explained as the execution times sum of whole sub-tasks determined to the processor. This progress is performed until we obtain needed free processors amount.

*2.2. **Schedule workflows interleaving***

In this type of algorithms, the method of interleaving is used to schedule the tasks of the workflow so that a number of scheduling tasks are selected at each turn of each workflow.

Bittencourt et al. [6] provided a static algorithm to solve the simultaneous scheduling problem of multiple workflows in grid systems. Quality of service criteria are in this algorithm, time and fairness. This algorithm selects a cluster of tasks for scheduling at each turn of each workflow (in order of logging), and uses clustering for the path to exploration for clustering. This routine continues until all clusters are executed from all workflows. The policy of prioritizing the ready-made clusters from each workflow to implement is the FCFS (First Come, First Served) priority.

Scheduling algorithm performs resource gap search for better use of existing resources. The approach of interleaving allows you to use the spaces created for sending data to process other workflows.

Xu et al. [8] considering the time and cost as quality of service criteria, provided a dynamic scheduling algorithm called Multiple QoS Constrained Scheduling Strategy of Multi-Workflows (MQMW) for the cloud computing infrastructure. The goal of this algorithm is to reduce the average running time and the cost of workflow in a competitive environment.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Due to that clients first provide workflow by the QoS needs of them. Then the system specializes suitable services to the progressing tasks of workflow, schedules tasks on services in order to needs of QoS, also the environment of cloud.

Given to schedule workflow actively, optimize decision of resource division, a system that they offered includes three core elements: Executor, Preprocessor, and Scheduler. Preprocessor computes four ready tasks attributes: accessible number of service, covariance for the cost, time, share of time and cost. Furthermore, Preprocessor computes workflow cost, time extra. Then, this determines ready tasks Scheduler queue that is the arranged group including whole of the tasks of various clients who are waiting for being scheduled. Then, Scheduler re-computes above tasks attributes in queue, then re-arrange whole of the tasks in queue in order to strategy that will be argued as follow. Executor chooses optimal service for continuous performing tasks in queue. While the task completes, Executor informs Preprocessor that task appertain to situation of completion. Competition performed with continued, active happen initialed connection between the elements of core. 1) Submission of Workflow: while the novel workflow receives, this is determined to Preprocessor. Then, Preprocessor computes whole of the ready tasks attributes. 2) Preprocessing: After computing whole of the ready tasks attributes in workflow, Preprocessor appends ready tasks in queue. At the first time, just the entrance tasks will be determined. Then, above the news with task Executor completion, Preprocessor will distinguish when each substitute tasks become prepare, determine them. Attributes of task information is determined along with task. 3) Task scheduling: Whenever there are services accessible, the task is waiting in queue, Scheduler will re-compute whole the tasks recently ready in queue, arrange whole the tasks, then perform again: a) eliminate first task in queue; b) Allocate task to a service that is optimal suited; c) append task in next circle queue when there are not services that are not able to complete task. 4) Task completion notification: if the task successfully completes, Executor will inform a task completion situation Preprocessor.

Duan and his colleagues [9] considered scheduling multiple large-scale applications issue on the clouds which are hybrid. The scheduling multiple large-scale parallel workflow applications on the heterogeneous computing systems such as hybrid clouds is the fundamental NP-complete issue which is critical for facing QoS different kinds (Quality of Service) needs. Also they map large-scale applications scheduling problem suggested of the real-world, specialized with the colleague, symmetrical task bags large amount which are basis narrow resources, however, good potential which is open for the optimization. Problem of scheduling is formed as the novel continued the game that is cooperative, also offer the connection, multi objective algorithm of the storage-aware which optimizes two objectives of user (economic cost, execution time) when it fulfilling two restrictions (storage needs, network bandwidth).

The applications which are large-scale are studied as the tasks that are independent great number and also they are linked via the dependencies of control and data. This algorithm concentrates on the large-scale workflows specialized with the (thousands to millions) colleague parallel (independent) tasks high amount which influence the execution, interconnected via dependencies of control and data flow.

In this issue, there are the n applications group (not considering the entry times of them) including the tasks which are able to be grouped in K various BoTs, the environment of cloud includes M places. An application makespan is its BoTs maximum completion time. Multi-objective scheduling problem aim is finding an answer which assigns whole of the tasks to for places like that makespan, whole of the applications economic cost are reduced, bandwidth, storage needs are fulfilled.

The algorithm studies three main parameters: strategies, players, payoff specification. In this algorithm, first the game studies the cooperative game of K-player that in every managers of K application (like players) tries at the sure time instances to reduce a BoT execution time regarded to the whole tasks amount of it, also the rate of progressing of it in every place. Clearly, use that every manager of application manages a BoT execution. Every managers aims, are reducing economic cost, execution time of the BoTs when fulfilling storage, bandwidth restrictions. And use that clients just pay for the optimal computation, so, cost is independent on processors used number.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

The game theory which is cooperative is worried by conditions while player groups coordinate the acts of themselves that is most important algorithmic mechanism which creates the games have “transferable profit”. On the other hand, the player by raised profit has this skill for making up the several players by reduced profit.

A group from Hunan University presented two algorithms for multiple workflows scheduling. The first one was the Fairness-based Dynamic Multiple Heterogeneous Selection Value (F_DMHSV) [14]. The algorithm consisted of six steps which were task prioritization, task selection, task allocation, task scheduling, the arrival of new workflow handling, and task monitoring. The task prioritization used a descending order of heterogeneous priority rank value (HPRV), which included the out-degree (i.e., number of successors) of the task. The task was selected from the ready tasks pool based on the maximum HPRV. Furthermore, the task was allocated to the processor with minimum heterogeneous selection value (HSV) that optimized the task allocation criteria using the combination of upward and downward rank. The task, then, was scheduled to the earliest available processor with minimum HSV.

*2.3. **Merge of workflow graphs***

In this type of algorithms before the start scheduling, the workflow graphs are merged together and placed within a workflow graph.. Then the resulting workflow of graphs is scheduled.
Bittencourt et al. [6] provided a static algorithm to solve the scheduling problem of multiple workflows in grid systems. Their goal is to reduce the running time of the workflow so that the total time is minimized and provides fairness. At this algorithm, first according to [10] DAG node is known the entry task (tentry), final node is known as the exit task (texit). Whole of DAGs have just one entry, also one exit task. When the DAG has more than one entry/more than one exit, one entry task, one exit task, the two of them by zero cost, are added to graph, by the edges which are useless communication them to main entry or exit tasks. Then, the last workflow graph is scheduled by utilizing method of PCH.
Sharif and colleagues [11] provide 2 online algorithms for scheduling the multiple workflows below the restrictions of deadline, privacy, when studying hybrid cloud environment active aspect. In this article, the resources of cloud are formed as same as the Amazon’s Elastic Compute Cloud (EC2) by instances which are billed hourly.
At the model, private, public cloud represent the different computation services by the specification which is individual like CPU kind, cost, and privacy access (described in privacy part). At the model of us, the resource of the private cloud by specification which is same is billed less than the public resource because of SaaS (software as a service) model among important action, also the users of it. It concepts that flowing the workflow application in the resources of public cloud would cost additional in the term of outsourcing client’s workflows for the resources which are public.
At this work, privacy constraint is related to the elements of workflow, i.e., tasks, data. Private tasks capability must not be disclosed to clients with no needed access privilege. There are 3 privacy privilege levels to every task of workflow. Which states that the task is able to be developed on the two resources of public, private with no constraints. Also, the resources are determined by the privacy tags like public, private.
Proposed algorithms are (OMPHC-PCPR, OPHC-TR) which deployed to the actively schedule multiple workflows on the environments of HC studying the privacy, deadline restrictions of them. The OMPHC-PCPR first separates the input DAG in order to predefined privacy levels of its tasks, then locates some critical ways to every sub-DAG. Whole of the ways are queued, then scheduled on suitable resources. The OPHC-TR, queued whole of the input DAG tasks, then addresses them to chosen resources in order to the privacy of them. They also suppose that such algorithms are performed in the different scheduling intervals. At every scheduling interval, whole arrival workflows are gathered at the parallel fashion for producing the

solid single DAG for feeding scheduler. Some of the DAGs parallel aggregation occurs with adding the dummy-start node, joining this to whole of start nodes of DAG and also adding the dummy-exit node to whole of the exit nodes of DAGs.

OMPHC-PCPR input algorithms are the combined DAG containing some workflows, the related deadlines. At the first of every scheduling interval, this algorithm is required, also it begins with knowing controlling way.

Executing multi-terminal cut algorithm for separating input DAG in k sub-graphs that k is privacy levels number according to the part of privacy model. This causes in 3 sub-graphs including privacy levels. Then, tasks scheduling attributes are computed in the stage 6. Every task in the DAG has 3 scheduling attributes: (1) Latest Finish Time (LFT), (2) Earliest Finish Time (EFT), (3) Earliest Start Time (EST), (3) whole partial critical paths (PCP) and the critical path (CP) of every sub-graph are achieved.

PCP is distinguished with the determined, also the critical parent:

Whole of the sub-graphs PCs, PCPs are ranked in order to the rows of them. Whole of the ways queued are determined in order to the hybrid cloud used resources. On the other hand, the PC/PCP is de-ranked, whole way is scheduled on cheapest resource instance which is able to complete whole tasks in way before the latest complete times. Here are sufficient free gaps in the recently leased instances for executing whole the given way tasks before recent task LFT in way. When no one of such situations are willing the novel instance of cheapest applicable resource is made by attention to the way privacy privilege.

This algorithm, online scheduling to the Privacy in Hybrid Clouds using Task Ranking (OPHC-TR), schedules input DAG variously of OMPHCPCPR with scheduling tasks separately (for the total way). The practicable instance to the task is chosen first with addressing a task privacy level for suitable resources, then examined that whether the task is able to be performed on that instance before the latest finish time (LFT) of it. The OMPHC-PCPR out-performs OPHC-TR with reducing cost with 50 percent.

Liu et al. [12] proposed the time dependence related to the strategy of scheduling to the multiple workflows. This strategy studies every workflow special structure, assess its preference of it in order to the indictors of it, also the relationship of it by the other workflows, and throw away the workflows part by the low preference. Proposed strategy is able to decrease throwing workflows rate, develop the completion percentage of workflow, also usages of resource when consenting budget, deadline restriction. The strategy of time-dep will separate whole of the DAGs with the preference of them. The strategy of time-dep chose the DAGs part that is able to be finished in budget, also allocate such DAGs in the set of sDAG, then combines DAGs into the sDAG in the large DAG, and utilize the algorithm of way cut scheduling to schedule the whole task. This procedure utilizes of Amazon EC2 pricing model, also simulator of Cloudsim.

Arabnejad et al. [13] here explain the novel strategy of scheduling, Multi-QoS Profit-Aware scheduling algorithm (MQ-PAS), to schedule the concomitant workflow applications by the multiple QoS restrictions, here, cost, time. Algorithm of MQ-PAS includes two basic stages: firstly, it chooses the task of every prepare workflow, also determine the preference for every task according to maintaining time to the application deadline, accessible budget. Secondly, to higher preference task, the MQ-PAS chooses the appropriate resource according to the quality measure computed to every resource. Proposed algorithm tries for increasing provider revenue with regarding to jobs budget in progress of decision. Totally, in the most on-line scheduling systems, with no in-advance storing, scheduler is known while the executing task completes, here is at least one free processor that is accessible. As the other online schedulers, MQ-PAS includes the two basic phases, the phase of processor selection, and the phase of task selection.

Algorithm of MQ-PAS must choose the appropriate task to be performed between the whole tasks of present tasks set, that is filled with the present tasks belong to the every provided, not completed application of workflow. Generally, two procedures are utilized to fill present tasks set: a) first gather one present task of every workflow, b) append whole of the present tasks belong to every not completed application of workflow. In this article, authors propose the novel strategy (rankD) for determining the secondary preference for every task belong to the workflow in present tasks set. Rank D preference amount includes two main agents: a) parameter of cost b) parameter of time.

Phase of processor selection has responsibility to choose the resource that is affordable to recent task, also it is repeated till any more tasks maintain in present tasks set. The novel strategy to phase of processor selection according to the QoS needs is suggested. For maintaining used time, cost, the limit amount to every agent is required. A provided algorithm has the complexity of low time, creating this appropriate to use in true infrastructures which are heterogeneous.

Group of Hunan University published energy-efficient algorithms for multiple workflows scheduling, which combined the Deadline-driven Processor Merging for Multiple Workflow (DPMMW) that aimed to meet the deadline, and the Global Energy Saving for Multiple Workflows (GESMW) aimed to lower the energy consumption [15]. DPMMW was a clustering algorithm which allocated the clustered tasks in a minimum number of processors, so the algorithm can put idle processors into sleep mode. Meanwhile, GESMW reassigned and adjusted the tasks to any processor with minimum energy consumption in the global scope. The combination of DPMMW&GESW was exploited to get a lower energy consumption. This approach was different from the previous two energy-efficient algorithms that focused on virtual machine level manipulation. This group presented two opposite approaches to a scheduling with different objectives. However, in both approaches, the works emphasize on a similar strategy of the resource selection. In their first work, the algorithm focuses on selecting various resources to minimize the makespan, while in the second one, it is selecting different machine with various energy efficiency to minimize the energy consumption. These resource selection strategies can improve the scheduling result by combining them with efficient task scheduling approaches. In Table 1, these algorithms are compared.

Table 1. Compare Multiple Workflows Scheduling Algorithms

Author and year	Compare with other algorithms	Categorization of algorithms	Simulation environment	constraints	Goals	The name of the algorithm
Xu Meng et al (2009)	RANK_HYBD Yu & Shi (2008)	interleaving	An experimental simulator	Cost and time	Decrease makspan	MQMW
Bittencourt et al. (2010)	Between their algorithms	Interleaving and Merge method and Independent	Grid shared environment	schedule length and fairness	good average makespan and provides fairness	Sequential, Gap search, Interleave, Group
Varalakshmi et al (2011)	FCFS and backfilling	Independent	Open Nebula	execution time, reliability	improvement in CPU utilization	OWS
Sharif et all (2014)	Between their algorithms	Merge method	hybrid cloud	Deadlines and privacy	Decrease cost	OMPHC-PCPR and

						OPHC-TR
Duan et al. (2014)	(G-Min-min), (G-max-min, G-sufferage, G-MCT, G-OLB, (G-MET	interleaving	hybrid cloud	bandwidth and storage requirements	Decrease makespan and economic cost	GMO
Liu et al (2016)	MDW-W, HEFT	Merge method	Cloudsim	budget and deadline	maximize the throughput and Resource efficiency	Time-dep
Xie et al. (2017a)	RANK_HYBD, OWM, and FDWS	interleaving	programmed in Java to generate a variety of test workflows	timing-constraints	reduce the overall scheduling length and the deadlines of workflows	F_DMHSV
Xie et al. (2017b)	DEWTS, DPMMW&EESMW	Merge method	simulated CPCS	energy management and real-time constraint	minimize the makespan, minimize the energy consumption	DPMMW&GESMW
Arabnejad et al. (2018)	MAX-MIN*, MIN-MIN*, FDWS2	Merge method	SimGrid toolkit	budget and deadline	increases provider revenue	MQ-PAS

3. CONCLUSIONS

When an operating environment and resources are shared between multiple workflows. Scheduling algorithms should consider more arrangements than scheduling a workflow. The key goals in this area are: achieving quality of service criteria, deadlines, reduce the completion time and cost of multiple workflows, allocate appropriate tasks to processors, and fairness in the time and cost of implementing the workflow. On the other hand, the challenges can be expressed as follows. 1) Achieving the desirable customer quality of service, (2) Time (minimizing overall runtime and minimizing the average runtime for any workflow is inevitable, reducing it is challenging, 3) Cost (proper use of the time of slack in the resources that cost are paid for them and considering the time intervals in cloud computing causes the problem to be hardened), 4) solving the problem of fairness; 5) improving the problem of allocating tasks to processors (the sub problem that needs to be solved repeatedly and its cascade error down to the overall problem.
In this research, after reviewing the classification of multiple workflow scheduling algorithms, several examples were presented in the cloud, and the articles were compared together.

REFERENCES

[1] Singh, Sukhpal, and Inderveer Chana. "A survey on resource scheduling in cloud computing: Issues and challenges." Journal of grid computing 14, no. 2 (2016): 217-264.

[2] Bryk, Piotr, Maciej Malawski, Gideon Juve, and Ewa Deelman. "Storage-aware algorithms for scheduling of workflow ensembles in clouds." Journal of Grid Computing 14, no. 2 (2016): 359-378.

[3] Danelutto, Marco, Massimo Torquati, and Peter Kilpatrick. "A green perspective on structured parallel programming." In Parallel, Distributed and Network-Based Processing (PDP), 2015 23rd Euromicro International Conference on, pp. 430-437. IEEE, 2015.

[4] Bittencourt, Luiz F., and Edmundo RM Madeira. "Fulfilling task dependence gaps for workflow scheduling on grids." In Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp. 468-475. IEEE, 2007.

[5] Zhao, Henan, and Rizos Sakellariou. "Scheduling multiple DAGs onto heterogeneous systems." In Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, pp. 14-pp. IEEE, 2006.

[6] Bittencourt, Luiz Fernando, and Edmundo RM Madeira. "Towards the scheduling of multiple workflows on computational grids." Journal of grid computing 8, no. 3 (2010): 419-441.

[7] Varalakshmi, P., Aravindh Ramaswamy, Aswath Balasubramanian, and Palaniappan Vijaykumar. "An optimal workflow based scheduling and resource allocation in cloud." In International Conference on Advances in Computing and Communications, pp. 411-420. Springer, Berlin, Heidelberg, 2011.

[8] Xu, Meng, Lizhen Cui, Haiyang Wang, and Yanbing Bi. "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing." In Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on, pp. 629-634. IEEE, 2009.

[9] Duan, Rubing, Radu Prodan, and Xiaorong Li. "Multi-objective game theoretic schedulingof bag-of-tasks workflows on hybrid clouds." IEEE Transactions on Cloud Computing 2, no. 1 (2014): 29-42.

[10] Zhao, Henan, and Rizos Sakellariou. "Scheduling multiple DAGs onto heterogeneous systems." In Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, pp. 14-pp. IEEE, 2006.

[11] Sharif, Shaghayegh, Javid Taheri, Albert Y. Zomaya, and Surya Nepal. "Online multiple workflow scheduling under privacy and deadline in hybrid cloud environment." In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on, pp. 455-462. IEEE, 2014.

[12] Liu, Shaowei, Kaijun Ren, Kefeng Deng, and Junqiang Song. "Time dependence based scheduling strategy for multiple workflows on IaaS cloud platform." In Computer, Consumer and Control (IS3C), 2016 International Symposium on, pp. 784-788. IEEE, 2016.

[13] Arabnejad, Hamid, and Jorge G. Barbosa. "Reprint of "Multi-QoS constrained and Profit-aware scheduling approach for concurrent workflows on heterogeneous systems"." Future Generation Computer Systems 78 (2018): 402-412.

[14] Xie, Guoqi, Liangjiao Liu, Liu Yang, and Renfa Li. "Scheduling trade-off of dynamic multiple parallel workflows on heterogeneous distributed computing systems." Concurrency Comput.-Parctice Experience 29, no. 8 (2017): 1-18.

[15] Xie, Guoqi, Gang Zeng, Junqiang Jiang, Chunnian Fan, Renfa Li, and Keqin Li. "Energy management for multiple real-time workflows on cyber–physical cloud systems." Future Generation Computer Systems (2017).