

# Performance Comparison of Three Batch-Mode Scheduling Heuristics

Vishu Narula  
BFCET  
Bathinda

Jyoti Bansal  
Research Scholar,  
PTU Kapurthala

Shaveta Garg, PhD  
GZS, PTU Campus  
Bathinda

Paramjit Singh, PhD  
GZS, PTU Campus  
Bathinda

## ABSTRACT

Grid scheduling issue has been an exploration hotspot lately. Some custom heuristics have been utilized to upgrade it and have got some great results. In any case, selecting the best heuristic to use in a given domain remains a troublesome issue. So to beat this, a few examinations have been made in this paper which will give the data that which heuristic will give better Makespan, Flowtime and Average completion time value. So in this paper, three heuristics i.e., Min-Min, Max-Min & LJFR-SJFR are chosen, compared, analyzed and executed by using Gridsim 5.2.

## General Terms

Scheduling Heuristics.

## Keywords

Grid and Desktop Grid computing, Min-Min, Max-Min & LJFR-SJFR

## 1. INTRODUCTION

The recognition of the Internet and the accessibility of capable PCs and rapid systems as ease merchandise segments are changing the way we utilize PCs today. These specialized open doors have prompted the likelihood of utilizing graphically distributed and multi-proprietor resources to settle huge scale issues in science, designing, and trade. This new approach is known as Grid Computing [1, 2]. Conversely, Desktop grids are utilized as a part of a functional processing idle model that can transform huge computational activities in different application field, utilizing the idle cycles of heterogeneous resources (for the most part desktop PCs) joined over the Internet. The reputation of the Internet has made another much expansive scale open door for Grid computing. Points of fact, many desktop PCs, whose idle cycles can be changed to run Grid applications, are joined with wide-zone systems both in the business enterprises and in the home. These new stages for high throughput applications are called Desktop Grids [3] & [4].

The administration of resources and planning of utilizations in such huge scale conveyed frameworks is an impenetrable undertaking. To demonstrate the adequacy of individual resource and related scheduling computations, their execution needs to be assessed under diverse situations, for example, fluctuating number of resources and clients with distinctive fundamentals. So for this, we utilize some scheduling heuristics for scheduling the tasks and resources in this grid environment. These scheduling heuristics are divided into two sections i.e., knowledge-based and knowledge-free scheduling. Further these scheduling heuristics are subdivided into online and batch-mode scheduling. In online scheduling, tasks are allocated to resources when it arrives at the scheduler and this allotment is not changed once it is allotted. Another side in batch mode, tasks are not allotted onto the

resources as they arrive; rather they are gathered into a queue and it is inspected for mapping at prescheduled times which is known as mapping events.

The primary objective of our paper is to compare the performance of three traditional batch-mode heuristics i.e., min-min, max-min and LJFR-SJFR with respect to three parameters i.e., Makespan, Flow time and Average completion time value. The most popular optimization criterion is minimization of Makespan i.e. the finishing time of the latest task. Makespan measures the throughput of grid system. It can be defined as:  $Makespan = \max \{C_{uv}, v=1, \dots, n\}$  where  $C_{uv}$  is completion time of task( $T_u$ ). Conversely Flowtime is the sum of the finishing times of tasks. Flowtime measures the Quality of Service of the grid system. It can be defined as:  $Flowtime = \sum \{C_{uv}, v=1, \dots, n\}$ . The Average completion time will return the average period of time taken by tasks to complete its execution.

The rest of this paper is organized as: Section 2 will characterize related work. Section 3 will provide the heuristics description. Section 4 will analyze the test problem. Section 5 will define the comparison and experimental results. At the end section 6 will conclude the work and provides the future work.

## 2. RELATED WORK

Min-min, Max-min and LJFR-SJFR had been utilized as a part of the task scheduling in computational grid. The investigation on them has received good results and their effectiveness had been demonstrated. There is additionally numerous related enhanced works that are done under this study.

In [5] & [6], on-line load adjusting calculation for desktop networks are depicted and it is analyzed that desktop Grid scheduling can be performed in a unified manner or in a completely dispersed manner and they needn't bother with a worldwide scheduler. In [7] & [8], Schedulers in this class accept the information of the execution time of individual tasks, and exploit different kind of static or dynamic resource data to perform resource determination. In [9] & [10], the two batch-mode heuristics i.e., Min-min and max-min has been discussed. The authors' have discussed its disadvantages and have tried to overcome it. In [11], eleven online and batch mode heuristics have been examined and authors' will analyze all the heuristics with the assistance of parameter and pronounce that GA will reliably give the best results. In [12] & [13], the exploratory results will pronounce that min-min heuristic will give best result for minimizing Flowtime and the proposed heuristic (Min-Max) will announce the best results for minimizing Makespan. In [14], a trial assessment of the heuristics is performed in three sections. In the first part, the on-line mode heuristics are looked at utilizing different measurements. The second part includes a

correlation of the batch mode heuristics and third part portrays that batch mode heuristics performs better over the on-line mode heuristics. In [15], different sorts of online and batch mode heuristics have been analyzed and they have discussed that there is a necessity of adjusting the load in the middle of the resources.

The above mentioned traditional heuristics are popular, effective and are used in many studies. So far, a lot of works have been done to investigate which heuristic provides minimum Makespan or minimum Flowtime or which heuristic provides both minimum Makespan and fowtime value. So to clear this we will perform a performance comparison of these heuristics. To facilitate these comparisons, some simplifying assumptions were made. For these studies, let a meta-task be defined as a collection of independent tasks with no inter task data dependencies. It is also assumed that the length of each task and the information of all available resources are known beforehand.

### 3. HEURISTICS DESCRIPTION

There are a few heuristics for grid computing. This segment will depict the batch mode heuristics.

#### 3.1 Min-min

Min-Min heuristic starts by gathering the collection of every unassigned task. Fundamentally, in this scheduling is done in two stages. In the first stage, it chooses all the tasks those who have least expected completion time i.e., earliest expected finishing time on the comparing resource. In second stage, the task with the general least expected completion time is selected and will get appointed to the comparing resources. After the allotment of tasks, allotted tasks will get excluded from the set and the procedure is repeated until all tasks in the set are mapped [16].

#### 3.2 Max-Min

Max-Min is all that much like Min-Min with the exception of in stage 2. Like Min-Min it also works in two stages and same as Min-Min it will likewise begins by selecting all the tasks the tasks that have least expected completion time. In second stage, here Max-Min will relegate the assignment of the tasks that have greatest expected finishing time relating to every resource. After the tasks that have been allotted will get excluded from the set and the procedure is rehashed until all tasks in the set got mapped [16].

#### 3.3 LJFR-SJFR

Longest Job to Fastest Resource-Shortest Job to Fastest Resource (LJFR-SJFR) heuristic starts by selecting every single unmapped task. Here, LJFR-SJFR heuristic will also works in two stages. In first stage, it will first allot the biggest task to the resources by utilizing Max-min heuristic i.e., it will first choose those tasks that have greatest expected completion time. In second stage, remaining tasks will get allocated by utilizing min-min and max-min heuristic alternatively i.e. smallest tasks on quickest resource took after by biggest task on speediest resource [16].

### 4. TEST PROBLEMS

For reasonable comparison of distinctive scheduling heuristic, this segment will outline the illustrations of above characterized heuristics. Assume that m resources  $R_v(v=1... m)$  need to process n tasks  $T_u(u=1... n)$ . Presently in like manner to the comparing heuristics table1, table2, table 3 will characterize the expected execution time of every assignment.

**Table 1: Min-Min heuristic**

Tasks (size)	R1(400)	R2(500)	R3(600)
T1(3000)	7.5	6	5
T2(4000)	10	8	6.66
T3(7000)	17.5	14	11.66
T4(9000)	22.5	18	15
T5(12000)	30	24	20
T6(14000)	35	28	23.33
T7(15000)	37.5	30	25
T8(20000)	50	40	33.33
T9(21000)	52.5	45	35
T10(22000)	55	44	36.66

**Table 2: Max-Min Heuristic**

Tasks (size)	R1(400)	R2(500)	R3(600)
T10(22000)	55	44	36.66
T9(21000)	52.5	42	35
T8(20000)	50	40	33.33
T7(15000)	37.5	30	25
T6(14000)	35.5	28	23.33
T5(12000)	30	24	20
T4(9000)	22.5	18	15
T3(7000)	17.5	14	11.66
T2(4000)	10	8	6.66
T1(3000)	7.5	6	5

**Table 3: LJFR-SJFR Heuristic**

Tasks (size)	R1(400)	R2(500)	R3(600)
T10(20000)	55	44	36.66
T9(17000)	52.5	42	35
T8(15000)	50	40	33.33
T1(2000)	7.5	6	5
T7(12000)	37.5	30	25
T2(3000)	10	8	6.66
T6(11000)	35.5	28	23.33
T3(4000)	17.5	14	11.66

T5(9000)	30	24	20
T4(8000)	22.5	18	15

By experimentally performing the corresponding succession of steps relating to every heuristic by using GridSim 5.2 [17], table 4, table 5 & table 6 will mirror the outcomes and table 7 will delineate the completion time ( $CT_{uv}$ ) of tasks  $T_u$  on resources  $R_v$  individually.

**Table 4: Result of Min-Min**

Resources	Speed	Task Assigned
R1	400	T4,T7
R2	500	T2,T5,T8
R3	600	T1,T3,T6,T9,T10

**Table 5: Result of Max-Min**

Resources	Speed	Task Assigned
R1	400	T8,T5,T1
R2	500	T9,T6,T3
R3	600	T10,T7,T4,T2

**Table 6: Result of LJFR-SJFR**

Resources	Speed	Task Assigned
R1	400	T8,T3,T5,T4
R2	500	T9,T2,T6
R3	600	T1,T7,T10

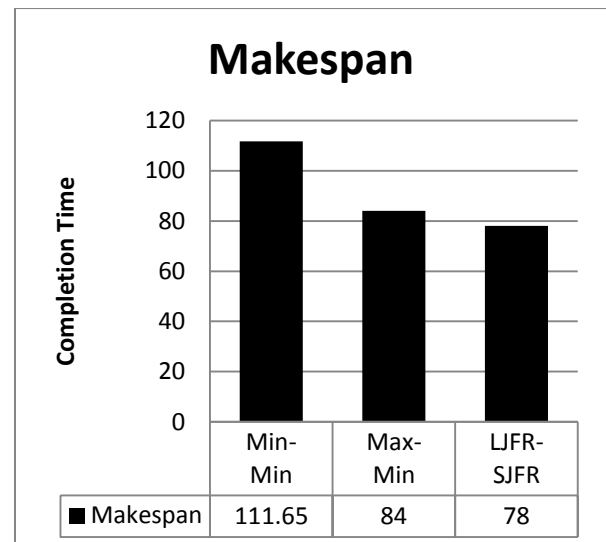
**Table7: Completion time ( $CT_{uv}$ ) of each heuristic**

Tasks	$CT_{uv}$ of Min-Min	$CT_{uv}$ of Max-Min	$CT_{uv}$ of LJFR-SJFR
T1	5	36.66	41.66
T2	8	42	50
T3	16.66	50	22.5
T4	22.5	61.66	75
T5	32	70	52.5
T6	39.99	80	78
T7	60	76.66	66.66
T8	72	84	50
T9	74.99	60	42
T10	111.65	66	36.66

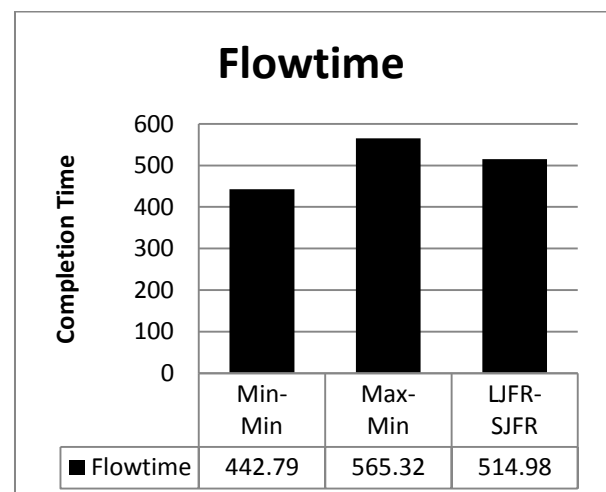
## 5. COMPARISON AND EXPERIMENTAL RESULTS

By using GridSim 5.2 [17], we have compared the performance of the above mentioned heuristics for

minimizing Makespan, Flowtime and Average Completion Time. Following examples explains the concepts which are based on 10 tasks and 3 resources. The obtained Makespan, Flowtime and Average completion time using mentioned heuristics are compared in figure 1, 2 and 3 respectively. In these figures, the first column indicates the instance name, and the second, third, fourth column depicts the Makespan, Flowtime and Average completion time of Min-Min, Max-Min & LJFR-SJFR heuristics i.e., '111.66, 87.5, 90', '442.83, 671.83, 609.166', & '44.28, 67.183, 60' respectively. As it is evident from the figures, Min-Min heuristic can minimize Flowtime and Average completion time value better than others. Conversely Max-Min provides the better Makespan value than the others. And LJFR-SJFR can minimize the Flowtime and Average completion time value better than Max-Min. Separated this we have investigated that: In Min-Min, the schedule won't stays optimal when the smaller number of tasks gets more than larger one & in Max-Min, the schedule won't stay optimal when the quantity of larger tasks gets more than smaller one. Now all these circumstances will leads to load imbalance on resources. So there is need to have some load balancing heuristic for dealing with this issue.



**Figure 1: Comparison results between heuristics on Makespan**



**Figure 2: Comparison results between heuristics on Flowtime**

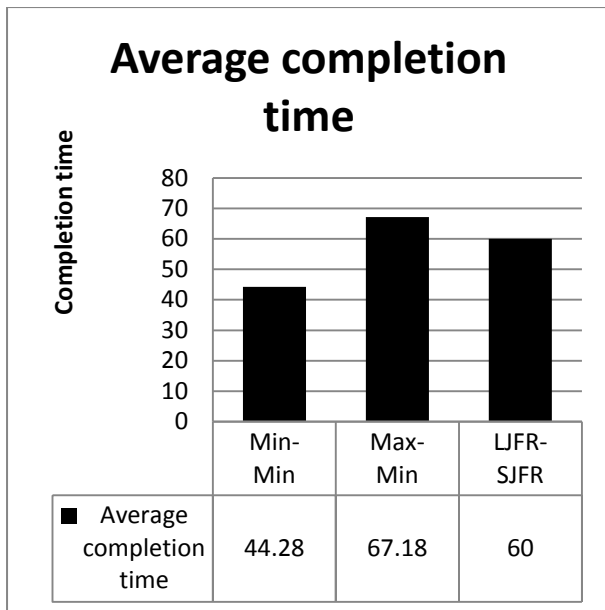


Figure 3: Comparison results between heuristics on Average Completion Time

## 6. CONCLUSION AND FUTURE WORK

In this paper we have compared three heuristics for scheduling in grid environment. The objective of our paper is to give performance comparison of three batch-mode scheduling heuristics. The examination is made to give the information that which heuristic gives better Makespan, Flowtime and Average completion time value. The exploratory results will demonstrate that Min-Min heuristic can get the best results for minimizing Flowtime and Average completion time value. Conversely Max-Min heuristic can get the best results for minimizing Makespan and the LJFR-SJFR heuristic can acquire the better results for minimizing Flowtime and Average completion time value. In the course of this study, we have broken down that there are shots of having load imbalanced among the allotment of tasks to the resources. So for future work, we have chosen to remove this load imbalance by proposing some new heuristics.

## 7. REFERENCES

[1] Mark Baker<sup>1</sup>, Rajkumar Buyya, et al. 2002, Grids and Grid technologies for wide-area distributed computing, *Software: Practice and Experience*, 32(15), pp. 1437-1466.

[2] Patricio Domingues, Artur Andrzejak, et al. 2006, Scheduling for fast turnaround time on institutional desktop grid, *CoreGRID TechRep*, 0027.

[3] SungJin Choi, HongSoo Kim, et al. 2007, Characterizing and classifying desktop grid, In null, pp. 743-748, IEEE.

[4] Issam Al-Azzoni, Douglas G. Down, et al. 2010, Dynamic scheduling for heterogeneous desktop grids, *Journal of Parallel and Distributed Computing*, 70(12), pp.1231-1240, ELESEVIER.

[5] Fatos Xhafa. 2007, Immediate mode scheduling of

independent', jobs in computational grids, In *Advanced Information Networking and Applications, AINA'07. 21st International Conference*, pp. 970-977, IEEE.

[6] K.Hemant K. Reddy, Manas Ranjan Patra, et al. 2012, An adaptive scheduling mechanism for computational desktop grid using gridgain, *Procedia Technology*, 4, pp.573-578, ELESEVIER.

[7] Francine Berman, Richard Wolski, et al. 2003, Adaptive computing on the grid using AppLeS, *Parallel and Distributed Systems, Transactions on*, 14(4), pp.369-382, IEEE.

[8] Derrick Kondo, Andrew A. Chien, et al. 2004, Resource management for rapid application turnaround on enterprise desktop grids, In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, pp. 17, IEEE.

[9] HE XiaoShan, SUN XianHe, et al. 2003, QoS guided min-min heuristic for grid task scheduling, *Journal of Computer Science and Technology*, 18(4), pp.442-451.

[10] Hesam Izakian, Ajith Abraham, et al. 2009, Performance comparison of six efficient pure heuristics for scheduling meta-tasks on heterogeneous distributed environments, *Neural Network World*, 19(6), pp.695-710.

[11] Braun, Siegel, et al. 2001, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems, *Journal of Parallel and Distributed computing*, 61(6), pp. 810-837.

[12] Hesam Izakian, Ajith Abraham, et al, 2009, Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments, In *Computational Sciences and Optimization, CSO 2009, International Joint Conference*, pp. 8-12, IEEE.

[13] Naglaa M. Reda, A. Tawfik, et al. 2014, Sort-Mid tasks scheduling algorithm in grid computing, *Journal of Advanced Research*, ELESEVIER.

[14] Muthucumar Maheswaran, Shoukat Ali, et all. 1999, Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems , In *Heterogeneous Computing Workshop, HCW 99 Proceedings Eighth*, pp.30-44, IEEE.

[15] Lee, Y. H., Leu, S., & Chang, R. S, 2011, Improving job scheduling algorithms in a grid environment , *Future generation computer systems*, 278, pp.991-998, ELESEVIER.

[16] Chaturvedi, A. K., & Sahu, R, 2011, New heuristic for scheduling of independent tasks in computational grid , *International Journal of Grid and Distributed Computing*, 43, pp.25-36.

[17] Buyya, Rajkumar, and Manzur Murshed. 2002, Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing *Concurrency and computation: practice and experience* 14, pp. 1175-1220.