International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Grid Computing Model to Solve Job Shop Scheduling and Flow Shop Scheduling by Fuzzy C-Mean Algorithm

Ajendra Kumar, Preet Pal Singh, Dipa Sharma, Pawan Joshi

Abstract*: This paper presents two computing model in grid environment to utilize the waiting time of a job on particular machines in Job Shop Scheduling and Flow Shop Scheduling for minimize the makespan or total elapsed time. To determine the sequencing of a job we have applied Fuzzy C-Mean (FCM) clustering algorithm in both Job Shop Scheduling problem and Flow Shop Scheduling problem. Flow Shop Scheduling is a classified case of Job Shop Scheduling in which a specific job sequence is pursued strictly. Two illustrative examples of scheduling problems have been solved by this method and compared our results to some other existing methods discussed in the literature. The experimental result shows that the scheduling system using grid computing can allocate the makespan of service jobs effectively and more efficiently.

Keywords: Grid Computing, Job Shop Scheduling, Flow Shop Scheduling, Fuzzy C-Mean Algorithm (FCM).

I. INTRODUCTION

Grid computing is used to solve complex problem by group of independent computer systems. In Grid computing it combines the computers or servers, from multiple heterogeneous domains to solve single tasks that cannot be solved or can be difficult to solve by single personal computer. Grid is type of parallel and distributed systems that allows the aggregation, sharing and selection of geographically distributed resources dynamically depending on their availability, users QOS requirements, performance, capability and cost during run time. The fundamental idea of grid computing is to make use of the ideal CPU cycles and storage of millions of computer systems across a worldwide so that we can make a network which is flexible, pervasive and economical and which can be used by anyone easily who needs it. It is also defined as a service for sharing computing power and data storage capacity over the internet. The grid computing model allows companies to use a large number of computing resources on demand, no matter where they are located. Presently, grid computing has been widely accepted, studied and given attention by researchers.

A grid computing can be classified into three categories [25]-

1. Computational grid
2. Data grid
3. Service grid

In computational grid, number of servers or processors can be shared for high throughput applications and computation intensive computing, whereas the work of data grid is to collect the data from data archives such as digital libraries or data warehouses that are distributed in wide area network (WAN), and the work of service grid is to provide services which cannot be accomplished by single machine [25]. Demand grid, collaborative grid and multimedia grid are also known as the parts of service grid [25].

The basic grid model generally connected with number of processors and each of them are connected by a number of computational resources. Users, resources, resource broker and grid information service (GIS) are four important parts of grid models. To execute the jobs, users submit their jobs to the resource broker in the grid. Then resource broker splits the jobs into various tasks and distributes to several resources according to the user's requirements and availability of resources [17]. The work of GIS is to provide the information of all the resources to the meta-scheduler, which helps the resource broker for scheduling [17].

The job shop scheduling problem, which is also known as the machine sequencing problem is that of finding an optimal sequence for processing n-jobs on m-machines. In this type of problem the completion time of each operation is known and given operation on a given time has to be performed on a specified machine. Job shop scheduling is usually a strong NP- complete problem of combinatorial optimization problems and is the most typical one of the production scheduling problems. The job shop scheduling problem is a generalization of the classical job shop problem, in which finite jobs are to be processed on finite number of machines. The main goal of this method is to minimize the makespan or total elapsed time.

If we have n- jobs and m-machines, then in a job shop problem, n-jobs have to be processed on m-machines. The job shop problem is among the hardest combinatorial problem as compared with flow shop scheduling problem and travelling salesman problem. The flow shop scheduling problem is a restricted version of job shop scheduling problem.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Ajendra Kumar, Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya Haridwar (U.K), India. Email: ajendra.gkv@gmail.com
Preet Pal Singh, Department of Mathematics, Pt. L.M.S. (P.G) College, Rishikesh (U.K), India. Email: ppsingh003@gmail.com
Dipa Sharma, Department of Mathematics, S.D.M. Government (P.G) College, Doiwala (U.K), India. Email: dipa2014sharma@gmail.com
Pawan Joshi*, Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar (U.K), India. Email: pawanjoshi425@gmail.com

It is considered as one of the general production scheduling problems in which n-different jobs must be processed by m-machines in the same order. It should be noted that in flow shop scheduling the processing sequence of all jobs are same, that is all the jobs have to follow the same sequence. The flow shop scheduling problem with m-machine permutation is considered in which, each job i; $i = 1, 2, 3 \ldots n$ needs to be processed on each machine j; $j = 1, 2, 3 \ldots m$. each job has one operation on each machine and all job has the same ordering sequence on each machine at any time [1]. Each machine can process at most one job at a time [1] in which Pre-emption is not allowed. Our main aim is to find the processing order of n-jobs to minimizing the makespan. The capacity of the queue for each machine is unlimited. For n-number of jobs, $n!$ sequences are possible and for each machines the distinct possible schedules are $(n!)^m$.

This paper proposed a new method to solve Job Shop Scheduling and Flow Shop Scheduling in Grid Computing by FCM algorithm. The role of FCM is to find the job sequence in Job Shop Scheduling as well as in Flow Shop Scheduling problems. FCM algorithm is one of the most generally used fuzzy clustering algorithms. This technique was first introduced by J.C.Dunn in 1973 which was later on enhanced and improved by J.C.Bezdek in 1981. FCM has several uses in an extensive range of applications such as image analysis, medical diagnosis, shape analysis, target recognition, etc…

The paper is organized into different sections. Related work is reviewed in section 2. In section 3, framework of grid model is described and working procedure of model 1 and model 2 is discussed in section 4. We have solved some examples of Scheduling problem by model 1 and model 2. Comparison table of Job Shop Scheduling and Flow Shop Scheduling is presented in section 5. Finally we draw some conclusion in section 6.

II. RELATED WORK

Grid computing is a type of data management and computer infrastructure, designed as a support primarily for scientific research. Also used in various commercial concepts, business research, entertainment and also by government of different countries. Job scheduling in grid computing is a vast are of research which has been used by many researchers for their research purposes. The single service model and the multiple services model are used to predict the completion time of jobs in a service Grid [11]. The single service model provides only one type of service whereas; the multiple services model provides multiple types of services for the completion time of a job that runs in a Grid [11]. An ant colony optimization for dynamic job scheduling in grid environment had been presented by Lorpunmanee.S et.al [17]. In this paper, the authors had addressed the problem by developing a general framework of grid scheduling by using dynamic information and an ant colony optimization algorithm to improve the decision of scheduling [17]. Grid computing is defined as a service for sharing computing power and data storage capacity over the internet. Bhoyar.A.A et.al in 2015 developed a model of job scheduling in grid environment over IPV6, in which they established a cross regional scheduling mechanism to accomplish the job scheduling and job management in the grid computing and resources will be used more efficiently and effectively [7].

In job shop scheduling problem, each jobs are processed on each machines for sure amount of time [6]. GA is one of main heuristic techniques which have been used by many researchers in grid environment. A conventional GA for job shop problem has been presented by Nakano.R et.al [19]. In this paper the solution is represented as a binary genotype even though the job shop scheduling is an ordering problem. A two row chromosomes structure which is based on working procedure and machine distribution through GA to solve job shop scheduling problem has been described by Ye.Li, et.al [16]. Egon balas describes an implicit enumeration procedure that solves the problem by generating a sequence of circuit free graphs and solving slightly amended critical path problem or each graph in the sequence [3]. A family of algorithm for finding optimal schedules for job shop had been described by Barker.J.R et.al [4]. This algorithm is of branch and bound type but have a complete schedule associated with each node of the search tree. Branch and bound (BAB) algorithm is also known as primary algorithm to solve job shop scheduling problem. This algorithm used by so many authors such as Muth and Thompson 1963, Balas 1969, Mcmahon and Florian 1975, Barker and Mcmahon 1985, Carlier and Pinson 1989 and many more. Branch and bound method is based on one machine scheduling problem and is made more efficiently by several propositions which limit the search tree by using immediate selections [8, 9]. To solve flexible job shop scheduling problem, a Pareto approach based on the hybridization of fuzzy logic and evolutionary algorithm had been presented by Kaeem.I et.al [14]. This approach exploits the knowledge representation capabilities of fuzzy logic and adaptive capabilities of evolutionary algorithm. The objective is to minimize the makespan, total workload of machines and workload of the loaded machines. Storer.R.H et.al developed a method to solve job shop scheduling problem. In this paper the authors presented two methods, which are based on novel definition of solution spaces and of neighborhood in these spaces [21]. The proposed method integrate the problem which is common to operation research problem and uses local search approaches from artificial intelligence in order to obtain desirable properties from both of the two methods. To solve job shop scheduling problem Xia.W et.al proposed a hybrid optimization approach for multi-objective flexible job shop scheduling problem [24]. In this paper author combines two optimization techniques particle swarm optimization (PSO) and simulated annealing to solve the job shop scheduling problem.

In flow shop scheduling there are n-jobs and m- machines, where a strict sequence of operation is followed. A minimal downtime and minimal waiting time are the constraints in the continuous flow of process. Production facilities are generally found to be useful in flow shop scheduling problem. Baskar.A et.al developed a paper on flow shop scheduling by using a new heuristic algorithm using pascal's triangle method [5]. This paper made an attempt to find the best sequence of jobs so that the makespan is Minimum. The paper is easily coded in any high level language to run in a computer for effective and fast computation.

An O (n log n) time heuristic that is based on, Johnson's algorithm to solve flow shop scheduling problem had been presented by Chen.B et.al [10]. In this paper the authors generated a schedule by proposed algorithm which is 5/3 times that of an optimal schedule; thereby it reduces the previous best available worst case performance ratio of 2. Flow shop scheduling has been solved by so many optimization techniques, such as GA, artificial neural network, etc…. the use of artificial neural network (ANN) model applied to n-jobs, m-machines real flow shop scheduling problem with the objective to minimize the makespan [1]. An algorithm using a heuristic technique for sequencing n-jobs through m- machines in a flow shop under no wait constraint had been presented by Nailwal.K.K et.al [18, 15]. This algorithm describes a continuous processing of the jobs through the machines in flow shop scheduling without wait which is defined as No-wait flow shop scheduling [18].

III. FRAMEWORK OF GRID COMPUTING MODEL

The problem is stated as follows-

A set of n-jobs are to be processed on m-machines. Each job has its processing time, due date time, arrival time and release time. Our main aim is to find minimum makespan of the jobs which should be completed every machine exactly once. To solve the job shop and flow shop problem, we have developed two grid model named as, "Distributed grid computing model to solve scheduling problem by FCM algorithm" which is our model 1 and "Solving scheduling problem by FCM algorithm in supercomputing grid environment" which is our model 2. Now to understand both the model we should understand the grid computing environment first. The detailed information about grid environment is given as follows;

In our grid models the grid computing environment have grid clients, simulator, job dispatcher, grid information service (GIS), Meta scheduler, users or grid connected servers and grid job pool. Grid client's entities submit the jobs and detailed information to Meta scheduler and simulator. Where the simulator collects the information of machines about there waiting time, arrival time, release time from grid clients and passes the information to grid information service (GIS) and Meta scheduler collects the information of connected grid servers or users about their capacities or memory from grid clients and passes the information to GIS. GIS plays a very important role in our grid model, because it collects all the information about the model and passes meaningful information to job dispatcher. Job dispatcher urges the jobs according to the information given by GIS. Grid job pool is used by our model 2 in this paper. In grid job pool a server has been kept which can increase its capacity with the help of connected servers in grid computing environment, and hence it behaves like super computer which can complete the jobs in least time as possible. Now the detailed information about our model is given as follows;

Section 3.1: Working procedure of both the models

Let there are n-jobs which are processed on m- machines. We are solving job shop and flow shop problem so only one job is to be done at particular machines at one time. That is two or more jobs cannot be done at the same time on one machine. In both of our model we have used FCM algorithm to find the job sequence. The two of our model has following steps to find the makespan in job shop problem;

Step 1: Apply FCM algorithm to find cluster centers. Note that cluster centers should be at least 2. The number of job sequence is depending on the choice of our cluster centers. Suppose if we chose m-cluster centers then we get m-job sequence. In this paper we have chosen 2-cluster centers for each problem to reduce the complexity of the problem. FCM algorithm has the following steps to calculate the centers.

- choose random cluster centre at least 2,
- compute a membership matrix by following formula,

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}; \; for, 1 \le i \le c, 1 \le j \le n$$

; Where $\|x_i - c_j\|$ is the distance from point i to current cluster centre j

- calculate the new c-cluster centre by following formula,

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{\ m} x_i}{\sum_{i=1}^{N} \mu_{ij}^{\ m}}; \; for, 1 \le j \le C$$

After finding the centers from FCM algorithm, move to step 2, which has been described below.

Step 2: Find the job sequences according to their distance of centers, from the sum of all jobs in all machines represented by T_i, in ascending order. (Since we have chosen two cluster centers and therefore we will get two job sequences for each problem)

Step 3: After finding the job sequences find the makespan by our grid model 1 and model 2.

Step 4: since we have chosen 2-cluster centers, therefore we will get two makespan for two sequences. The final makespan is given by following formula;

Final makespan = minimum (makespan by C_1, makespan by C_2)

Model 1 and Model 2 are used to utilize the waiting time of particular jobs on particular machines. Waiting time can be calculated as:

$$\Delta_j = |(M_O)_i - (M_I)_{i+1}| ; where, \; j = 1,2,3,...n \; and \; i = 1,2,3,...m$$

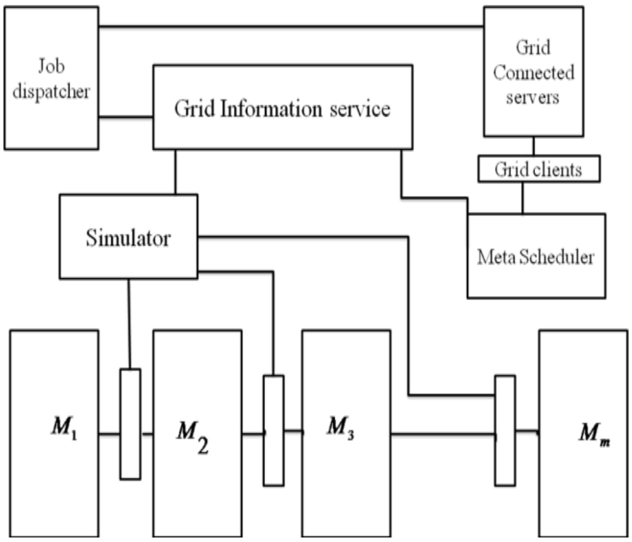
Δ_j is the waiting time of job j, $(M_O)_i$ is the OUT TIME of machine i and $(M_I)_i$ is the IN TIME of next machine.

Model 1 and Model 2 come in action if and only if, $\Delta_j > 0$.

Model 1 uses the waiting time by distributed grid computing, in which job dispatcher dispatches the jobs according to the user's capacity which are connected in grid environment, where as model 2 use the waiting time by grid job pool in which server which is kept in grid job pool behaves like supercomputer with the help of grid connected servers. Model 1 is named as, "Distributed grid computing model to solve scheduling problem by FCM algorithm" and Model 2 named as, "solving scheduling problem by FCM algorithm in supercomputing grid environment". Utilization of grid model 1 and model 2 has been given following.

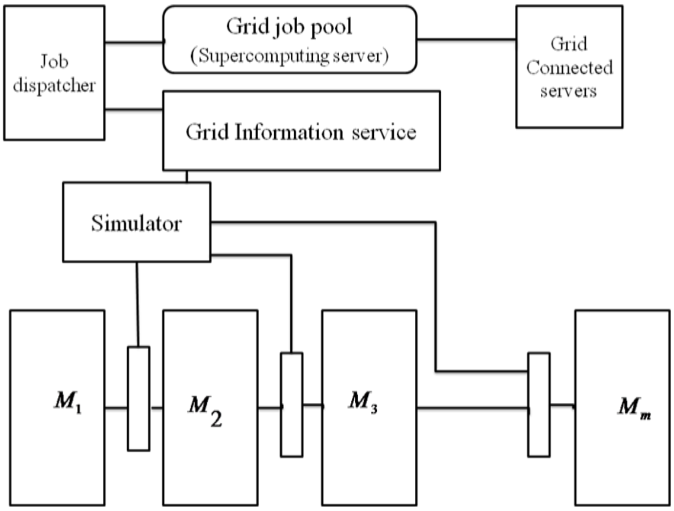
Working process of model 1 and model 2(solving job shop and flow shop scheduling problem by FCM algorithm in distributed grid computing and supercomputing grid environment) to find makespan:
After finding the job sequence with the help of FCM algorithm we have to calculate the makespan. Now in model 1, if we will get waiting time between the, "OUT time" of machine M_i and the, "IN time" of machine M_j, for each job J_k, where $i \neq j;\ i, j = 1, 2, 3, ..., m$ and $k = 1, 2, 3, ..., n$ then this waiting time can be used by our model 1 and model 2.

Utilization of waiting time in model 1: The information of waiting time, arrival time, and released time is given by grid clients (which are fixed between every machine) to simulator. After then simulator passes the information of machines to GIS which collects the information of connected grid servers or users capacity, passes the information to job dispatcher. Job dispatcher, dispatches the jobs to the connected servers on the basis of their capacity which matches the capacity of particular machines where the job is to be completed for a given time period. This should be done so that the continuity of the jobs could not break. If the job is finished before given time period then the IN time of a job equals the OUT time of job on the particular machine, and if job is not finished then the remaining work should be done on particular machine for the remaining time. Hence in this way the waiting time of a job can be utilized by model 1.



Distributed grid computing model (Model 1)

Utilization of waiting time in model 2: Similarly as in model 1, the information of waiting time, arrival time and release time has been given by grid clients to simulator and it passes the information to GIS. Where GIS also collects the information of connected grid servers and users and passes it to job dispatcher. But in this model, job dispatcher does not compare the capacities of connected grid servers with machines such as in model 1. Job dispatcher dispatches the job directly to grid job pool. Since in grid job pool there is a server which connected with grid servers and uses there memory such as RAM, ROM, etc… to increase its capacity so that it behave like supercomputer. And hence the job given by job dispatcher is finished always before time.



Super Computing Grid Environment (model 2)

In the next section we have solved some examples of job shop and flow shop scheduling problem by model 1 and model 2 and then compare the results by different authors in comparison table.

IV. JOB SCHEDULING PROBLEM BY MODEL1 AND MODEL 2

This section introduced the solutions of some examples of scheduling problem, by both of our Distributed grid computing model **(**model 1) and Super Computing Grid Environment (model 2).

Example 1- solving scheduling problem by model 1

Jobs	Machines				
	Machine 1	***Machine 2***	***Machine 3***	***Machine 4***	***Machine 5***
Job 1	6	4	1	2	8
Job 2	5	5	3	4	9
Job 3	4	3	4	5	7
Job 4	7	2	2	1	5

This is 4-jobs and 5-machines problem. Our aim is to find the minimum makespan by model 1.

Solution- to compute the minimum makespan by model 1, we have to use following steps;

Step 1- calculate $T_i; T_i$ = sum of i^{th} row; where, $i = 1,2,3,4$

$$T_1 = 6 + 4 + 1 + 2 + 8 = 21$$

$$T_2 = 5 + 5 + 3 + 4 + 9 = 26$$

$$T_3 = 4 + 3 + 4 + 5 + 7 = 23$$

$$T_4 = 7 + 2 + 2 + 1 + 5 = 17$$

Step 2- apply FCM algorithm to find cluster centres, note that cluster centers should be at least 2. In this problem we have taken two cluster centers.

S.No	C_1	C_2
1.	0.5891	0.4109
2.	0.0684	0.9316
3.	0.1232	0.8768
4.	0.9599	0.0401

; C_1 =18.1799 and C_2 =24.2483

Step 3- after finding cluster centers C_1 and C_2, find the job sequence according to their distance of each T_i from the centers in ascending order. Note that for $n \geq 2$ cluster centers, $\exists\, n \geq 2$ job sequences. In this problem we have taken 2-cluster centers therefore ∃ 2-job sequences.

Hence for C_1, job sequence is (4, 1, 3, 2) and for C_2, job sequence is (3, 2, 1, 4).

Step 4- now for both job sequences we have to calculate the makespan in grid environment by model 1, which has been describes below;

For job sequence (4, 1, 3, 2),

	M_1		M_2		M_3		M_4		M_5	
	I	**O**	**I**	**O**	**I**	**O**	**I**	**O**	**I**	**O**
J_4	0	7	7	9	9	11	11	12	12	17
J_1	7	13	13	17	17	18	18	20	20	28
J_3	13	17	17	20	20	24	24	29	29	36
J_2	17	22	22	27	27	30	30	**34**	**36**	43

Here, $\Delta_2 = |(M_O)_4 - (M_I)_5| = |34-36| = 2\,units$, that is for J_2, there is a waiting time of 2 units between M_4 and M_5, which has been utilized by our model 1.

Therefore OUT TIME of $M_5 = 36+9-2 = 43$ Which is makespan for this sequence are **43 units.**

Now to find makespan for second sequence (3, 2, 1, 4)

	M_1		M_2		M_3		M_4		M_5	
	I	**O**	**I**	**O**	**I**	**O**	**I**	**O**	**I**	**O**
J_3	0	4	4	7	7	11	11	16	16	23
J_2	4	9	9	14	14	17	17	**21**	**23**	30
J_1	9	15	15	19	19	**20**	**21**	**22**	**30**	30
J_4	15	22	22	24	24	26	26	**27**	**30**	32

$$Here,\ \Delta_2 = |(M_O)_4 - (M_I)_5| = |21-23| = 2\,units,$$

$$\Delta_1 = |(M_O)_3 - (M_I)_4| = |20-21| = 1\,units,\ and$$

$$\Delta_1 = |(M_O)_4 - (M_I)_5| = |22-30| = 8\,units,$$

$$\Delta_4 = |(M_O)_4 - (M_I)_5| = |27-30| = 3\,units,$$

Where, Δ_2, Δ_1 *and* Δ_4 are the waiting time for job 2, 1 and 4 respectively,

Hence makespan for sequence (3, 2, 1, 4) is **32 units.**

Step 5-

Final makespan = min (makespan by C_1, makespan by C_2)

= min (43, 32) = **32 units.**

Example 2- solving scheduling problem by model 2:

Jobs	machines			
	Machine 1	***Machine 2***	***Machine 3***	***Machine 4***
Job 1	7	15	14	21
Job 2	11	18	18	6
Job 3	2	13	11	16
Job 4	14	4	27	14
Job 5	18	11	32	16

Solution- The first three steps are same as model 1.

Hence on applying FCM we get C_1 and C_2 as follows;

S.No	C_1	C_2
1.	0.9346	0.0654
2.	0.6602	0.3398
3.	0.9569	0.0431
4.	0.0611	0.9389
5.	0.0489	0.9511

Where, C_1 =50.0879 and C_2 =67.1508

$$T_1 = 57$$
$$T_2 = 53$$
$$T_3 = 42$$
$$T_4 = 59$$
$$T_5 = 77$$

Hence job sequence for C_1 is (2, 1, 3, 4, 5) and for C_2 is (4, 5, 1, 2, 3).

Makespan by C_1 :

	M_1		M_2		M_3		M_4	
	I	**O**	**I**	**O**	**I**	**O**	**I**	**O**
J_2	0	11	11	29	29	47	47	53
J_1	11	**18**	**29**	**29**	**47**	**47**	**53**	53
J_3	18	**20**	**29**	**29**	**47**	**47**	**53**	53
J_4	20	34	34	**38**	**47**	**47**	**53**	53
J_5	34	52	52	63	63	95	95	111

Makespan for sequence (2, 1, 3, 4, 5) is **111 units.**

Makespan by C_2 :

	M_1		M_2		M_3		M_4	
	I	*O*	*I*	*O*	*I*	*O*	*I*	*O*
J_4	0	14	14	18	18	45	45	59
J_5	14	32	32	**43**	**45**	**45**	**59**	59
J_1	32	**39**	**43**	**43**	**45**	**45**	**59**	59
J_2	39	50	50	68	68	86	86	92
J_3	50	**52**	**68**	**68**	**86**	**86**	**92**	92

Makespan for sequence (4, 5, 1, 2, 3) is **92 units.**

Final makespan = min (makespan by C_1, makespan by C_2)

= min (111, 92) = **92 units.**

Similarly we have solved different examples of job shop scheduling and flow shop scheduling from different papers and compare their results with our results, which has been shown by comparison table in the next section.

V. EXPERIMENTAL COMPARISON

In this section we have shown the comparison graph of flow shop scheduling by Fig 1 and comparison graph of job shop scheduling by Fig 2.1 and Fig 2.2, whereas table 1 shows the comparison of flow shop problems and Table 2.1 and 2.2 shows the comparison of job shop scheduling problem.

- **Comparison graph and table of Flow Shop Scheduling and Job Shop Scheduling.**

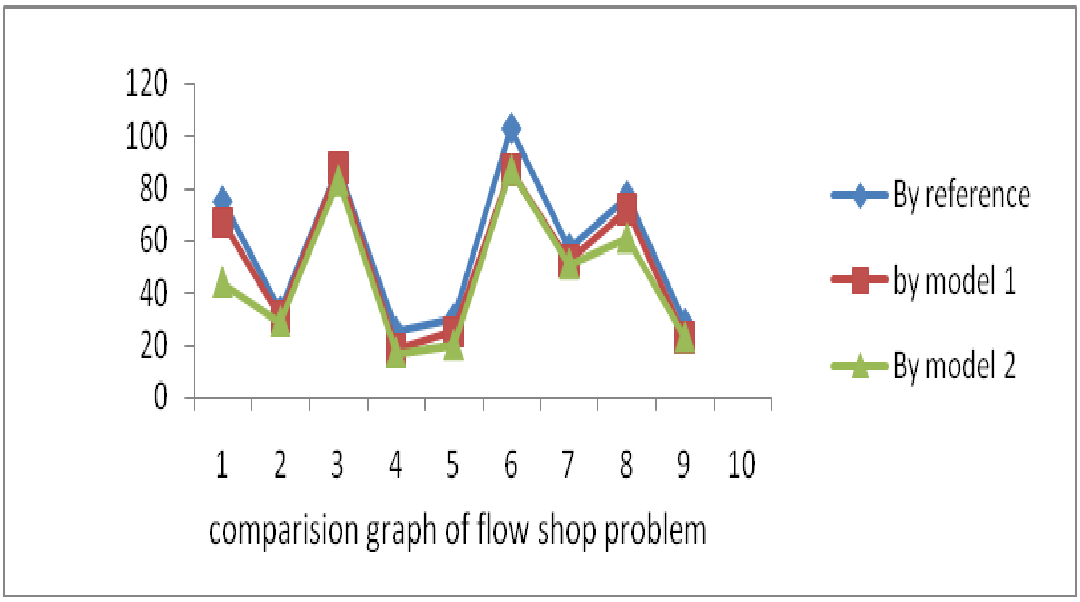


Fig 1

Table 1

S.No	Job×Machine	References	Makespan by references(in units)	**Makespan by our proposed model**	
				Makespan by model 1(in units)	Makespan by model 2(in units)
1.	6×4	Laha.C et.al [15]	75	67	44
2.	4×5	Gupta.J.N.D [12]	33	31	29
3.	6×3	Szwarch [22]	88	88	83
4.	5×3	Nailwal.K.K et.al [18]	25	18	17

5.	4×4	Baskar.A et.al [5]	1) gupta heuristic – 31 2) RA- heuristic- 31 3) CDS- 30 4) Palmar- 30 5) Baskar- 30	25	20
6.	10×10	Baskar.A et.al [5]	1) gupta heuristic - 103 2) RA- heuristic- 97 3) CDS- 102 4) Palmar- 99 5) Baskar- 103	87	87
7.	4×5	Akyol.D.E et.al [1] And Nawaz [20]	[1]- 57 [20]- 54	52	51
8.	5×5	Krishna.V.P et.al [23]	1) CDS- 79 2) Krishna.V.P et.al- 77	72	61
9.	6×3	Chen.B [10]	28	23	23

From this table it clearly indicates that our proposed model is much better to solve flow shop scheduling problem.

Now in next comparison table we have compared job shop scheduling problem by our proposed models.

- **Comparison table of job shop scheduling problem:**

Comparison table of 6×6 matrix problem:

Table 2.1

S.No	Papers	Algorithm used	Makespan
1.	Balas 1969 [3]	BAB	55
2.	Mcmahon 1975	BAB	55
3.	Barker 1985 [4]	BAB	55
4.	Carlier 1989 [9]	BAB	55
5.	Nakano 1991 [19]	GA	55
6.	By us	By model 1- By model 2-	42 35

Where BAB stands for branch and bound algorithm, and GA stands for genetic algorithm.

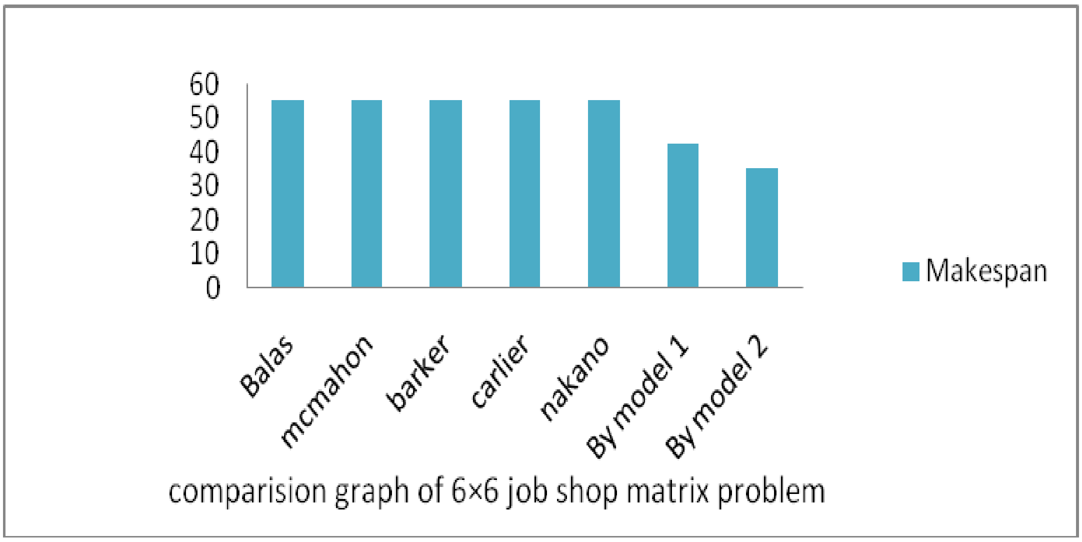


Fig 2.1

- **Comparison table of different order matrices in job shop scheduling:**

Table 2.2

S.No	Job × machine	Makespan (Total elapsed time)			
		By SAI method [13]	By Johnson's method	By model 1 (our proposed model)	By model 2 (our proposed model)
1.	4×5	43	43	32	32
2.	5×2	37	40	34	26
3.	5×3	40	42	33	33
4.	5×4	125	127	102	92
5.	4×4	122	122	122	122
6.	4×6	133	133	129	129

Hence from this comparison table we can say that these two models are very helpful to solve job shop problem and flow shop problem.

Fig 2.2

VI. CONCLUSION

In this paper we have presented two models in grid computing to solve Job Shop Scheduling and Flow Shop Scheduling problem with the help of FCM algorithm. The role of FCM algorithm in both of the models is to find the job sequence. In model 1 we have used distributed Grid computing whereas supercomputing grid environment have been used to utilize the waiting time of a job on a particular machine for a fixed time period where, dispatching of jobs to the connected grid servers is done by job dispatcher for fixed time period.

The experimental results for both Job Shop Scheduling and Flow Shop Scheduling by model 1 and model 2 are better as compared with other scheduling algorithms. The main aim of this study is to introduce a new efficient method for solving scheduling problem such as, Job Shop Scheduling and flow Shop Scheduling problem which also shows the applicability of grid computing to scheduling problems.

REFERENCES

1. Akyol D.E., "Application of Neural Network to Heuristic Scheduling Algorithm". Computers and Industrial Engineering, Vol- 46, 679-696, 2004.
2. Ashok A.S., and Hari D.P., "Grid Computing: Various Job Scheduling Strategies". International Journal of Computer Applications, 27-30, 2012.
3. Balas E., "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm". 941-957, 1968.
4. Barker J.R., and Mcmahon G.B., "Scheduling the General Job Shop". Management Science, Vol- 3(5), 594-598, 1985.
5. Baskar A., and Anthony X.M., "A New Heuristic Algorithm using Pascal's Triangle to Determine more than one Sequence having optimal or Near optimal Makespan in Flow Shop Scheduling Problems". International Journal of Computer Application, Vol- 39(5), 9-15, 2012.
6. Bhatt N., and Chauhan N.R., "Genetic Algorithm Application on Job Shop Scheduling Problem". International Conference on Soft Computing Techniques and Implementation, 7-14, 2015.
7. Bhoyar A.A., and Dharmik R.C., "Design and Implementation of Job Scheduling in grid Environment over IPV6". International Journal of Computer Science and Mobile Computing, Vol-4(4), 243-250, 2015.
8. Brucker P., Jurisch B., and Sievers B., "A Branch and Bound Algorithm for the Job Shop Scheduling Problem". Discrete Applied Mathematics, Vol-49, 107-127, 1994.
9. Carlier J. and Pinson E., "An Algorithm for Solving the Job Shop Problem". Management Science, Vol- 35(2), 164-176, 1989.
10. Chen B., Glass C.A., Potts C.N. and Struscrich V.A., "A New Heuristic for Three Machine Flow Shop Scheduling". Operation Research, Vol-44(6), 891-898, 1996.
11. Gao Y., Rong H., and Huang J.Z., "Adaptive Grid Job Scheduling with Genetic Algorithm". Future Generation Computer System, Vol- 21, 151-161, 2005.
12. Gupta J.N.D., "Optimal flow Shop Schedules with no Intermediate storage Space". 235-243, 1976.
13. Gupta S., Ali I., and Ahmed A., "A New Algorithm for Solving Job Shop Sequencing Problem". International Journal of Computer Science Engineering, Vol- 5(2), 93-100, 2016.
14. Kacem I., Hammadi S., and Borne P., "Pareto-Optimality approach for flexible Job Shop Scheduling problems: Hybridization of Evolutionary Algorithm and Fuzzy Logic". Mathematics and Computers in Simulation, vol- 60, 245-276, 2002.
15. Laha D., and chakraborty U.K., "A Constructive Heuristic for Minimizing Makespan in no wait Flow Shop Scheduling". International Journal in Advance Manufacturing Technology, Vol- 41, 97-109, 2009.
16. Li Y., and Chen Y., "A Genetic Algorithm for Job Shop Scheduling". Journal of Software, Vol-5(3), 269-274, 2010.
17. Lorpunmanee S., Sap M.N., Abdullah A.H. and Inwai C.C., "An Ant Colony Optimization for Dynamic Job Scheduling In grid Environment". Vol-1(5), 1343-1350, 2007.
18. Nailwal K.K., Gupta D., and Sharma S., "Minimizing Makespan in Flow Shop under no wait Restriction on Jobs". International Conference on Electrical, Electronics and optimization Techniques, 2016.
19. Nakano R., "Conventional Genetic Algorithm for Job Shop Problems". ICGA, 474-479, 1991.
20. Nawaz M., and Ham E.E.E.I., "A Heuristic Algorithm for the m-Machines, n-Jobs Flow Shop Scheduling Problem". The International Journal of Management Science, Vol- 11(1), 91-95, 1983.
21. Storer R.H., wu S.D., and Vaccari R., "New Search Spaces for sequencing problems with Application to Job Shop Scheduling". Management Science, Vol- 38(10), 1495-1509, 1992.
22. Szwarch W., "Solvable cases of the Flow Shop Problem without Interruptions in Job Processing". Naval Research Logistic quarterly, vol- 30, 179-183, 1983.
23. Velpuri G.K., and Vijay P., "A Proposed method for Permutation Flow Shop Scheduling". International journal of Emerging Technology and Advance engineering, Vol- 4(3), 209-213, 2013.
24. Xia W., and Wu Z., "An Effective Hybrid optimization Approach for Multi-objective Flexible Job Shop Scheduling Problems". Computers and Industrial Engineering, Vol- 48, 409-425, 2005.
25. Yadav K., Jindal D., and Singh R., "Job Scheduling In Grid Computing". International Journal of Computer Application, Vol-69(22), 13-16, 2013.

AUTHORS PROFILE

Dr. Ajendra Kumar is an assistant professor at the department of mathematics and statistics in Gurukula Kangri Vishwavidyalaya, Haridwar. He received his master degree and completed his Ph.D. from Gurukula Kangri Vishwavidyalaya Haridwar. His area of research work includes Mathematical Modeling and Graph Theory. He has more than 20 Research publications, 4 edited book and one DST sponsored research project on "Impact of Industrialization and human risk with special reference to Heavy metals". He is also an editorial board member of journal of applied and natural science foundation.

Dr. Preet Pal Singh is an assistant professor at the department of mathematics in Pt. L.M.S. Govt (P.G) College Rishikesh, Dehradun (U.K). He received his master degree in applied mathematics from IIT Roorkee and completed his Ph.D. from Gurukula Kangri Vishwavidyalaya, Haridwar. His area of research includes Distributed Systems, Task Allocation, Fuzzy Theory and Mathematical Modeling.

Dr. Dipa Sharma is an assistant professor at the department of mathematics in S.D.M Govt (P.G) College Doiwala, Dehradun (U.K). She did M Sc and M Phil from C.C.S University Meerut and completed her Ph.D. from Gurukula Kangri Vishwavidyalaya, Haridwar. Her area of research includes Mathematical Modeling, Fuzzy Theory and Optimization Techniques.

Pawan Joshi is currently pursuing PhD from Gurukula Kangri Vishwavidyalaya Haridwar. He received his master degree from Pt.L.M.S Govt (P.G) College, Rishikesh (U.K). His area of research work includes Grid Computing, Fuzzy theory and task Allocation. He is currently working on neural network to solve some real life problems in grid computing system.