

A Fuzzy Based Mechanism For Allocation of Grid Resources

¹Poonguzhali, M. and ²S. Shanmugavel

¹Department of ECE, Narasu's Sarathy Institute of Technology, Poosaripatty, Salem-636305, India

²Department of Electronics and Communication Engineering, Anna University, Chennai-600 025, India

Received 2012-05-15, Revised 2012-09-05; Accepted 2013-05-23

ABSTRACT

One of the main challenges in Grid computing is efficient allocation of resources (CPU-hours, network bandwidth) to the tasks submitted by users. In our previous work a technique to allocate resources in a grid environment using predicted data has been proposed. We propose utilization of the predicted data the resources were classified into three types; they are permanent resources, semi-permanent and sporadic resources. These types of resources may become available for a time that is either higher than the dwelling time or lower than the dwelling time in a grid environment. As the nature features are not known in such classification and then allocation mechanism, the performance cannot be increased further. In order to avoid such problem, in this study, a prediction model and an allocation factor are introduced. These parameters are determined for the sporadic type and semi-permanent type of resources and they are used in the fuzzy-based resource allocation mechanism. The incorporation of these parameters in the resource allocation leads to a remarkable resource utilization rate and makespan. This can be observed from the simulation and comparative results. From the results, it can be said that the proposed resource allocation mechanism has proved the performance in a dynamic environment.

Keywords: Fuzzy Inference System (FIS), Utilization in Terms of Minimum Makespan (UMM), Prediction Model, Allocation Factor, Semi-Permanent

1. INTRODUCTION

Grid Computing is a form of distributed computing that involves coordination and sharing of computing application, data storage or network resources across dynamic and geographically dispersed organizations. (Rafee and Rahimzadeh, 2009; Richard *et al.*, 2008; Vijaya *et al.*, 2009; Abba *et al.*, 2012). Grid can be distinguished from conventional distributed computing by its focus on large scale resource sharing, high performance and solving compute/data intensive applications. Grid supports researchers and scientists from diverse organizations to share information, instruments, data and compute and storage resources dynamically in a flexible and secure manner (Vijaya *et al.*, 2009; Puri and Dev, 2012). It is a reliable technology for the process of making scheduling decisions involving allocating jobs to resources over multiple administrative

domains. Grid Environment is usually viewed as a hierarchical problem with two levels. The first level, called meta-scheduling, a grid Environment selects the resources to be used by a job. At second level, called local scheduling, a local scheduler schedules the jobs assigned to it (Naisipour *et al.*, 2008; Christodoulopoulos *et al.*, 2009). The pool of resources can be assumed fixed or stable in traditional parallel and distributed computing environments, in a Grid dynamicity exists in the networks and computational resources. Scalability and adaptability are two important factors that must be taken into account in setting up a grid system. First, a network shared by many execution domains cannot provide guaranteed bandwidth. This is particularly true for Wide-Area Networks like internet. Second, both the availability and capability of computational resources will exhibit dynamic behavior (Yien *et al.*, 2011; Wankar, 2008).

Corresponding Author: Poonguzhali, M., Department of ECE, Narasu's Sarathy Institute of Technology, Poosaripatty, Salem-636305, India

Developing middleware that offers basic functionality for example, the capability to query for information regarding the resources and the capability to schedule jobs onto the resources is done by majority of the works in Grid computing. Dynamic competitive capabilities offer value for a long period of time comprising competitive advantages obtained from customs and policies of different organizations (Darby and Tzeng, 2010; Suwan *et al.*, 2012a). Two significant agent types present in the environment, which is fundamentally a multi-agent system, are resource agents and user agents. In majority of the cases, users can be regarded as individual agents that generate jobs and try to access resources for their execution, or as external resource brokers that map jobs in favor of other individual users. A multi-agent architecture that addressed resource management and application execution with support for Quality of Services (QoS) in grid scheduling algorithms is presented by (Keerthika and Kasthuri, 2012; Marowka, 2000).

According (Boukerram and Azzou, 2006) to job behavior in the job waiting queue is considered as an important factor for scheduling algorithm. The data access cost is also aggregated with the job waiting queue in order to reduce the job turnaround time. Grid scheduler is responsible for receiving jobs from grid users, selects feasible resources for those jobs according to acquired information and finally generates jobs to resource mappings. The number and speed of the available processors, system memory, as well as storage space are generally used to describe the resources. Resource-sharing jobs that must be mapped utilizing a resource allocation system to respective resource providers exist with users (Latip *et al.*, 2011; Odeh *et al.*, 2009; Farooq *et al.*, 2009).

A small number of researchers have addressed the problem from the perspective of learning and adaptation, even though considerable attention has been paid to the resource allocation problem in Grid computing (Senthilnathan and Purusothaman, 2012; Suwan *et al.*, 2012b). At the same time, the possibility of effectively solving the resource allocation problems using groups of autonomous learning agents has been proved by Multi-Agent Systems (MAS) and distributed AI communities. The following are present in certain resource allocation problems: (1) identification of a proper service and the resources, (2) allocating the resources based on specific conditions like pricing or priority dynamical allocation and updating of the status of the resources (Kamalam and Bhaskaran, 2012).

In order to analyze the utilization measure achieved in the grid resource allocation mechanism, we have reviewed the recent related works, in which it has been found out that the resource allocation mechanism achieves good utilization when using fuzzy logic. However, the lack of consideration of dynamic nature and forecasted allocation in those works claims that the achieved utilization is not reliable. Hence, to solve the issue we have proposed a new classification strategy along with a simple-fuzzy based resource allocation mechanism in the previous paper (Poonguzhali and Shanmugavel, 2011). But in the previous paper, the resource allocation was based on historical data and the classification was considered as from forecasted data. This is found to be a bottleneck for the performance improvement of the proposed grid resource allocation mechanism. Hence, in this study, we introduce a Prediction Model and Allocation Factor, which can be calculated from standard distribution functions and they are incorporated in the fuzzy-based resource allocation mechanism (Poonguzhali and Shanmugavel, 2011). Thus obtained Prediction Model and Allocation Factor based Fuzzy grid resource allocation mechanism improves the resource utilization with minimum make span.

1.1. Related Work

Though plenty of related works are available in the literature, a handful of significant highly related works are reviewed here (Kamalam and Bhaskaran, 2012) have proposed grid architecture as a collection of clusters with multiple worker nodes in each cluster, where the resources may join or leave the environment at any time and the jobs also arrives at different intervals of time. In this proposed method, the dynamic environment, to maximize the resource utilization and to minimize the makespan an effective grid scheduling technique was needed. They proposed a new scheduling algorithm Novel Adaptive Decentralized Job Scheduling Algorithm (NADJSA) that applies both Divisible Load Theory (DLT) and Least Cost Method (LCM) and also considers the user demands. The proposed Novel Adaptive Decentralized Job Scheduling Algorithm was compared with the Decentralized Hybrid Job Scheduling Algorithm. Conclusion: The proposed Novel Adaptive Decentralized Job Scheduling Algorithm minimizes the makespan, improves the resource utilization and satisfies the user demands and well suits for the grid environment. Their experimental results have proved that optimization scheduling allocates the cheapest resources to ensure that the deadline can be met and computation is minimized.

Jiang *et al.* (2008) have proposed scheduling algorithms for batch-mode data-intensive jobs is a key issue in data-intensive Grid applications. It focuses on how to minimize the overhead of transferring the required data set to the executing grid site. Existing approaches pay attention to the access cost of a data-intensive job at each executing grid site for replicating the required data set. In this Scheduling algorithm they presented method from potential behaviors of jobs in the waiting queue at each grid site when the access cost was evaluated. The algorithm has mainly examined the influence of potential behaviors on the access cost and proposes a data-intensive job scheduling algorithm with potential behaviors. Furthermore, the main focus of our algorithm lies on paper shows that it has better performance in mean job time of all jobs, total number of replications, total number of local files accesses and effective network usage than the scheduling algorithm based on access cost.

Chapman *et al.* (2007) have predicted the CPU resource utilization using their predictive grid scheduling framework, which follows Kalman filter theory. Their experimental results have proved that they have achieved 15-20% precision in their prediction. The subsequent observation of utilization has also confirmed the enhancement of scheduling quality, when compared with the other approaches.

Ramesh and Krishnan (2012) have presented an optimal resource sharing algorithm in Grid Computing. Resource sharing required more optimized algorithmic structure, otherwise the waiting time and response time are increased and the resource utilization is reduced. In order to avoid such reduction in the performances of the grid system, an optimal resource sharing algorithm is required. Ramesh and Krishnan (2012) have introduced a utility function in optimal resource sharing algorithm. In this paper, a hybrid algorithm for optimization of load sharing was proposed. The hybrid algorithm contains two components which were Hash Table (HT) and Distributed Hash Table (DHT). The algorithm has mainly examined the relationship of optimal resource sharing between optimization tasks and load sharing of existing systems.

Chen and Lu (2008) have introduced a utility function in grid resource scheduling algorithm. The algorithm has mainly examined the relationship between the execution time, cost and the user utility function to solve the heterogeneity issue of user requirements in grid resource allocation. The algorithm has accomplished a good performance and it has compromised the drawbacks in time-based optimization algorithms and cost-based optimization algorithms.

Wankar (2008) have proposed Open Grid Forum (OGF) is an organization that resulted from the merger of the Global Grid Forum (GGF) and the Enterprise Grid Alliance (EGA). GGF was an international organization that started in 1999, with the focus on the development of open standards for grid software interoperability, common practices, agreements and other related issues and proposed several specifications with the help of several working groups. The grid project started with the aim of using high-end computational resources, networks, databases and scientific instruments owned and managed by multiple organizations. Globus was one of the most successful projects in grid computing to test these specifications. Although it could overcome many technological barriers, many were still remains as open questions. In this proposed method, they discussed about grid, Globus Toolkit and present some of technical challenges the grid community faces. Further, they provided future research directions in Grid Computing.

Farooq *et al.* (2009) have simulated a new middleware framework for Grids that achieves user satisfaction by providing QoS guarantees for Grid applications. In this proposed method, they providing Scalability, flexibility, quality of service provisioning, efficiency and robustness were the desired characteristics of most computing systems. Although the emerging Grid computing paradigm was scalable and flexible, achieving both efficiency and quality of service provisioning in Grids was a challenging task but is necessary for the wide adoption of Grids. Grid middleware should also be robust to uncertainties such as those in user-estimated runtimes of Grid applications. In this proposed method, they presented a complete middleware framework for Grids that achieves user satisfaction by providing QoS guarantees for Grid applications, cost effectiveness by efficiently utilizing resources and robustness by intelligently handling uncertain runtimes of applications. Finally, they have validated the experimental results and have proven their performance of resource utilization of the grid with a high success rate of jobs and reduction in the total execution time of submitted jobs.

2. MATERIALS AND METHODS

The proposed technique works mainly from the historical involvement of every grid resources and the dwelling time of the resources in the grid. As the reliability fact for the permanent and the semi-permanent type of resources are high, the only consideration is with the sporadic type of resources. The sporadic resources are the resources whose dwelling time cannot be predicted by any means.

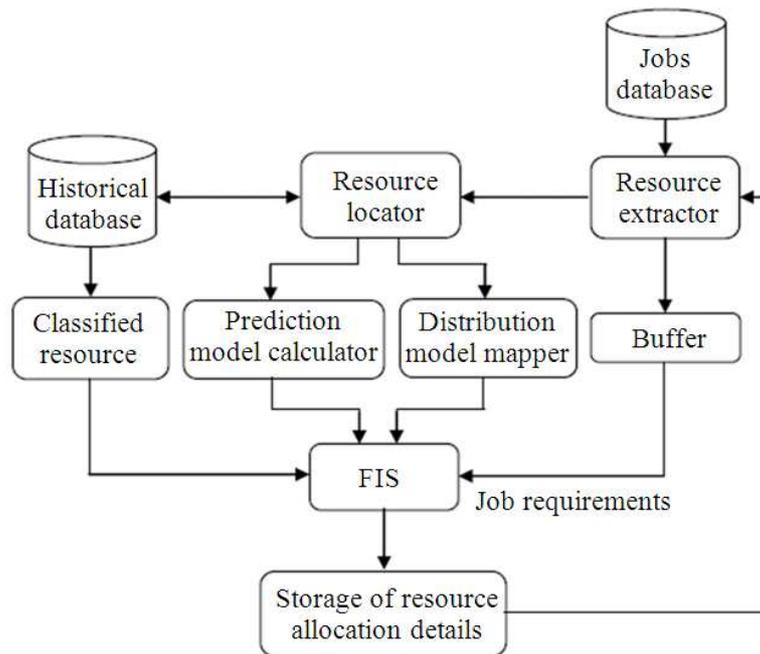


Fig. 1. Architectural view of the proposed resource allocation mechanism

Hence, in this study they are considered as the random variables and accordingly, an allocation factor is determined. The allocation factor is considered in the enhanced version of the fuzzy-based grid resource allocation technique to allocate the grid resources to the submitted jobs with a remarkable level of allocation performance. Moreover, a prediction model is generated from the dwelling time of the sporadic resources of past time slots. A factor is considered from the model, which is so called as model factor. Even though, the semi-permanent can be considered as reliable, the lesser dwelling time may leads the semi-permanent type of resources as sporadic in the future. To avoid such circumstances, the prediction model as well as allocation factor is also considered for the semi-permanent type of resources. It could be clearly understandable from the overall structure of the proposed technique, which is given in the Fig. 1.

Again the heart of the architecture is Fuzzy Inference System along with the compendium of processing of members. Mainly, two databases involved in the architecture namely Jobs database and Historical database. The jobs database holds the details of the jobs that are to be submitted to the grid whereas historical database holds the dwelling time of the resources that were available at the time of previous time slots. The

resource extractor extracts the individual resources that are required by the jobs and sends it for the further blocks for allocation. The extractor holds the resource, its requirement duration and the job, which requires the resource and the priority of the job. The buffer holds the resource and the corresponding details that are to be subjected to allocation mechanism. When the allocation is performed for the resource that is in the buffer, the buffer will be cleared and then loaded by new resource and its details.

The resource locator locates the required resource in the historical database and the extracts the dwelling time of the particular resource of the previous time slots. The prediction model calculator generates a prediction model and calculates the model factor where as the distribution model mapper maps the distribution of the resources with the standard distribution functions and determines the allocation factor. The classified resource mainly contains the resources under three classes, namely, permanent, semi-permanent and sporadic. All the resultants from the blocks, namely, dwelling time of the classified resources, Prediction model calculator, Distribution model mapper and the Job buffer, are given to the Fuzzy Inference System to make a decision about whether the job can be allocated to the resource or not.

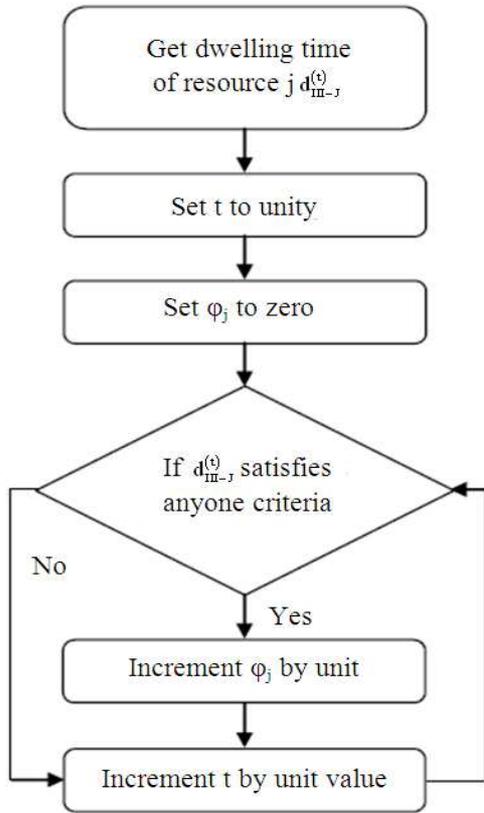


Fig. 2. Flowchart for determining ϕ_j

The major contribution relies on predicting the resource information and determining the allocation factor using the random distribution functions. Let the availability of the grid resources in the past N time slots be $[R_{i-j}^{(0)} @ d_{i-j}^{(0)}]$, $[R_{ii-j}^{(0)} @ d_{ii-j}^{(0)}]$ and $[R_{m-j}^{(0)} @ d_{m-j}^{(0)}]$, where, $0 \leq t \leq N-1$ and $j \in [0, N_R - 1]$. Here, N_R is the number of maximum number of resources available in the grid. In the proposed methodology, a runtime prediction model is generated for the sporadic and semi-permanent type of resources even though we have a classification model is available for all type of resources. In parallel with the prediction model, an allocation factor is determined. Both the prediction model and the allocation factor are determined only at the time of processing or analyzing a resource to allocate for the submitted jobs.

2.1. Determining the Prediction Model

When a resource is given, a prediction model is developed for the particular resource and the upcoming dwelling time is determined using the prediction model.

The prediction model mainly consists of two steps, namely, determining data variation factor ϕ and generating the model.

2.2. Determining Data Variation Factor ϕ

For every resource to be allocated, a data variation factor is determined based on the dwelling time of the particular resource in the past N time slots. ϕ_j , where, J refers to the resource ID, is determined mainly based on six criteria, which are given below. The process of determining ϕ_j is described as a flowchart in Fig. 2.

Criterion 1:

- (a) $d_{m-j}^{(0)} > d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} > d_{m-j}^{(t+1)}$

Criterion 2:

- (a) $d_{m-j}^{(0)} < d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} < d_{m-j}^{(t+1)}$

Criterion 3:

- (a) $d_{m-j}^{(0)} < d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} = d_{m-j}^{(t+1)}$

Criterion 4:

- (a) $d_{m-j}^{(0)} > d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} = d_{m-j}^{(t+1)}$

Criterion 5:

- (a) $d_{m-j}^{(0)} = d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} > d_{m-j}^{(t+1)}$

Criterion 6:

- (a) $d_{m-j}^{(0)} = d_{m-j}^{(t-1)}$
- (b) $d_{m-j}^{(0)} < d_{m-j}^{(t+1)}$

2.3. Generation of Polynomial

Once ϕ_j is determined, a polynomial equation is generated, which is further used as the prediction model. The polynomial equation is generated in such a way that it should have the degree of ϕ_j . The equation can be given as:

$$\phi_j = a_N^{(j)} t_j^N + a_{N-1}^{(j)} t_j^{N-1} + a_{N-2}^{(j)} t_j^{N-2} + \dots + a_2^{(j)} t_j^2 + a_1^{(j)} t_j^1 + a_0^{(j)} \quad (1)$$

Equation (1) is solved for the coefficients by substituting different past dwelling times and so the final solved equation is obtained. This is used as the prediction model to determine the further dwelling times of the upcoming period. The similar process is done with the semi-permanent type of resources.

	x ₁	x ₂	x ₃	x ₄	x ₅	Y
1	min	min	min	min	min	min
2	min	min	min	min	max	max
3	min	min	min	max	min	max
4	min	min	min	max	max	max
5	min	min	max	min	min	min
6	min	min	max	min	max	max
7	min	min	max	max	min	max
8	min	min	max	max	max	max
9	min	max	min	min	min	min
10	min	max	min	min	max	min
11	min	max	min	max	min	min
12	min	max	min	max	max	min
13	min	max	max	min	min	min
14	min	max	max	min	max	max
15	min	max	max	max	min	max
16	min	max	max	max	max	max
17	max	min	min	min	min	min
18	max	min	min	min	max	min
19	max	min	min	max	min	min
20	max	min	min	max	max	min
21	max	min	max	min	min	min
22	max	min	max	min	max	max
23	max	min	max	max	min	max
24	max	min	max	max	max	max
25	max	max	min	min	min	max
26	max	max	min	min	max	max
27	max	max	min	max	min	max
28	max	max	min	max	max	max
29	max	max	max	min	min	min
30	max	max	max	min	max	min
31	max	max	max	max	min	min
32	max	max	max	max	max	max

Fig. 3. Fuzzy rules generated for the proposed resource allocation mechanism

2.4. Determining the Allocation Factor

In order to determine the allocation factor α , the dwelling time of a particular resource is assumed to be distributed by following any of the distribution functions. Here, we consider three distribution functions, which are very common in dealing with the random variables, namely normal distribution, poisson distribution and uniform distribution. From the distribution functions, the allocation factor is determined as follows:

$$\alpha_j = 0.33 \sum_{i=1}^3 \zeta_i \tag{2}$$

In Equation (2) ζ_1 , ζ_2 and ζ_3 are the normal, poisson and uniform distribution function values for the subjected resource's dwelling time of the past time slots. Once α is determined, it is given along with the parameters ϕ , priority of the job, resource requirement

time and predicted dwelling time of the resource to the fuzzy inference system.

2.5. Generation of Fuzzy Rules

In the previous work, only the priority status of the job, requirement time of a particular resource and the predicted dwelling time of the resource, which is available in a particular category, is used as input variables in generating fuzzy rules. In this study, as already mentioned, an allocation factor and model data, which is obtained from the prediction model, is also used in generating the fuzzy rules. However, the addend parameters are applicable for the sporadic type and semi-permanent type of resources. In order to avoid the complexity in generating fuzzy rules, only the fuzzy states of MIN and MAX are utilized instead of using MIN, MID and MAX. The fuzzy rules are as follows Fig. 3. The generated fuzzy rules are given to the Fuzzy Inference System (FIS) for self-learning.

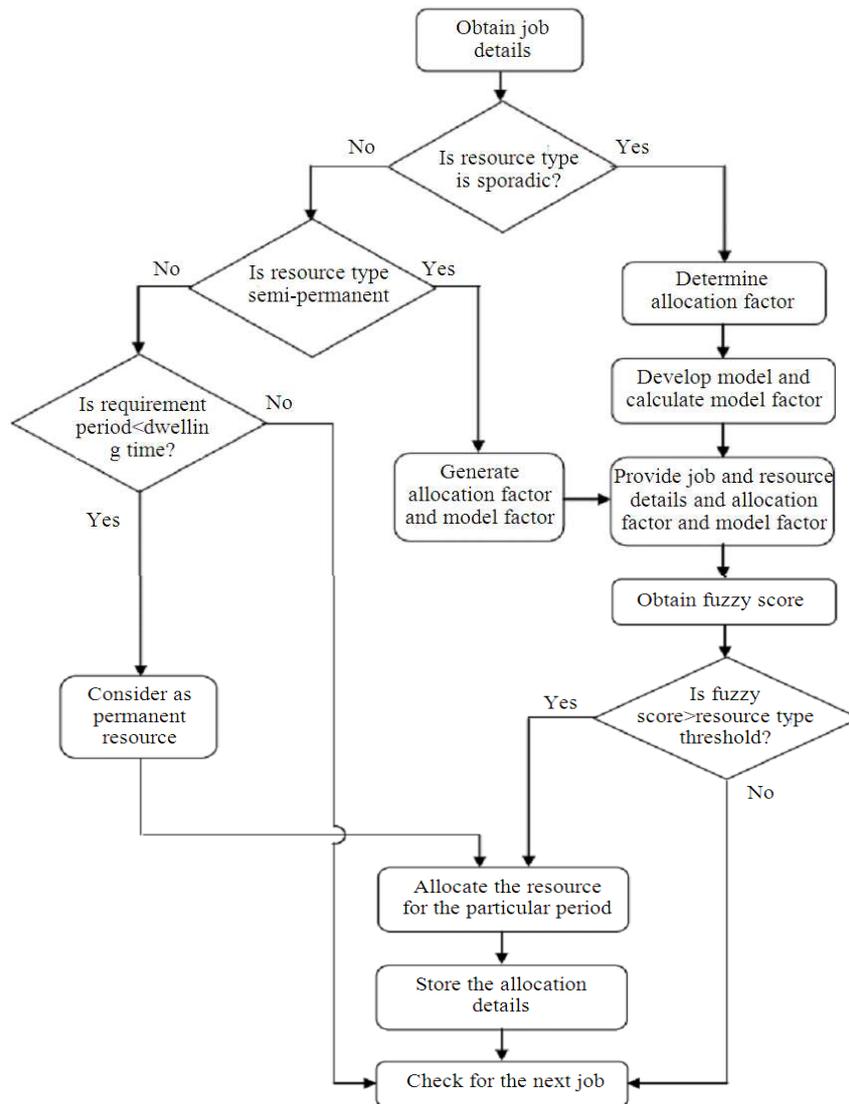


Fig. 4. Flowchart describing the entire proposed resource allocation mechanism

2.6. Allocation Mechanism

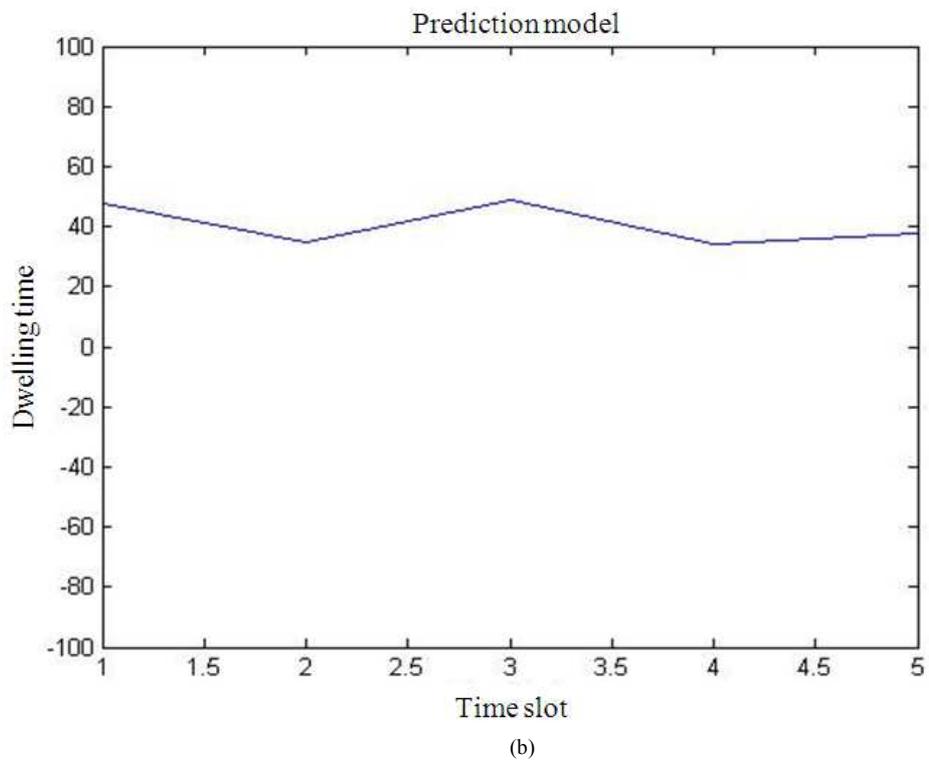
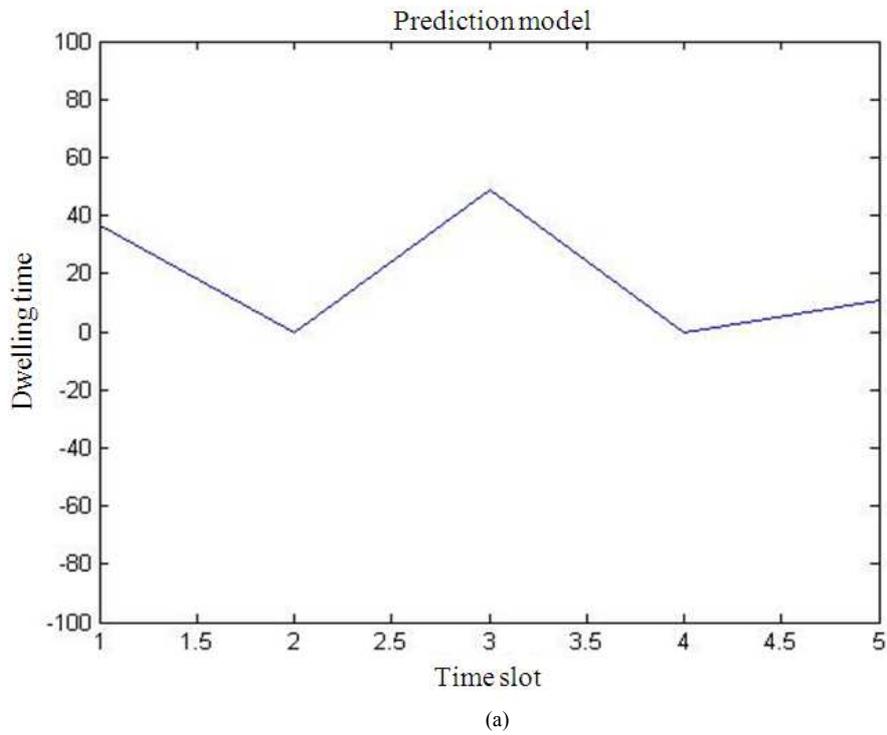
The allocation mechanism performs resource allocation on runtime by checking the job priority, required resource and the requirement time, estimate dwelling time of the resource, the availability of resource in every class, the allocation factor and the model factor ϕ . The allocation mechanism is different from the previous work only in handling the resources from sporadic type and semi-permanent type. The flowchart is given in Fig. 4, which illustrates the proposed allocation mechanism.

By following the procedures that are given in the above flow chart, the resources, which have demand for

a job, are allocated to the job based on its priority, their requirement period, dwelling time of the required resource, probability and prediction model factors. The rest of the procedures are similar to that of the procedures that are followed in the previous paper (Poonguzhali and Shanmugavel, 2011).

3. RESULTS

The proposed resource allocation technique was implemented in the working platform of MATLAB (version 7.10) with system specifications, Intel (R) core i5 CPU, 3.20GHz and 3GB RAM.



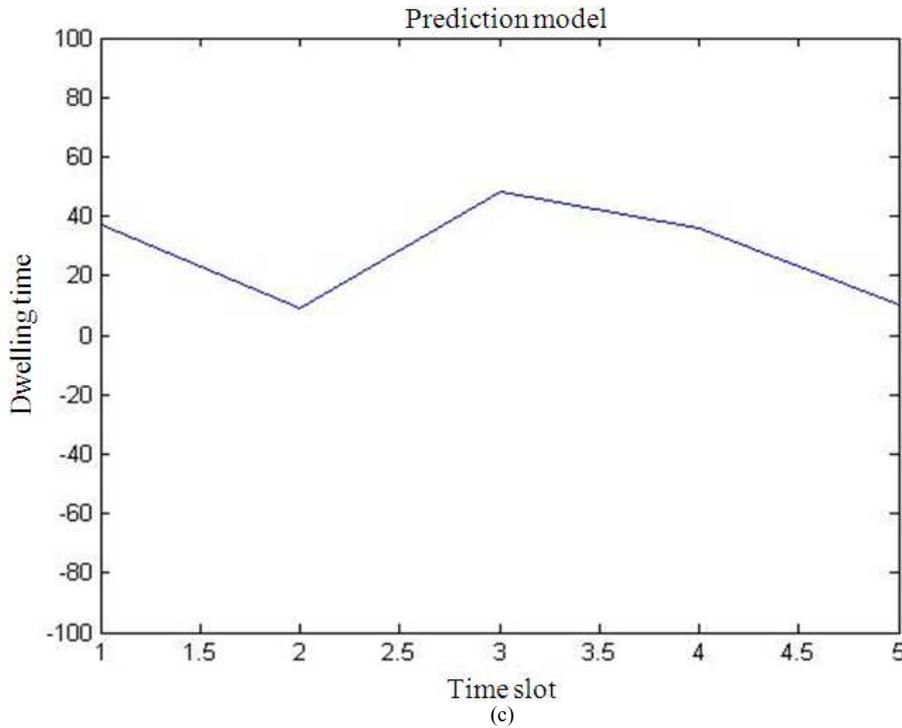


Fig. 5. Prediction model for Resource ID (a) 10, (b) 5 and (c) 7 of sporadic type

Table 1. Specifications of the historical database

Parameters	Values
No. of resources N_R	20
No. of time slots N	5
Min/Max dwelling time for permanent type resource	15/15
Min/Max dwelling time for semi-permanent type resource	10/15
Min/Max dwelling time for sporadic type resource	0/10

The performance of the technique was analyzed by executing with different synthetic job datasets, fuzzy thresholds and existing techniques. Hence, firstly we describe the dataset and its generation, secondly we analyze the results and finally, the technique is compared with the existing resource allocation techniques using the performance measures utilization rate, failure rate and makespan (Hao *et al.*, 2008; Foster *et al.*, 2006; Poonguzhali and Shanmugavel, 2011).

3.1. Dataset Description

The main requirement for the resource allocation technique is the historical dataset. Here the historical dataset is simulated with $N = 5$ time slots. The historical dataset is assumed to be classified as permanent, semi-permanent and sporadic.

Table 2. Job dataset I

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	86.67	13.33	0.46034	43.3401
0.3	0.4	86.55	13.45	0.38709	43.2811
0.3	0.5	86.45	13.55	0.36951	43.2314
0.3	0.6	85.66	14.34	0.39551	42.8359
0.3	0.7	84.97	15.03	0.40039	42.4909
0.4	0.3	84.63	15.37	0.38848	42.3210
0.4	0.4	84.42	15.58	0.40594	42.2158
0.4	0.5	84.25	15.75	0.40401	42.1308
0.4	0.6	84.19	15.81	0.43693	42.1004
0.4	0.7	83.92	16.08	0.39539	41.9659
0.5	0.3	83.74	16.26	0.44778	41.8752
0.5	0.4	83.53	16.47	0.39379	41.7710
0.5	0.5	82.81	17.19	0.42988	41.4105
0.5	0.6	82.79	17.22	0.30253	41.4028
0.5	0.7	82.53	17.48	0.40952	41.2707
0.6	0.3	81.46	18.54	0.30987	40.7376
0.6	0.4	81.31	18.69	0.41835	40.6606
0.6	0.5	81.09	18.91	0.35636	40.5516
0.6	0.6	80.62	19.38	0.37157	40.3163
0.6	0.7	80.50	19.50	0.39379	40.2560
0.7	0.3	80.44	19.56	0.40952	40.2257
0.7	0.4	80.39	19.61	0.38248	40.2011
0.7	0.5	80.11	19.89	0.46951	40.0600
0.7	0.6	80.02	19.98	0.35157	40.0167
0.7	0.7	80.00	20.00	0.43778	40.0054

Table 3. Job dataset II

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	73.33	26.67	0.46987	36.6703
0.3	0.4	73.16	26.85	0.36874	36.5867
0.3	0.5	73.02	26.98	0.48935	36.5150
0.3	0.6	73.02	26.98	0.48935	36.5150
0.3	0.7	73.01	26.99	0.36587	36.5117
0.4	0.3	72.98	27.02	0.36584	36.4967
0.4	0.4	72.68	27.32	0.34265	36.3472
0.4	0.5	72.30	27.70	0.45505	36.1554
0.4	0.6	72.00	28.00	0.43666	36.0057
0.4	0.7	71.91	28.09	0.37505	35.9616
0.5	0.3	71.83	28.17	0.39848	35.9212
0.5	0.4	71.48	28.52	0.49344	35.7450
0.5	0.5	71.23	28.77	0.35302	35.6220
0.5	0.6	71.03	28.97	0.40991	35.5210
0.5	0.7	70.96	29.04	0.43649	35.4857
0.6	0.3	70.83	29.17	0.38303	35.4214
0.6	0.4	70.79	29.21	0.35251	35.4020
0.6	0.5	70.66	29.34	0.32033	35.3377
0.6	0.6	70.51	29.49	0.47106	35.2602
0.6	0.7	70.40	29.60	0.47514	35.2052
0.7	0.3	70.40	29.60	0.40745	35.2061
0.7	0.4	70.24	29.76	0.32066	35.1277
0.7	0.5	70.13	29.87	0.41309	35.0710
0.7	0.6	70.02	29.98	0.45735	35.0154
0.7	0.7	70.00	30.00	0.35173	35.0070

Table 4. Job dataset III

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	80.00	20.00	0.30987	40.0079
0.3	0.4	79.60	20.40	0.42337	39.8058
0.3	0.5	79.15	20.85	0.30554	39.5830
0.3	0.6	78.66	21.34	0.34296	39.3371
0.3	0.7	78.35	21.65	0.43752	39.1806
0.4	0.3	78.14	21.86	0.40954	39.0760
0.4	0.4	77.88	22.12	0.34862	38.9470
0.4	0.5	77.51	22.49	0.38266	38.7614
0.4	0.6	77.29	22.71	0.48137	38.6501
0.4	0.7	76.94	23.06	0.37157	38.4766
0.5	0.3	76.85	23.15	0.48965	38.4300
0.5	0.4	76.31	23.69	0.32981	38.1624
0.5	0.5	75.88	24.12	0.34053	37.9472
0.5	0.6	75.04	24.97	0.35636	37.5269
0.5	0.7	74.78	25.22	0.41152	37.3959
0.6	0.3	74.32	25.68	0.41835	37.1659
0.6	0.4	73.98	26.02	0.43192	36.9957
0.6	0.5	73.13	26.87	0.47947	36.5701
0.6	0.6	73.03	26.97	0.34406	36.5221
0.6	0.7	72.76	27.24	0.37259	36.3866
0.7	0.3	72.42	27.58	0.34793	36.2170
0.7	0.4	72.05	27.96	0.32604	36.0325
0.7	0.5	71.13	28.87	0.48459	35.5701
0.7	0.6	71.08	28.92	0.35827	35.5468
0.7	0.7	70.00	30.00	0.43438	35.0056

Table 5. Job dataset IV

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	80.00	20.00	0.48137	40.0050
0.3	0.4	79.34	20.66	0.41835	39.6758
0.3	0.5	78.94	21.06	0.34296	39.4770
0.3	0.6	78.08	21.92	0.35827	39.0467
0.3	0.7	77.35	22.65	0.43649	38.6805
0.4	0.3	76.75	23.25	0.45505	38.3803
0.4	0.4	76.18	23.82	0.36874	38.0965
0.4	0.5	75.60	24.40	0.35173	37.8068
0.4	0.6	74.49	25.52	0.30987	37.2528
0.4	0.7	73.83	26.17	0.40039	36.9210
0.5	0.3	73.37	26.63	0.40594	36.6909
0.5	0.4	72.51	27.49	0.45772	36.2603
0.5	0.5	71.64	28.36	0.40904	35.8259
0.5	0.6	70.15	29.85	0.34262	35.0820
0.5	0.7	69.29	30.72	0.30206	34.6530
0.6	0.3	68.24	31.76	0.48130	34.1250
0.6	0.4	68.09	31.92	0.37127	34.0515
0.6	0.5	67.08	32.93	0.48065	33.5450
0.6	0.6	66.59	33.42	0.32081	33.3025
0.6	0.7	65.43	34.58	0.33033	32.7223
0.7	0.3	64.13	35.87	0.33001	32.0723
0.7	0.4	63.75	36.25	0.43151	31.8806
0.7	0.5	62.56	37.44	0.31551	31.2876
0.7	0.6	61.08	38.92	0.40031	30.5460
0.7	0.7	60.00	40.00	0.33333	30.0072

The priority and the other dwelling time ranges are set as in the previous paper (Poonguzhali and Shanmugavel, 2011). The specification for the generated historical dataset is given in the **Table 1**. A job dataset is generated, which is called as input dataset, was subjected to the mechanism for allocating the available resources. In the job dataset, a defined number of job IDs are generated. For every job ID, a defined number of resources and the requirement period are generated.

In the proposed technique, we have used prediction model and allocation factor in addition to the previous technique. The prediction model for a sample of three resources is given in **Fig. 5** Equation 3a-3c.

$$\varphi_{10} = 686 - 1215.8333t + 725.75t^2 - 173.1667t^3 + 14.25t^4 \quad (3a)$$

$$\varphi_5 = 248 - 372.8333t + 221.1667t^2 - 52.6667t^3 + 4.3333t^4 \quad (3b)$$

$$\varphi_7 = 405 - 667.75t + 377.5417t^2 - 84.25t^3 + 6.4583t^4 \quad (3c)$$

Table 6. Job dataset V

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	86.67	13.33	0.43053	43.3403
0.3	0.4	86.01	13.99	0.40552	43.0106
0.3	0.5	85.73	14.27	0.35151	42.8715
0.3	0.6	84.93	15.07	0.35551	42.4714
0.3	0.7	83.13	16.87	0.42533	41.5704
0.4	0.3	82.23	17.78	0.30534	41.1225
0.4	0.4	81.09	18.91	0.31505	40.5522
0.4	0.5	80.55	19.46	0.32033	40.2821
0.4	0.6	79.19	20.81	0.41305	39.6005
0.4	0.7	78.96	21.04	0.35113	39.4865
0.5	0.3	78.26	21.75	0.30531	39.1375
0.5	0.4	76.85	23.16	0.41152	38.4305
0.5	0.5	76.26	23.75	0.31151	38.1373
0.5	0.6	75.96	24.05	0.31255	37.9873
0.5	0.7	75.59	24.41	0.43433	37.8003
0.6	0.3	74.97	25.03	0.41335	37.4905
0.6	0.4	74.26	25.75	0.35113	37.1365
0.6	0.5	74.28	25.72	0.34202	37.1467
0.6	0.6	73.81	26.20	0.40035	36.9107
0.6	0.7	73.03	26.98	0.31551	36.5222
0.7	0.3	72.55	27.46	0.43005	36.2803
0.7	0.4	72.10	27.90	0.37652	36.0561
0.7	0.5	71.52	28.48	0.43479	35.7653
0.7	0.6	71.89	28.11	0.38219	35.9510
0.7	0.7	70.00	30.00	0.45674	35.0050

Table 7. Job dataset VI

S_{th-II}	S_{th-III}	Utilization (%)	Failure rate (%)	Make span (sec)	Consolidated score
0.3	0.3	81.33	18.67	0.430396	40.6700
0.3	0.4	80.93	19.07	0.400614	40.4704
0.3	0.5	80.66	19.34	0.371774	40.3358
0.3	0.6	80.07	19.93	0.388320	40.0405
0.3	0.7	79.36	20.64	0.413120	39.6852
0.4	0.3	78.94	21.06	0.384850	39.4756
0.4	0.4	78.45	21.55	0.356200	39.2310
0.4	0.5	78.04	21.96	0.382756	39.0256
0.4	0.6	77.43	22.57	0.415576	38.7202
0.4	0.7	77.11	22.89	0.378706	38.5607
0.5	0.3	76.81	23.19	0.409432	38.4103
0.5	0.4	76.13	23.87	0.417256	38.0702
0.5	0.5	75.56	24.44	0.368796	37.7858
0.5	0.6	74.99	25.01	0.344794	37.5012
0.5	0.7	74.63	25.37	0.398784	37.3204
0.6	0.3	73.96	26.04	0.401180	36.9854
0.6	0.4	73.69	26.32	0.385036	36.8506
0.6	0.5	73.25	26.75	0.395766	36.6304
0.6	0.6	72.91	27.09	0.381570	36.4606
0.6	0.7	72.42	27.58	0.377472	36.2157
0.7	0.3	71.99	28.01	0.384992	36.0006
0.7	0.4	71.71	28.30	0.367442	35.8609
0.7	0.5	71.09	28.91	0.423498	35.5501
0.7	0.6	70.82	29.18	0.389938	35.4155
0.7	0.7	70.00	30.00	0.402792	35.0053

In the proposed technique, the fuzzy thresholding has been performed in two locations. One is at the point of evaluating the fuzzy score that deals with the sporadic resource type and the other is at the point of evaluating the fuzzy score that deals with semi-permanent resource type. Here, the threshold values (S_{th-II} and S_{th-III}) are varied from 0.3-0.8 and the corresponding performance metrics are observed. A quantitative analysis is made with the fuzzy threshold, which is used in evaluating the fuzzy score and the corresponding utilization, failure rate and makespan. The analytical results are tabulated in **Table 2-7**.

The makespan values shows uncertain variation and lessening the threshold increases the utilization rate and hence minimizes the failure rate. So, here we can determine the best threshold values by either considering the maximum utilization rate or by considering the makespan values. To make convenient selection, here we introduce a consolidated measure, which is given in the final column of the between the utilization rate and makespan as follows Equation 4:

$$\text{Consolidated measure} = 0.5 \left(U + \frac{1}{\left(\frac{100}{MS_{\max}} \right) MS} \right) \tag{4}$$

where, U, MS and MS_{\max} utilization, makespan and maximum makespan values of the dataset respectively. The selection of fuzzy thresholds can also be considered by the above mentioned formulation.

3.2. Comparative Analysis

To substantiate the proposed technique, it is compared with the three successful conventional resource allocation techniques such as GA-based resource allocation, SA-based resource allocation and PSO-based resource allocation and also with our previous resource allocation technique.

4. DISCUSSION

Based on the specifications that are given in **Table 1**, the dataset is generated. For instance, the semi-permanent type resources are generated in between the range (Jiang *et al.*, 2008) where their dwelling times are in between the range (Chapman *et al.*, 2007) arbitrarily. Likely, the dataset has been generated for all the time slots and also for the other resource types. Another dataset has also been generated in the similar fashion for the current time slot.

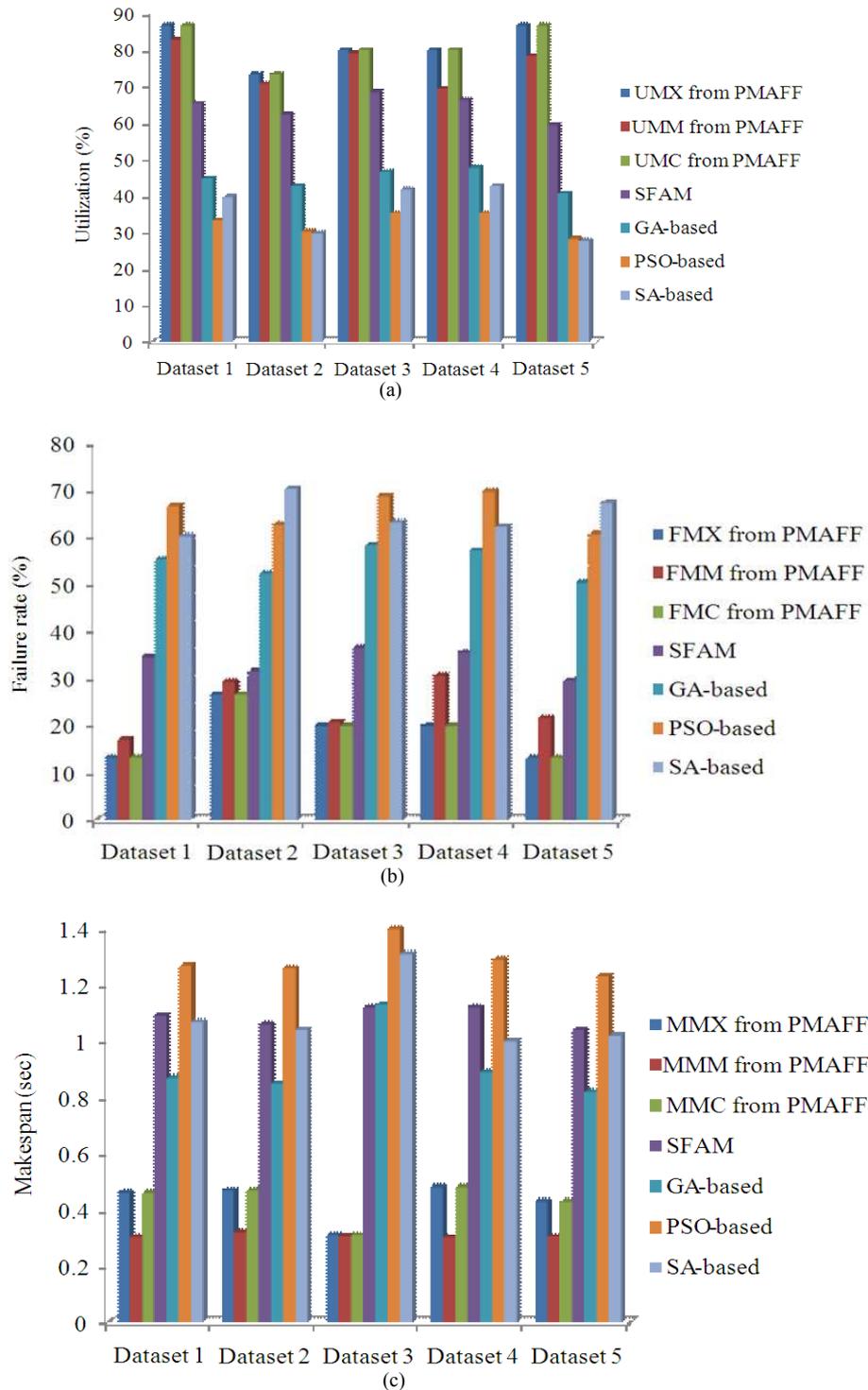


Fig. 6. Comparison graph between PMAFF, GA-based, SA-based, PSO-based and Simple Fuzzy-based Grid Resource Allocation Mechanism (SFAM) for different job datasets in terms of (a) utilization, (b) failure rate and (c) makespan

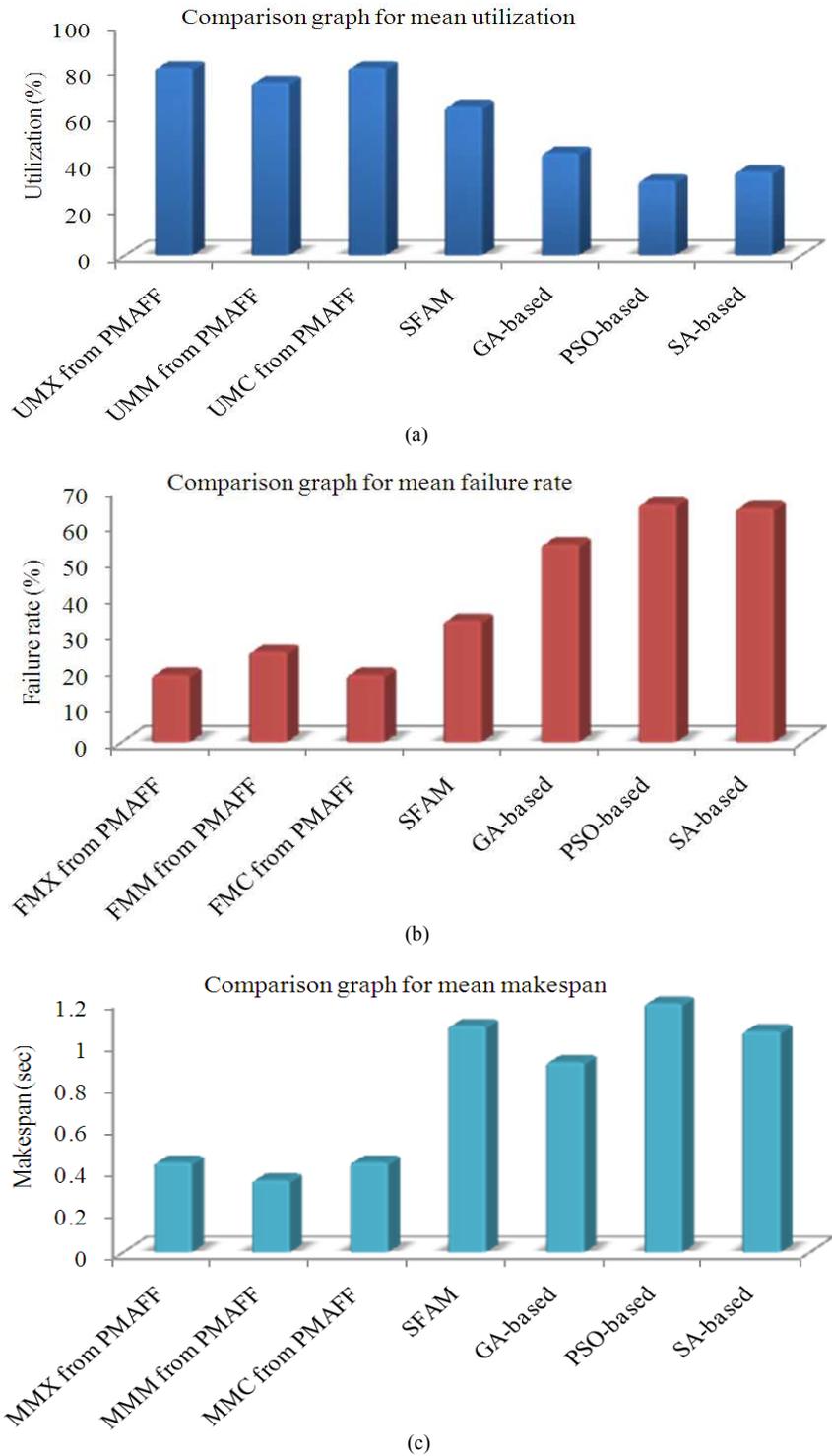


Fig. 7. Comparison graph between PMAFF, GA-based, SA-based, PSO-based, SFAM using mean (a) utilization, (b) failure rate and (c) makespan that are taken for the five job datasets

We have selected three utilization rates, failure rates and makespan values that are obtained from the proposed methodology and they are compared against the above said resource allocation methodologies. The first utilization rate, can be called as Utilization in terms of Minimum Makespan (UMM), is the obtained utilization value in which we have achieved minimum makespan. Second Utilization rate is the Maximum utilization rate (UMX) and the third utilization rate (UMC) is the obtained utilization in which we have achieved a maximum consolidated measure value. We have determined the three failure rates FMM, FMX and FMC by considering the corresponding failure rates of the selected utilization rates, UMM, UMX and UMC respectively. The first makespan value, which is notated as MMM, is nothing but the makespan value corresponding to UMM. The second makespan value, which is notated as MMX is the obtained makespan value when the utilization reaches maximum and MMC is the consolidated makespan value. These UMM, UMX, UMC, FMM, FMX, FMC MMM, MMX and MMC are marked as bold in the **Table 2** In all the datasets, it can be observed that, the UMX and UMC, FMX and FMC and MMX and MMC are same, respectively. This means that utilization makes a great impact over the consolidated score and not the makespan values. The previously proposed technique i.e., SFAM fails to achieve minimum makespan when compared to GA-based allocation mechanism. This can be acceptable as utilization has higher impact rather than makespan; however, proposed method has achieved both in utilization and makespan values. This can be observed from **Fig. 6**, which illustrates the performance of proposed method in all the submitted job datasets and from **Fig. 7**, which illustrates the mean of the performance that are observed for the individual datasets.

5. CONCLUSION

The study proposed a PMAFF grid resource allocation mechanism by the incorporation of two newly introduced parameters, prediction model and allocation factor in the fuzzy resource allocation mechanism. The mechanism introduced a new classification scheme based on dwelling time of the grid resources in this past time slots. Based on the values, it developed a prediction model to determine the dwelling time of the upcoming time slot. Moreover, it considered three standard distribution functions to determine an allocation factor. Using

these details, a simple fuzzy-based resource allocation mechanism was devised to allocate the resources to the submitted job datasets. As the fuzzy threshold selection was found to be critical in achieving expected utilization rate and makespan values, an analysis was made to make the threshold selection more convenient. With different selection parameters, the obtained utilization rate, failure rate and makespan values were compared against the successful heuristic search resource allocation algorithms such as GA-based resource allocation technique, SA-based resource allocation technique and PSO-based resource allocation technique. From the results, it was observed that the PMAFF resource allocation mechanism achieved good performance measures rather than the aforesaid techniques. It also proved that it has improved the previously proposed simple-fuzzy based resource allocation technique.

6. REFERENCES

- Abba, H.A., N.B. Zakaria and N. Haron, 2012. Grid resource allocation: A review. *Res. J. Inform. Technol.*, 4: 38-55.
- Boukerram, A. and S.A.K. Azzou, 2006. Implementation of load balancing algorithm in a grid computing. *Am. J. Applied Sci.*, 3: 1810-1813. DOI: 10.3844/ajassp.2006.1810.1813
- Chapman, C., M. Musolesi, W. Emmerich and C. Mascolo, 2007. Predictive resource scheduling in computational grids. *Proceedings of the IEEE International Conference of Parallel and Distributed Processing Symposium*, Mar. 26-30, IEEE Xplore Press, Long Beach, CA., pp: 1-10. DOI: 10.1109/IPDPS.2007.370306
- Chen, J. and B. Lu, 2008. Load balancing oriented economic grid resource scheduling. *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Dec. 19-20, IEEE Xplore Press, Wuhan, pp: 813-817. DOI: 10.1109/PACIIA.2008.324
- Christodoulopoulos, K., V. Sourlas, I. Mpakolas and E. Varvarigos, 2009. A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks. *Comput. Commun.*, 32: 1172-1184. DOI: 10.1016/j.comcom.2009.03.004
- Darby, P.J. and N.F. Tzeng, 2010. Decentralized QoS-aware checkpointing arrangement in mobile grid computing. *IEEE Trans. Mob. Comput.*, 9: 1173-1186. DOI: 10.1109/TMC.2010.80

- Farooq, U., S. Majumdar and E.W. Parsons, 2009. Achieving efficiency, quality of service and robustness in multi-organizational Grids. *J. Syst. Software*, 82: 23-38. DOI:10.1016/j.jss.2008.03.064
- Foster, I., C. Kesselman and J.M. Nick, 2006. The physiology of the grid: An open grid service architecture for distributed systems integration. *Phys. Grid*, 1: 267-272.
- Hao, Y., Y. Xu, G. Liu and Z. Pan, 2008. An expectation trust benefit driven algorithm for resource scheduling in grid computing. *Proceedings of the 3rd International Conference on Innovative Computing Information and Control*, Jun. 18-20, Dalian, Liaoning, pp: 88-88. DOI: 10.1109/ICICIC.2008.139
- Jiang, J. H. Ji, G. Xu and X. Wei, 2008. Scheduling algorithm with potential behaviors. *J. Comput.*, 3: 15-59. DOI: 10.4304/jcp.3.12.51-59
- Kamalam, G.K. and V.M. Bhaskaran, 2012. Novel adaptive job scheduling algorithm on heterogeneous grid resources. *Am. J. Applied Sci.*, 9: 1294-1299. DOI: 10.3844/ajassp.2012.1294.1299
- Keerthika, P. and N. Kasthuri, 2012. An efficient fault tolerant scheduling approach for computational grid. *Am. J. Applied Sci.*, 9: 2046-2051. DOI: 10.3844/ajassp.2012.2046.2051
- Latip, R., H. Ibrahim and F.A. Al-Hanandeh, 2011. Scientific data sharing using clustered-based data sharing in grid environment. *Am. J. Econ. Bus. Admin.*, 3: 146-149. DOI: 10.3844/ajebsp.2011.146.149
- Marowka, A., 2000. The GRID: Blueprint for a new computing infrastructure. *Sci. Int. J. Parallel Distrib. Comput.*, 3: 248-252.
- Naisipour, M., M.H. Afshar, B. Hassani and A.R. Firoozjaee, 2008. Collocation discrete least square (CDLS) method for elasticity problems and grid irregularity effect assessment. *Am. J. Applied Sci.*, 5: 1595-1601. DOI: 10.3844/ajassp.2008.1595.1601
- Odeh, S., R. Faqeh, L.A. Eid and N. Shamasneh, 2009. Vision-based obstacle avoidance of mobile robot using quantized spatial model. *Am. J. Eng. Applied Sci.*, 2: 611-619. DOI: 10.3844/ajeassp.2009.611.619
- Poonguzhali, M. and S. Shanmugavel, 2011. Fuzzy-based simple and proficient resource allocation technique for dynamic grid resources. *Int. J. Next-Generation Comput.*
- Puri, S. and H. Dev, 2012. Deliberate secure grid computing blueprint design in Indian context. *Int. J. Comput. Network Inform. Sec.*, 4: 33-42.
- Rafee, R. and H. Rahimzadeh, 2009. A new method for constructing the coefficients of pressure correction equation for collocated unstructured grids. *Am. J. Applied Sci.*, 6: 93-100. DOI: 10.3844/ajassp.2009.93.100
- Ramesh, D. and A. Krishnan, 2012. An optimal load sharing technique for grid computing. *Am. J. Applied Sci.*, 9: 1101-1106. DOI: 10.3844/ajassp.2012.1101.1106
- Richard, R.J.A., A.A. Joshi and C. Eswaran, 2008. Implementation of computational grid services in enterprise grid environments. *Am. J. Applied Sci.*, 5: 1442-1447. DOI: 10.3844/ajassp.2008.1442.1447
- Senthilnathan, T. and T. Purusothaman, 2012. A multi-agent based data replication mechanism for mobile grid. *Am. J. Applied Sci.*, 9: 542-552. DOI: 10.3844/ajassp.2012.542.552
- Suwan, I., A. Brandt and V. Ilyin, 2012a. Secure selection of multiple resources based on virtual private network for computational grids. *J. Math. Statist.*, 8: 361-372. DOI: 10.3844/jcssp.2011.1881.1887
- Suwan, I., A. Brandt and V. Ilyin, 2012b. Multilevel evaluation of coulomb lattice sums of charge systems. *J. Math. Statist.*, 8: 361-372. DOI: 10.3844/jmssp.2012.361.372
- Vijaya, S., A.N. Ma and Y.M. Choo, 2009. A gate to gate assessment of environmental performance for production of crude palm kernel oil using life cycle assessment approach. *Am. J. Environ. Sci.*, 5: 267-272. DOI: 10.3844/ajessp.2009.267.272
- Wankar, R., 2008. Grid computing with globus: An overview and research challenges. *Int. J. Comput. Sci. Appl.*, 5: 56-69.
- Yien, J.M., K.H. Chen, K.P. Huang and C.J. Huang, 2011. Managerial decision and resource reallocation: A dynamic capability perspective. *Am. J. Space Sci.*, 7: 632-634. DOI: 10.3844/ajjssp.2011.632.634