



# **A Comparative Study on Various Scheduling Algorithms in Cloud Environment**

P Kowsik<sup>1</sup>, K.Rajakumari<sup>2</sup>

PG Scholar, Department of Information Technology, SNS College of Technology, Coimbatore, TamilNadu, India<sup>1</sup>.

Associate Professor, Department of Information Technology, SNS College of Technology, Coimbatore, TamilNadu, India<sup>2</sup>.

**ABSTRACT:** Cloud computing is an emerging technology and it allows users to pay as you need and has the high performance. Cloud computing is a heterogeneous system as well and it holds large amount of application data. In the process of scheduling some intensive data or computing an intensive application, it is acknowledged that optimizing the transferring and processing time is crucial to an application program. Due to novelty of cloud computing field, there is no many standard task scheduling algorithm used in cloud environment. Especially that in cloud, there is a high communication cost that prevents well known task schedulers to be applied in large scale distributed environment. Today, researchers attempt to build job scheduling algorithms that are compatible and applicable in Cloud Computing environment Job scheduling is most important task in cloud computing environment because user have to pay for resources used based upon time. Hence efficient utilization of resources must be important and for that scheduling plays a vital role to get maximum benefit from the resources. In this paper we have surveyed different types of scheduling algorithms and tabulated their various parameters, scheduling factors and so on. Existing workflow scheduling algorithms does not consider reliability and availability. In this paper presents a novel heuristic scheduling algorithm, called hyper-heuristic scheduling algorithm (HHSA), to find better scheduling solutions for cloud computing systems. The results show that HHSA can significantly reduce the makespan of task scheduling compared with the other scheduling algorithms.

**KEYWORDS:** Cloud Computing, Particle Swarm, Optimization, Simulated Annealing, Improved Particle, Swarm Optimization, Proposed Algorithm, scheduling, Heuristic Algorithm.

## **I. INTRODUCTION**

Cloud computing technology is a new paradigm for utility virtualized resources, designed for end users in a dynamic computing environment to provide reliable and guaranteed services [1]. In this environment, on-demand services of computing intensive applications can be increased by user requests. According to the number of users rapidly increasing, the virtualization is one of main techniques to improve the utilization of physical re-sources in cloud environments [2]. It allows abstraction and isolation of underlying physical resource and reduces the number of hardware equipment. These virtualization techniques include network virtualization and allow the operator to create several Virtual Machines (VMs) on a single physical server. Specifically, VMs can be designed by increasing or decreasing the CPU power and/or the number of CPUs [3]. In order to efficiently utilize the virtualized resources to execute computing-intensive application, the effective meta-job scheduling algorithms are needed. The traditional job scheduling problem schedule meta-job of applications across computation resources in order to reduce the jobs completed time while ignoring the specific shared nature of the network resource [4].

In a supercomputer datacenter, the scheduling problem is enhanced by scheduling a set of applications from different users to the set of computation resources while maximizing system utilization. Previous researchers have been conducted in this area leading to an extensive study of the major theoretical and practical results [5]. However, with the emergence of the virtualization techniques in computational cloud systems, new scheduling algorithms are required to deal with concerns originating from the cloud infrastructure. In cloud computing, the objective of job scheduling algorithm is to achieve high system throughput, improve the load balance and minimizing the meta-job total processing time while matching the meta-job requirements with available virtualized resources. Job scheduling is a general



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

problem of mapping a set of jobs to a set of VMs to fulfil the user's requests within cloud environment. The objective of job scheduling is to achieve high system throughput and minimizing the meta-job total processing time while matching the meta-job requirements with available virtualized re-sources.

The scheduler in this environment needs to consider virtualized resources and user's required constraint to get better match between applications and resources. When users consider a variety of network resources related to the quality of service, job scheduling becomes more complicated in cloud computing. Many traditional heuristics scheduling algorithms such as Minimum Completion Time (MCT), minimum-Minimum Completion Time (Min-Min), Maximum-Minimum Completion Time (Max-Min) does not consider the network impact of cloud applications. We present a V-heuristic scheduling algorithm to address the match of jobs from the application and the status provided by the diverse virtualized resources in cloud computing. Consequently, the scheduling algorithm improved the efficiency and the utilization of a cloud system. To provide this service, cloud service provider (CSP) deploys virtual network from infrastructure provider (InP) and one or multiple VMs on single physical machine that are coordinated together.

Within service provider, heterogeneous physical networks are abstracted by using virtualization technologies which can provide efficient data transmission among of VMs in cloud computing. We can provide hybrid heuristic scheduling algorithm for optimal virtualized resources utilization, maximum throughput, minimize the job processing time, load balancing over VMs and satisfy on-demand services of user requests.

## II. RELATED WORK

Optimization problems are in Class NP-hard. These problems can be solved by enumeration method, heuristic method or approximation method. In enumeration method, an optimal solution can be selected if all the possible solutions are enumerated and compared one by one. When number of instances is large, exhaustive enumeration is not feasible for scheduling problems. In that case heuristic is a suboptimal algorithm to find reasonably good solutions reasonably fast. Approximation algorithms are used to find approximate solutions to optimized solution. These algorithms are used for problems when exact polynomial time algorithms are known. Enhancing task data locality in large scale data processing systems is crucial for the job completion time. Most of the approaches to improve data locality are either greedy and ignore global optimization, or suffer from high computation complexity. This problem is addressed by proposing a heuristic task scheduling algorithm called Balance-Reduce (BAR) in [6].

Load balancing task scheduler balance the entire system load while trying to minimizing the make span of a given tasks set. Two different load balancing scheduling algorithms based on ant colony are proposed in [7] and [8]. Another ant colony based algorithm aims to minimize job completion time based on pheromone is proposed in [9]. Cloud Loading Balance algorithm [10], adds capacity to the dynamic balance mechanism for the cloud environment. The decision, which workloads to outsource to what cloud provider, should maximize the utilization of the internal infrastructure and minimize the cost of running the outsourced tasks in the cloud, while taking into account the applications' quality of service constraints. A set of heuristics, to cost-efficiently schedule deadlineconstrained computational applications, is proposed in [11]. Multi-objective meta-heuristics scheduling algorithm for multi-cloud environment is proposed in [12]. This algorithm tries to achieve application highavailability and fault-tolerance while reducing the application cost and keeping the resource load maximized. Because of the increasing large Web graph and social networks, cost-conscious large graph processing scheduling is important and a heuristic for the same is proposed in [13].

An optimized algorithm based on GA to schedule independent and divisible tasks adapting to different computation and memory requirements is proposed in [14]. Multi-agent genetic algorithm (MAGA) [15] is a hybrid algorithm of GA which solves the load balancing problem in cloud computing. COA (Course of action) planning involves resource allocation and task scheduling. A robust COA planning with varying durations based on GA is proposed in [16]. Reducing energy consumption is an increasingly important issue in cloud computing, more specifically when dealing with High Performance Computing (HPC). A multi-objective genetic algorithm (MO-GA), proposed in [17], optimizes the energy consumption, carbon dioxide emissions and the generated profit of a geographically distributed cloud computing infrastructure. Another parallel genetic algorithm based resource scheduling is proposed in [18]. Simulated annealing is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. An optimized algorithm for task scheduling based on



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

genetic simulated annealing algorithm in cloud computing is proposed in [19]. The scalability of a computing system can be mainly identified by size, geographical distribution, administrative constraints, heterogeneity, energy consumption and transparency.

A low complexity energyefficient heuristic algorithm for scheduling, proposed in [20], performs efficiently demonstrating their applicability and scalability. In batch mode, tasks are scheduled only at some predefined time. This enables batch heuristics to know about the actual execution times of a larger number of tasks. Min-min and Max-min are heuristics used for batch mode scheduling. Heuristics based improved Maxmin algorithm is proposed in [21] and the QoS Min-Min scheduling algorithm is proposed in [22]. Bag of tasks (BoT) applications are the one which execute independent parallel tasks. Heuristics proposed in [23] aims to maximize resource utilization while executing BoTs in heterogeneous sets of Cloud resources allocated for different numbers of hours. Another budget constraint scheduler proposed in [24] schedules large bags of tasks onto multiple clouds with different CPU performance and cost, minimizing completion time while respecting an upper bound for the budget to be spent. When providers cannot disclose private information such as their load and computing power, which are usually heterogeneous, the meta-scheduler needs to make blind scheduling decisions. In this case a deadline-constrained BoT application scheduling approach is proposed in [25].

Hai Zhong<sup>1</sup>, Kun Tao<sup>1</sup>, Xuejie Zhang [26] proposed an optimized scheduling algorithm to achieve the optimization or sub-optimization for cloud scheduling. In this algorithm an Improved Genetic Algorithm (IGA) is used for the automated scheduling policy. It is used to increase the utilization rate of resources and speed. Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya [27] presented a particle swarm optimization (PSO) based heuristic to schedule applications to cloud resources that takes into account both computation cost and data transmission cost. It is used for workflow application by varying its computation and communication costs. The experimental results show s that PSO can achieve cost savings and good distribution of workload onto resources. Reference [28] investigated the effectiveness of rescheduling using cloud resources to increase the reliability of job completion. Specifically, schedules are initially generated using grid resources while cloud resources are used only for rescheduling to deal with delays in job completion. A job in their study refers to a bag-of-tasks application that consists of a large number of independent tasks; this job model is common in many sciences and engineering applications. They have devised a novel rescheduling technique, called rescheduling using clouds for reliable completion and applied it to three well-known existing heuristics.

In fact, task assignment has been found to be NP-complete [29]. Since task assignment is NP-Complete problem, Genetic Algorithm (GA) has been used for task assignment [30]. But, genetic algorithm may not be the best method. Reference [31] has illustrated that the particle swarm optimization algorithm is able to get the better schedule than genetic algorithm in grid computing. Reference [32] has shown that the performance of Particle Swarm Optimization (PSO) algorithm is better than GA algorithm in distributed system. Not only is the PSO algorithm solution quality better than GA in most of the test cases, but also PSO algorithm run faster than GA. So, we use a method called Particle Swarm Optimization to optimize the task scheduling problem. In this paper, we focus on minimizing the total executing time and transferring time. Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi [33] worked on multiple workflows and multiple QoS. They implemented a strategy for multiple workflow management system with multiple Quality of Service. The access rate for scheduling is increased by using this strategy. This strategy minimizes the make span and cost of workflows. Topcuoglu et. al, [34] presented the HEFT algorithm. This algorithm finds the average execution time of each task and also the average communication time between the resources of two tasks. Then tasks in the workflow are ordered on a rank function. Then the task with higher rank value is given higher priority. In the resource selection phase tasks are scheduled in priorities and each task is assigned to the resource that complete the task at the earliest time.

Salehi, M.A. and Buyya, R. [35] proposed a market oriented hierarchical scheduling strategy which consists of both service level scheduling and tasklevel scheduling. The service level scheduling deals with the Task to Service assignment and the task level scheduling deals with the optimization of the Task to Virtual Machine assignment in local cloud data centers. Yu, J., Buyya, R. and Tham, C.K. [36] proposed a cost based workflow scheduling algorithm minimizes the execution cost while meeting the deadline for delivering results. It can also adapt to the dealys of service executions by rescheduling unexecuted tasks. Sakellariou, R., Zhao, H., Tsiakkouri, E. and Dikaiakos, M.D [37]



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

proposed a basic model for workflow applications that modelled as directed acyclic graph (DAGs) and that allow to schedule the nodes of DAG onto resources in a way that satisfies a budget constraint and is optimized for overall time.

Burke et al [38] propose a hyper-heuristic framework that implements commonly used graph colouring heuristics coupled with a random ordering heuristic. Tabu search is employed as the high-level search method for producing good sequences of the low-level heuristics. Each heuristic list produced by the tabu search algorithm is evaluated by sequentially using the individual heuristics to order the unscheduled events, and thus construct a complete timetable. This work also highlights the existence of two search spaces: the heuristic space and the problem solution space. The approach was tested on both course and exam timetabling benchmark instances with competitive results. A follow-up paper [39] compares the performance of several metaheuristics that operate on the search space of heuristics. Iterative techniques such as iterated local search and variable neighbourhood search were found to be more effective for traversing the heuristic search space. The study also implemented hybridisations of the hyper-heuristic framework with standard local search operating on the solution space, which was found to improve the performance of the overall system, making it competitive with state-of-the-art approaches on the studied benchmark instances. A further study [40] uses the notion of fitness landscapes to analyse the search space of graph colouring heuristics. These landscapes are found to have a high level of neutrality (ie, the presence of plateaus).

Furthermore, although rugged, they have the encouraging feature of a globally convex or big valley structure, which indicates that an optimal solution would not be isolated but surrounded by many local minima. Li et al [41] investigate two data mining techniques, artificial neural networks and binary logistic regression to find global patterns hidden in large data sets of heuristic sequences. With the trained classification rules, the performance of a resulting solution during the hyper-heuristic search can be predicted without the need to undertake the computationally expensive determination of the solution and calculation of the objective function. In the initial study [42], each element of the population is a variable length string, where each character represents a heuristic. The approach produced feasible examination timetables with soft constraints within the range of other search methods employed for this purpose, and outperformed previous hyper-heuristics on a number of the tested instances. The idea is to learn associations between problem states and adequate heuristics for timetabling. Specifically, the system tries to discover a set of labelled points in the space of the problem states.

Each label refers to a heuristic, and the algorithm works by repeatedly finding the nearest labelled point to the current condition and applies its label until a complete solution is built. Various different forms of problem-state description and methods of measuring the fitness were studied. The approach was able to generate fast and simple problem-solving algorithms that offer good performance over a range of exam and class timetabling problems. Sabar et al [43] utilise hierarchical hybridisations of four low-level graph colouring heuristics for producing even orderings. A combined difficulty index is calculated by considering all the orderings and events are scheduled according to this index. The approach produced competitive result on the studied benchmark instances. Numbers of authors have done work in the area of scheduling algorithms. Table 1 represents the comparative study of various scheduling algorithm, nature of scheduling algorithm, objective criteria i.e. the parameters which have been focused for optimization and the environment in which the scheduling algorithms were applied. The heuristic algorithms are priority based and mainly problem centric. The developer can use his own experience to assign priority to workflow applications and cloud resources.

## INFERENCE:

The scheduling in a cloud environment is a challenging task because of the following reasons:

1. The resource pool is central which caters to the needs of all the jobs. So it is difficult to predict which resources will be available at the time of actual execution of the jobs.
2. It is difficult to apply access control enforcement while the workflow is being executed, if the access rights of jobs change dynamically.
3. It is difficult to handle the dynamic workflow applications in which the structure of workflow graph changes with time.
4. It is difficult to reduce overhead involved while generating schedules for multiple workflows because there can be many users competing for common resources and decisions must be made in possible shortest time.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

5. It is difficult to achieve maximum possible utilization of resources while scheduling leveled workflow applications because of dependencies, and different load and resource requirements among different levels.
6. The scheduling decisions for workflow applications become complicated when made by multiple distributed schedulers in hybrid Cloud.
7. The virtual instances run on physical machines. When physical machine fails due to hardware failure or any other reason, the entire workflow application may need to be restarted. It is difficult to migrate one workflow application running on one virtual machine to another virtual machine.
8. It is difficult to achieve multi-objective criteria imposed by certain workflow applications. Different techniques produce different results in such situations and it becomes difficult to select a particular scheduling technique for a generalized class of applications.

Table.1. Comparative Study

Method	Objective criteria	Advantages	Disadvantages
Heuristic algorithm for DAG Scheduling	Makespan	It performs better than DLS	Lack of reliability
heuristic based strategy list scheduling	Makespan, Load Balance	Better optimization of makespan and load balance	Need to explore a scheduling algorithm with availability criteria
Improved Critical Path using Descendant Prediction	Makespan and Load Balance	Finds the schedule that results in makespan minimization and improve the utilization of resources.	To increase the energy efficiency, need to Schedule the workflow dynamically.
H-Green heuristic algorithm	Energy Efficient	reduce the power consumption	The computation cost is high
List Scheduling with genetic	Makespan, Economic Cost, Energy Consumption, Reliability	conserve more energy and achieve higher level of load balancing	Reschedule the unexecuted tasks is required to minimize the computation cost
Improved Cost-Based Algorithm for Task Scheduling	Cost, Performance	Improves the computation /communication ratio	Need to minimize the execution time which is significantly minimize the makespan
A Particle Swarm Optimization based Heuristic for Scheduling	Resource utilization, time	It is used for good distribution of workload onto resources	Lack of both reliability and availability criteria

### III. CONCLUSION

In this paper, we surveyed various existing workflow scheduling algorithms and tabulated them on the basis of nature of scheduling algorithm, type of algorithm, objective criteria and the environment to which the workflow scheduling algorithm was applied. From the literature reviewed, it is clear that lot of work has already been in the area of workflow scheduling but still there are many areas which require further attention. In this paper, a high-performance hyper-heuristic algorithm presented to find better scheduling solutions for cloud computing systems. The proposed algorithm uses two detection operators to automatically determine when to change the lowlevel heuristic algorithm and a



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

perturbation operator to fine tune the solutions obtained by each low-level algorithm to further improve the scheduling results in terms of makespan.

## REFERENCES

1. T. Dillon, C. Wu and E. Chang, "Cloud Computing: Issues and Challenges," Proceedings of the IEEE 24<sup>th</sup> International Conference Advanced Information Networking and Applications, Perth, 20-23 April 2010, pp.27-33.
2. F. Baroncelli, B. Martini and P. Castoldi, "Network Virtualization for Cloud Computing," Journal of Annals of Telecommunications, Vol. 65, No. 1-12, 2010, pp. 713-721.
3. T. D. Braun, H. J. Siegal, N. Beck, L. L. Boloni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, Y. Bin, D. Hensgen and R. F. Freund, "A Comparison Study of Static Mapping Heuristics for a Class of Meta-Tasks on Heterogeneous Computing Systems," Proceedings of the 8th Heterogeneous Computing Workshop, San Juan, 12 April 1999, pp. 15-29.
4. M. Maheswaran, S. Ali, H. J. Siegal, D. Hensgen and R.F. Freund, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems," Journal of Parallel and Distributed Computing: Special Issue on Software Support for Distributed Computing, Vol. 59, No. 2, 1999, pp. 107-131.
5. J. D. Ullman, "NP-Complete Scheduling Problems," Journal of Computer System Sciences, Vol. 10, No. 3, 1975, pp. 384-393.
6. Jiahui Jin, Junzhou Luo, Aibo Song, Fang Dong, Runqun Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing", IEEE, 2011
7. Xin Lu, Zilong Gu, "A load-adaptive cloud resource scheduling model based on ant colony algorithm", IEEE, 2011
8. Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization", IEEE, 2011
9. Xiangqian Song, Lin Gao, Jieping Wang, "Job scheduling based on ant colony optimization in cloud computing", IEEE, 2011
10. Zhang Bo, Gao Ji, Ai Jieqing, "Cloud Loading Balance algorithm", IEEE, 2011
11. Van den Bossche, R., Vanmechelen, K., Broeckhove, J., "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds", IEEE, 2012
12. Frincu, M.E., Craciun, C., "Multi-objective Metaheuristics for Scheduling Applications with High Availability Requirements and Cost Constraints in MultiCloud Environments", IEEE, 2012
13. Jian Li, Sen Su, Xiang Cheng, Qingjia Huang, Zhongbao Zhang, "Cost-Conscious Scheduling for Large Graph Processing in the Cloud", IEEE, 2011
14. Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing", IEEE, 200
15. Kai Zhu, Huaguang Song, Lijing Liu, Jinzhu Gao, Guojian Cheng, "Hybrid Genetic Algorithm for Cloud Computing Applications", IEEE, 2012
16. Luohao Tang, Cheng Zhu, Weiming Zhang, Zhong Liu, "Robust COA planning with varying durations", IEEE, 2011
17. Kessaci, Y., Melab, N., Talbi, E.-G., "A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures", IEEE, 2011
18. Zhongni Zheng, Rui Wang, Hai Zhong, Xuejie Zhang, "An approach for cloud resource scheduling based on Parallel Genetic Algorithm", IEEE, 2011
19. Gan Guo-ning, Huang Ting-lei, Gao Shuai, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment", IEEE, 2010
20. Diaz, C.O., Guzek, M., Pecero, J.E., Bouvry, P., Khan, S.U., "Scalable and Energy-Efficient Scheduling Techniques for Large-Scale Systems", IEEE, 2011
21. Gao Ming and Hao Li, "An Improved Algorithm Based on Max-Min for Cloud Task Scheduling", Yunnan University, China, 2011
22. Ching-Hsien Hsu, Tai-Lung Chen, "Adaptive Scheduling Based on Quality of Service in Heterogeneous Environments", IEEE, 2010
23. Gutierrez-Garcia, J.O., Kwang Mong Sim, "A Family of Heuristics for Agent-Based Cloud Bag-of-Tasks Scheduling", IEEE, 2011
24. Oprescu, A., Kielmann, T., "Bag-of-Tasks Scheduling under Budget Constraints", IEEE, 2011
25. Netto, M.A.S., Buyya, R., "Offer-based scheduling of deadline-constrained Bag-of-Tasks applications for utility computing systems", IEEE, 2009
26. Hai Zhong<sup>1</sup>, Kun Tao<sup>1</sup>, Xuejie Zhang<sup>1, 2</sup>, "An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems", in Fifth Annual China Grid Conference, 2010
27. Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar Buyya, "A Particle Swarm Optimization based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments".
28. Y. C. Lee, A. Y. Zomaya, Rescheduling for reliable job completion with the support of clouds, Future Generation Computer Systems 26 (2010) 1192-1199
29. V.M. Lo, "Task assignment in distributed systems", PhD dissertation, Dep. Comput. Sci., Univ. Illinois, Oct. 1983.
30. G. Gharooni-fard, F. Moein-darbari, H. Deldari and A. Morvaridi, Procedia Computer Science, Volume 1, Issue 1, May 2010, Pages 1445-1454, ICCS 2010.
31. L. Zhang, Y.H. Chen, R.Y. Sun, S. Jing, B. Yang, "A task scheduling algorithm based on PSO for Grid Computing", International Journal of Computational Intelligence Research. (2008), pp.37-43.
32. A. Salman, "Particle swarm optimization for task assignment Problem". Microprocessors and Microsystems, November 2002. 26(8):363-371.
33. Xu, M., Cui, L., Wang, H. and Bi, Y., "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing", IEEE 11th Int'l Symposium on Parallel and Distributed Processing with Applications, Chengdu, China, pp. 629-634 (2009)
34. Topcuoglu, H., Hariri, S., & Wu, M. Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. Parallel and Distributed Systems, IEEE Transactions on, 13(3), 260-274.
35. Salehi, M.A. and Buyya, R. "Adapting market-oriented scheduling policies for cloud computing", Proceedings of the 10th Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Busan, Korea, pp. 351-362 (2010).



# International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 11, November 2014**

36. Yu, J., Buyya, R. and Tham, C.K. "Cost-based scheduling of scientific workflow applications on utility grids", First Int'l Conference on e-Science and Grid Computing, Melbourne, Australia, pp. 140–147(2005)
37. Sakellariou, R., Zhao, H., Tsiakkouri, E. and Dikaiakos, M.D. "Scheduling workflows with budget constraints", In Integrated Research in GRID Computing, S. Gorlatch and M. Danelutto, Eds Springer- Verlag., pp. 189–202, (2007).
38. Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1), 177-192
39. Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of scheduling*, 12(1), 55-89.
40. Ochoa G, Qu R and Burke EK (2009). Analyzing the landscape of a graph based hyperheuristic for timetabling problems. In: Genetic and Evolutionary Computation Conference (GECCO 2009), ACM: New York, pp 341–348.
41. Li J, Burke EK and Qu R (2011). Integrating neural network and logistic regression to underpin hyper-heuristic search. *Knowledge-Based Systems* 24(2): 322–330.
42. Pillay N and Banzhaf W (2007). A genetic programming approach to the generation of hyper-heuristics for the uncapacitated examination timetabling problem. In: 13th Portuguese Conference on Artificial Intelligence (EPIA 2007), Lecture Notes in Computer Science, Vol. 4874, Springer: Berlin, pp 223–234.
43. Sabar NR, Ayob M, Qu R and Kendall G (2011). A graph coloring constructive hyperheuristic for examination timetabling problems. *Applied Intelligence* 37(1): 1–11.