

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235738379>

# Scheduling using improved genetic algorithm in cloud computing for independent tasks

Conference Paper · August 2012

DOI: 10.1145/2345396.2345420

CITATIONS

35

READS

2,923

2 authors:



[Amandeep Verma](#)

Panjab University

30 PUBLICATIONS 444 CITATIONS

[SEE PROFILE](#)



[Pardeep Kumar](#)

Panjab University

2 PUBLICATIONS 79 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Cloud Computing [View project](#)

# Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks

Pardeep Kumar

M.E. Student

UIET, Panjab University

Chandigarh, India

pardeep\_prof@yahoo.com

Amandeep Verma

Assistant Professor

UIET, Panjab University

Chandigarh, India

amandeepverma@pu.ac.in

## ABSTRACT

Cloud computing is a new technology and it is becoming popular day by day because of its great features. In this technology almost everything like hardware, software and platform are provided as a service. These services are charged from users on the pay-per-use bases. A cloud provider in cloud computing provides services on the basis of clients' requests. An important issue in cloud computing is the scheduling of users' requests means how to allocate resources to these requests, so that the requested tasks can be completed in a minimum time according to the user defined time. A good scheduling technique also helps in efficient utilization of the resources. Many scheduling algorithms have been researched like Min-Min, Max-Min, X-Sufferage, Genetic Algorithm, Particle Swarm Optimization etc. In this paper the three scheduling techniques Min-Min, Max-Min and Genetic Algorithm have been discussed and performance metrics of Min-Min and Max-Min have been shown. The performance of the standard Genetic Algorithm and the proposed Improved Genetic Algorithm have been checked against the sample data. A new scheduling idea is also proposed in which Min-Min and Max-Min can be combined in Genetic Algorithm.

## Keywords

Cloud Computing; Min-Min; Max-Min; Genetic Algorithm; Improved Genetic Algorithm

## 1. INTRODUCTION

As the IT technologies are growing day by day, the need of computing and storage are rapidly increasing. To invest more and more equipments is not an economic method for an organization to satisfy the even growing computational and storage need. So Cloud Computing has become a widely accepted paradigm for high performance computing, because in Cloud Computing all type of IT facilities are provided to the users as a service. In Cloud Computing the term Cloud is used for the service provider, which holds all types of resources for storage, computing etc. Mainly three types of services are provided by the cloud. First is Infrastructure as a Service (IaaS), which provides cloud users the infrastructure for various purposes like the storage system and

computation resources. Second is Platform as a Service (PaaS), which provides the platform to the clients so that they can make their applications on this platform. Third is Software as a Service (SaaS), which provides the software to the users and hence the users don't need to install the software on their own machines and they can use the software directly from the cloud. Due to the wide range of facilities provided by the cloud computing, the Cloud Computing is becoming the need of the IT industries. The services of the Cloud are provided through the Internet. The devices that want to access the services of the Cloud should have the Internet accessing capability. Devices need to have very less memory, a very light operating system and browser. Cloud Computing provides many benefits: it results in cost savings because there is no need of initial installation of much resource; it provides scalability and flexibility, the users can increase or decrease the number of services as per requirement; maintenance cost is very less because all the resources are managed by the Cloud providers.

The rest of this is organized as follows: section 2 tells about what related work has been done previously in the related field. Section 3 discuss about scheduling techniques: Min-Min and Max-Min. Section 4 is about the standard Genetic Algorithm. Section 5 discusses about the proposed Improved Genetic Algorithm. Section 6 shows the simulations and results of the new proposed algorithm and the performance comparison of the improved genetic algorithm and the standard genetic algorithm. Section 7 is about the conclusion and the future scope of this paper. Section 8 shows the referenced papers.

## 2. RELATED WORK

All Scheduling of tasks is a critical issue in Cloud Computing, so a lot of researches have been done in this area. The basic idea about scheduling in Cloud Computing is discussed in [1] [2]. Genetic algorithm and some of its modified versions are discussed in papers from [5] to [14]. Some of the scheduling techniques: Min-Min Algorithm, Max-Min Algorithm, Minimum Completion Time and Minimum Execution Time are discussed and compared in [15].

## 3. SCHEDULING TECHNIQUES

In this paper we assumed a set of tasks and a set of resources. Based on these tasks and resources, we have evaluated the performance of Min-Min and Max-Min algorithms. Performance is checked against the sample data of Table 1, in which we have considered 6 tasks and 4 machines and the execution times (in seconds) of all the tasks on all machines are shown.

(c) 2012 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the national government of India. As such, the government of India retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

ICACCI'12, August 03-05, 2012, CHENNAI, India

Copyright 2012 ACM 978-1-4503-1196-0/12/08...\$10.00

**Table 1. Execution Times**

	M0	M1	M2	M3
<b>T0</b>	160	400	80	200
<b>T1</b>	40	100	20	50
<b>T2</b>	100	250	50	125
<b>T3</b>	20	50	10	25
<b>T4</b>	140	350	70	175
<b>T5</b>	80	200	40	100

### 3.1 Min-Min Algorithm

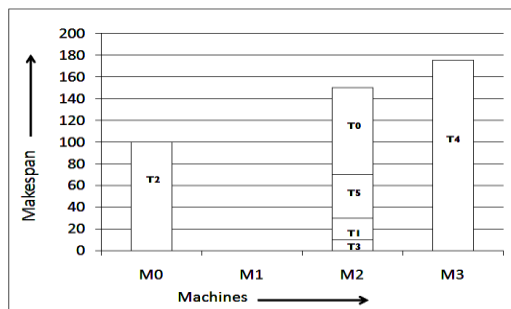
It begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources.

### 3.2 Max-Min Algorithm

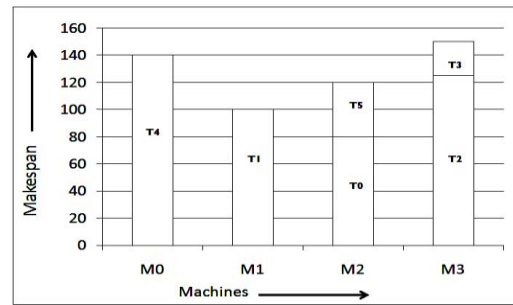
Max-Min is almost same as the min-min algorithm except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. Then according to that maximum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources.

We have assumed four machines and six tasks and the execution times of all the tasks on all machines as shown in Table 1. We have compared the performance of Min-Min and Max-Min and the assignment of tasks to the machines are shown in Figure 1 and Figure 2.

By using different scheduling techniques, the tasks are assigned in a different sequence to different machines for execution. When we apply the Min-Min and Max-Min, then the tasks are assigned to the machines as shown in the charts of Figure 1 and Figure 2.



**Figure 1. Task Assignment by Min-Min Algorithm**



**Figure 2. Task Assignment by Max-Min Algorithm**

The highest bar shows the makespan. We have got the different values of makespans by two techniques and these are:

Method Used	Makespan
Min-Min	175 seconds
Max-Min	150 seconds

Based on the different execution times of tasks on resources, one technique can outperform the other and the assignment of resources to the tasks can change i.e. if any task is assigned to a machine if we use one technique; the same task can be assigned to another machine if we use other technique.

## 4. GENETIC ALGORITHM

Genetic algorithm is the way of scheduling that is based on the biological concept of population generation. The main terms used in genetic algorithms are initial population, fitness function, selection, crossover and mutation.

### 4.1 Initial Population

Initial population is the set of all the individuals that are used in the genetic algorithm to find out the optimal solution. Every solution in the population is called as an individual. And every individual is represented as a chromosome for making it suitable for the genetic operations. From the initial population the individuals are selected and some operations are applied on those to form the next generation. The mating chromosomes are selected based on some specific criteria.

### 4.2 Fitness Function

A fitness function is used to measure the quality of the individuals in the population according to the given optimization objective. The fitness function can be different for different cases. In some cases the fitness function can be based on makespan while in cases it can be based on budget constraints.

### 4.3 Selection

We use the proportion selection operator to determine the probability of various individuals genetic to the next generation in population. The proportional selection operator means the probability which is selected and genetic to next generation groups is proportional to the size of the individual's fitness.

### 4.4 Crossover

We use single-point crossover operator. Single-point crossover means only one intersection was set up in the individual code, at that point part of the pair of individual chromosomes is exchanged.

## 4.5 Mutation

Mutation means that the values of some gene locus in the chromosome coding series were replaced by the other gene values in order to generate a new individual. Mutation is that negates the value at the mutate points with regard to binary coded individuals.

Genetic Algorithm works in the following manner:

1. Begin
2. Initialize population by random solutions
3. Evaluate each candidate
4. Repeat until (termination condition occur)
5. Do
  - a. Select parents
  - b. Recombine pairs of parents
  - c. Mutate the resulting offsprings
  - d. Evaluate new candidate
  - e. Select individuals for next generation
6. End

As per the condition shown in the above procedure, the evaluation of the candidates is repeated until we do not find the optimal/best solution.

If we apply this genetic algorithm on the sample data of Table 1, then the tasks are assigned as shown in Figure 3 and the makespan that we got using this technique is 260 seconds. This can be different if run this technique again on the same data because machines are assigned to tasks randomly for initial population generation and schedules and tasks are chosen randomly each and every time for the crossover and mutation operations respectively.

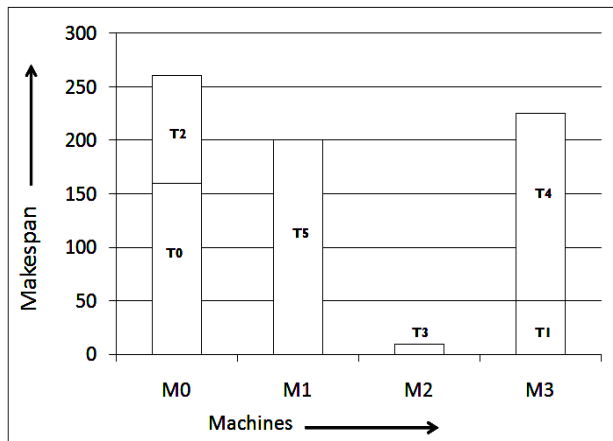


Figure 3. Task assignment by Standard Genetic Algorithm

## 5. IMPROVED GENETIC ALGORITHM

As we can conclude from Genetic Algorithm that the solutions that are fit; give better generations further when we apply genetic operators on them. So, an idea is proposed for generating initial population by using Min-Min and Max-Min, which can provide better initial population than if we choose the initial population randomly. So the proposed idea can give an Improved Genetic Algorithm.

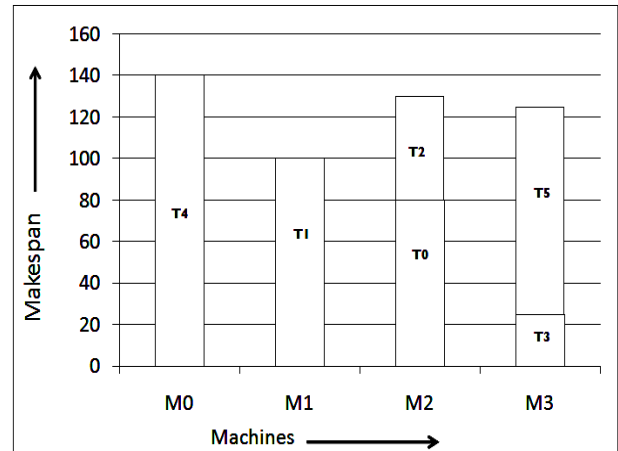


Figure 4. Task assignment by Improved Genetic Algorithm

The new Improved Genetic Algorithm will be like:

1. Begin
2. Find out the solution by Min-Min and Max-Min
3. Initialize population by result of Step 2
4. Evaluate each candidate
5. Repeat until (termination condition occur)
6. Do
  - a. Select parents
  - b. Recombine pairs of parents
  - c. Mutate the resulting offsprings
  - d. Evaluate new candidate
  - e. Select individuals for next generation
7. End

If we apply this improved genetic algorithm on the sample data of Table 1, then the tasks are assigned as shown in Figure 4. This technique results in a makespan of 140 seconds.

## 6. Simulations and Results

We have used CloudSim as a simulator for checking the performance of our improved algorithm and the standard Genetic Algorithm. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited efforts.

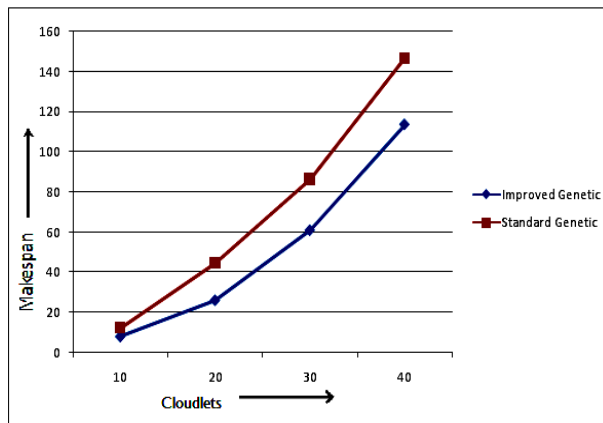
We have considered Virtual Machines as resource and Cloudlets as tasks/jobs. We have checked the performance of the algorithms in two cases: in first case, we have fixed the number of virtual machines and varied the number of cloudlets; in second case we case we have fixed the number of cloudlets and varied the number of virtual machines. The makespans that the algorithms produce are shown in tables and the graphs corresponding to these tables have been shown.

In first case, we have fixed the number of virtual machines as 10 and we are varying the number of cloudlets from 10 to 40 with a difference of 10. We have run each algorithm 10 times and the average of these 10 runs is noted down in Table 2, which is shown below:

**Table 2. Makespans for Fixed VMs and Varying Cloudlets**

VMs fixed : 10		Cloudlets varying			
		10	20	30	40
Method Used	Improved Genetic	8	26.1	60.9	113.5
	Standard Genetic	12.4	44.7	86.5	146.8

The performance of the first case according to the values of Table 2 is shown in graph of Figure 5. In this figure it can be seen that the improved genetic algorithm outperforms the standard genetic algorithm.



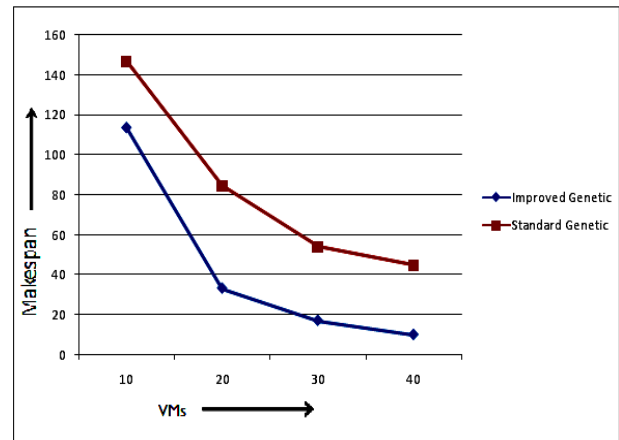
**Figure 5. Makespans for Fixed VMs and Varying Cloudlets**

In second case, cloudlets are fixed to 40 and the algorithm is run with a varying number of VMs starting from 10 to 40 with a difference of 10; means the first run is done with 10 VMs, second with 20 VMs and so on up to 40. The makespan can vary a little for each run, so each algorithm is run 10 times and average of these 10 runs is noted down in the Table 3.

**Table 3. Makespans for Fixed Cloudlets and Varying VMs**

Cloudlets fixed : 40		VMs varying			
		10	20	30	40
Method Used	Improved Genetic	113.5	33.2	17.1	10
	Standard Genetic	146.8	84.7	54	44.8

The performance of the second case according to the values of Table 3 is shown in graph of Figure 6.



**Figure 6. Makespans for Fixed Cloudlets and Varying VMs**

From both the graphs, it can be observed that the makespan of the Improved Genetic Algorithm is less than that of Standard Genetic Algorithm. So the new improved Genetic Algorithm can help in reducing overall execution time of the tasks and in proper utilization of resources.

## 6.1 Resource Utilization

We have calculated the resource utilization for both of the cases when we apply standard genetic algorithm and improved genetic algorithm. We have considered five problems set with fixed number of VMs and varying number of Cloudlets as:

P1: 4 VM, 5 Cloudlets

P2: 4 VM, 10 Cloudlets

P3: 4 VM, 12 Cloudlets

P4: 4 VM, 15 Cloudlets

P5: 4 VM, 20 Cloudlets

The MIPS of the VMs/Resources and the MI of Cloudlets are taken in the following manner: MIPS of the first VM is set to 250 and every subsequent VM is having MIPS with an increase of 50; MI of the first Cloudlet/Task is set to 1000 and every subsequent Cloudlet is having 500 more MI than the previous one.

Table 4 shows the resource utilization of two scheduling algorithms: standard genetic algorithm and improved genetic algorithm. The table shows that the improved genetic algorithm uses the resource effectively. So we can conclude from this table that improved scheduling algorithm uses the maximum amount of resources and reduces the makespan.

**Table 4. Resource Utilization (Makespan in Seconds)**

Problem Sets	Improved Genetic				Standard Genetic			
	R1	R2	R3	R4	R1	R2	R3	R4
P1	6	6	8	8	0	8	8	10
P2	24	24	22	24	22	18	29	26
P3	36	29	35	35	24	34	29	35
P4	54	52	50	50	38	61	66	38
P5	94	92	83	82	84	71	96	92

In Table 4, the resource utilization shows the makespan of resources is shown in seconds. Resource utilization (in percentage) for a particular problem is calculated using the formula:

$$RU_i = MS * 100 / TRU$$

**RU** = Usage of Resource in percentage

**MS** = Makespan of Meta task

**TRU** = Total Resource Usage – it is the total time for which the resource  $r_i$  is used

The experimental results achieved by using this formula for improved genetic algorithm and standard genetic algorithm are shown in Table 5, which shows the resource utilization in percentage. From Table 5, it can be observed that in case of improved genetic algorithm the utilization of resources is better than the utilization of resources in case of the standard genetic algorithm. Almost all the resources in case of improved genetic algorithm are utilized to a good extent, but in case of standard genetic algorithm there is a large gap in the percentage usage of resources. This means that in case of standard genetic algorithm some of the resources are used fully, but some resources are not used efficiently.

**Table 5. Resource Utilization in Percentage**

Problem Sets	Algorithm used	R1	R2	R3	R4
<b>P1</b>	<b>Improved Genetic</b>	75	75	100	100
	<b>Standard Genetic</b>	0	80	80	100
<b>P2</b>	<b>Improved Genetic</b>	100	100	91	100
	<b>Standard Genetic</b>	75	62	100	89
<b>P3</b>	<b>Improved Genetic</b>	100	80	97	97
	<b>Standard Genetic</b>	91	100	78	94
<b>P4</b>	<b>Improved Genetic</b>	100	96	92	92
	<b>Standard Genetic</b>	57	92	100	57
<b>P5</b>	<b>Improved Genetic</b>	100	97	88	87
	<b>Standard Genetic</b>	87	73	100	95

The percentage utilization of resources in Table 5 shows that the improved genetic algorithm results in better resource utilization and produces better results. So it can be concluded that the improved genetic algorithm outperforms the standard genetic algorithm.

## 7. CONCLUSION AND FUTURE SCOPE

Task scheduling is main problem in cloud computing scenario. Efficient task scheduling is essential for better utilization of resources. In this paper three scheduling algorithms are discussed such as Min-Min, Max-Min and Genetic Algorithm and a new scheduling algorithm that uses the three discussed scheduling algorithms; is proposed and devised. The experimental results show that the improved genetic algorithm minimizes the

makespan and utilizes the resources effectively than the standard genetic algorithm. In this paper we have used the makespan as fitness criteria for checking the fitness of the scheduling results. This idea can be further extended in which we can use execution cost of the resource as fitness criteria. This method can be adapted in existing cloud computing systems for decreasing makespan and better resource utilization.

## 8. REFERENCES

- [1] Huang Q.Y., Huang T.L. "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 673-675, 2010
- [2] Hsu C.H. and Chen T.L., "Adaptive Scheduling based on Quality of Service in Heterogeneous Environments", IEEE 4th International Conference on Multimedia and Ubiquitous Engineering (MUE), 2010, Cebu, pp. 1-6, 2010
- [3] Zhang Jinquan, Lina N. and Jiang Changjun, "A Heuristic Scheduling Strategy for Independent Tasks on Grid", IEEE Eighth International Conference on High-Performance Computing in Asia-Pacific Region, 2005. Proceedings, Beijing, 6 pp. 593, 2005
- [4] Xu Wang; Beizhan Wang; Jing Huang, "Cloud computing and its key techniques", IEEE International conference on Computer Science and Automation Engineering (CSAE), 2011, Shanghai, pp. 404-410, 2011
- [5] Yin H., Wu H., Zhou J., "An Improved Genetic Algorithm with Limited Iteration for Grid Scheduling", IEEE Sixth International Conference on Grid and Cooperative Computing, 2007. GCC 2007, Los Alamitos, CA, pp. 221-227, 2007
- [6] Guo G., Ting-Iei H., Shuai G., "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60-63, 2010
- [7] Zhao C., Zhang S., Liu Q., Xie J., Hu J., "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing", IEEE 5th International Conference on Wireless Communications, Networking and Mobile Computing WiCom '09, Beijing, pp.1-4, 2009
- [8] Loukopoulos T., Lampsas P., Sigalas P., "Improved Genetic Algorithms and List Scheduling Techniques for Independent Task Scheduling in Distributed Systems", IEEE, Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, Adelaide, SA, pp. 67-74, 2007
- [9] Abdulal W., Jadaan O.A., Jabas A., Ramachandram S., "Genetic Algorithm for Grid Scheduling using Best Rank Power", IEEE World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, Coimbatore, pp. 181-186, 2009
- [10] Wang J., Duan Q., Jiang Y., Zhu X., "A New Algorithm for Grid Independent Task Schedule :Genetic Simulated Annealing", IEEE World Automation Congress (WAC), 2010,Kobe, pp. 165-171, 2010
- [11] Verma R., Dhingra S., "Genetic Algorithm for Multiprocessor Task Scheduling", IJCSMS International



Journal of Computer Science and Management Studies,  
Vol.1, Issue 02, pp. 181-185, 2011

- [12] Jia Yu and Buyya R., “A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms”, IEEE Workshop on Workflows in Support of Large-Scale Science, 2006. WORKS '06, Paris, pp. 1-10, 2006
- [13] Mocanu, Maria E., Mihai F., Ionut A. M. and Nicolae T., “Cloud Computing – Task Scheduling based on Genetic Algorithms”, 2012 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, pp. 1-6, 2012
- [14] Xue Zhang; Wenhua Zeng, “Grid Workflow Scheduling Based on Improved Genetic Algorithm”, IEEE International Conference on Computer Design and Applications (ICCD), 2010, Qinhuangdao, pp. V5-270 – V5 273, 2010
- [15] M. Hememalini, “Review on Grid Task Scheduling in Distributed Heterogeneous Environment”, International Journal of Computer Applications 40(2), pp. 24-30, 2012