ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

Multi-agent model for scheduling of activities by applying shortest processing time (SPT) rule

R.Y.Youssef[1], M.I.Abdalla[2], M.H.Ghanim[3]

Computer Science, College of Science, ZagazigUniversity, Egypt[1]

Professor Dr, Electronics and Communication Dept, College of Engineering, Zagazig University, Egypt[2]

Professor Dr,Mathematics Dept., College of Science, Zagazig University Egypt.[3]

Abstract*:* Online scheduling is usually referred to as a closed loop that includes a feedback after scheduling execution. The whole system depends on data collection fed to the receivers of the reactive layer from sensors and various input devices. The practical success of the scheduling system in reacting to practical applications depends entirely on having the data ready for execution for the algorithm on time. This Paper introduce a methodology for enhancing the rate of data collection from the input devices, which acts as a front end to the reactive scheduling system and will have a direct impact on scheduling horizon, frequency, time limit selection and reasoning reuse.

We've got the solution for driving the multi-purpose machines and the way of control and enhance scheduling performance within the context of a dynamic production scheduling problem by using our Multi-Agent (MA) program model for the reactive scheduling methods. We've got the solution for driving the multi-purpose machines and get the way of controlling and enhancing the scheduling performance within the context of a dynamic production scheduling problem by using our Multi-Agent (MA) program model for the reactive scheduling methods, Multi-Agent (MA) using the Short Processing Time (SPT) that the rule of shorting time of the jobs in its queue. In this Paper, we compare between two methods (Multi-agent vs. fuzzy) to minimize the total processing time and improve the performance of the production lines. The results indicate that the Multi-agent approach has a better performance compared to the Fuzzy methodology.

Keywords: scheduling;Multi-Agent;Short Processing Time (SPT);fuzzy.

1. INTRODUCTION

This Paper will compare the results when using two methods (Multi-agent vs. Fuzzy) and it will explain the comparison of the Agent scheduling processing and Fuzzy scheduling processing to minimize the total processing time by using SPT (Short Processing Time), which is dispatching rule for performance measure, and improve the performance of the production lines. [1]

Scheduling problems are found in a lot of application domains. Well known is the scheduling of production where manufacturing operations have to be assigned to resources, like machines facilitate when are scarcely, scheduling also finds applications in very important areas, some of them include the scheduling of airline crews, space missions, projects in different domains and clinical surgery, event timetabling and processor scheduling.

In general, scheduling deals with the temporal assignment of activities to limited resources where a set of constraints has to be considered. Due to the exponential size of scheduling problems, it is quite difficult to create good or optimal schedules shown by optimized goal functions or other evaluation criteria.

This is not only the generation of a schedule where a hard problem found or maybe harder, and the normal case in everyday work is the adaptation of an existing schedule to the changing scheduling environment. The changes include events like resource breakdowns as well as limitations or prescriptions interactively given by the user of the scheduling system.

Scheduling systems have been built to support the users in performing their scheduling tasks. These systems are incorporated with the scheduling knowledge as well as presentations and database components.

1.1 SCHEDULING SYSTEMS

The agent-based systems are attractive technology for solving scheduling problems when comparing with solving single complex mathematical programs for obtaining the optimum solutions for a number of reasons: [1, 5]

1. The uncertainties in the availability of resources and the dynamics of job processing times and job arrival rate (and the environment as a whole) are overwhelmingly large for solving a single optimization problem repeatedly, that is every iteration requires waiting for a significant period of time in order to obtain an optimal solution.
2. The optimization problem is a multi-objective problem in which the objective function itself (for example, the weights on multiple objectives) changes frequently.
3. The data required to solve the optimization problem are more accurate if the optimization is solved at the source where data are gathered.
4. The application in which obtaining the optimality of the solution is not considered a significant result, but rather a sub-optimal solution based on accurate data representation (so that the schedule will be more accurate when

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

executed). That result produced in shorter period is considered significant.

5. The system design issue requires distributed information processing and integration of planning and execution, where the focused issues such as information persistence, self-configurability of the distributed entities, and scalability and inter-operability of the software are critical.

Based on the above reasoning, it is inferred here that the agent-based systems are appropriate alternative means of scheduling and resource allocation methodology for the following scheduling application:

Dynamic factory scheduling: The examples of dynamic factories are the ones where job-processing time is uncertain (Example; chemical factories and part testing facilities), there is a need for frequent change in machine set-up (Example; a customized factory), thebreakdown/availability of resource is stochastic in nature and frequent, and the job arrival rate is high and uncertain (Example; a customized factory connected to a web-enabled ordering site).

1.2 Scheduling with agent

The most basic philosophical definition of an agent is that it is a persistent software entity that receives and sends signals/events while acting autonomously on the behalf of users. This is popularly known as the sensor-effecter definition of an agent. By this definition, For example, the piece of Java code shown in Figure 1 can be called an agent.

```
Class agent
{
public Object anObject = new Object();
void main()
{
while(true)
{
synchronized(anObject)
{
try
{
anObject.wait();
}
catch(InterruptedExceptionie)
{
};
// Act autonomously
// Send an event (notify another object)
}
}
}
}
```

Figure 1: A Simple Agent Code

In a large agent-based system where the individual agent code may be simple, the number of agents in the system may be so large that the complexity of interactions between agents becomes overwhelming; some even consider an agent to be composed of several objects shown in Figure 1 that share data and threads, which adds further level of complexity. Although theoretically these are correct interpretations of the above definition of an agent translated into a programming source code, the task of programming such agent implementations becomes cumbersome for non-trivial tasks order simplify the programming of agents, we redefine below an agent solely from the programming standpoint:

An agent is a group of event-driven activities that share data, thread, and execution concurrency structure, from the philosophical standpoint, an activity of an agent can be considered as an active role instant of the agent that describes only one single behavior of the agent being programmed. From a programming standpoint, an activity can be defined as follows:

An activity is an active object that operates on internal data. The internal data and methods of the active object are encapsulated, meaning that they are not directly accessible to other objects, for example, a robotic agent with two arms can be described by two mutually-dependent activities in control of left arm and right arm motion respectively. Similarly, an agent representing a factory shop floor manager can have buyer and seller activities, each negotiating tasks to subcontract/allocate and accept for maximum payoff. These definitions of the agent and activities are consistent with the philosophical definition of agents. For example, an agent is persistent if it has at least one active object all the time. Furthermore, the agent can be made autonomous if its activities generate timer events directed to them. [5]

The activities can be provided with support methods to generate events and receive events. The activities can also be designed to send events or respond to events that represent a user's behavior and hence act on the behalf of the user. The concept behind the programming of agents is analogous to that of Object Oriented Programming (OOP). An object encapsulates data and contains methods to manipulate the data. We view an agent as encapsulating activities or active objects, and have a concurrency structure to manipulate their execution. This particular concept of programming agents is called Activity Centric Programming (ACP). We present a brief overview of ACP and the need of ACP style of programming for agent programming. [2] The programming of multi-agent modelling as shown in Appendix A. In this Paper an overview of scheduling problems, techniques to solve scheduling problems, scheduling systems as well as research issues in scheduling is given.Scheduling, in general, means the temporal assignment of activities to resources where several constraints have to be fulfilled. A wide variety of application areas are faced with scheduling problems. Some of these areas are the manufacturing of products, crew schedules, project scheduling in different areas and ship building. In most of these areas, the temporal goals are most important; this meant that meeting the due dates is the main goal of scheduling, because this will satisfy the customers by enabling the companies to reach their demands. But also, the other goal is minimizing the total of the processing time because it is very important and often a mix of different goals could be founded. These entire reasons make the finding of the good schedule is very difficult task. Several aspects of scheduling are notable. Creating a schedule in advance for a period of time is called predictive scheduling, when repairing the schedules due to the actual events is called reactive scheduling. [3] Some of these events are the scheduling of machine breakdowns, maintenance intervals or the events from the

DOI 10.17148/IJARCCE.2015.4584

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

logistic level such as new or cancelling orders. Reactive scheduling means adapting the schedule to the new situation using appropriate actions to handle all of the events. When looking at the scheduling process in an organization, we recognize that the scheduling is incorporated in the decision structures of the companies, we always find humans who have to decide, interact or control within the process. Thus scheduling also has a very important interactive dimension.

Interactive scheduling combines both predictive and reactive scheduling with the requirements of a user who wants to keep the decisions in his hands like introducing new orders, cancel orders, change priorities and set operations to specific schedule positions. These decisions have to be regarded within the scheduling process.

2. SCHEDULING WITH FUZZY-APPROACHES

Fuzzy scheduling provides the possibility to deal with the inherent dynamic and incompleteness of the scheduling area. [6] It allows the representation (by Fuzzy sets and linguistic variables) and the inference (by Fuzzy rules) from vaguely formulated knowledge. The main types of imprecise scheduling information addressed by Fuzzy sets are:

- Vaguely defined dates or durations such as due dates.
- Vague definitions of preferences such as preferences between alternatives.
- Uncertainty about the value of scheduling parameters such as the process time.
- Aggregated knowledge such as machine groups instead of individual machine.

The components and contracture of Fuzzy controller are shown in figure 2:

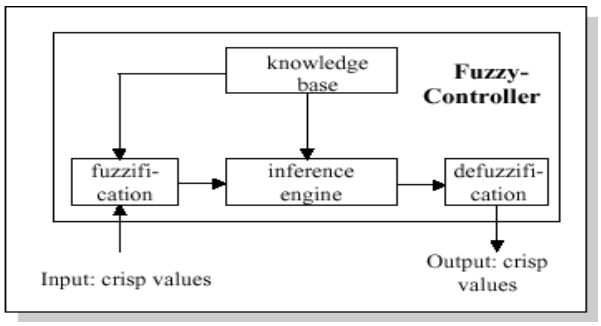

Figure 2: Fuzzy controller

In order to handle the imprecise information with Fuzzycontrollers, the following steps have to be performed:

- Transformation of scheduling data into a knowledge representation that can be handled by a Fuzzy controller (fuzzification). Imprecise knowledge is represented by linguistic variables denoting the possible values like {very low, low, normal, high, very high}.
- For each of the possible values a membership function (Example; triangular) is given, which is used in combining and processing the Fuzzy sets.
- Processing of the Fuzzy scheduling knowledge towards a decision by means of given rules and integration of Fuzzy arithmetic to deal with imprecise or vague data. Fuzzy sets and rules are stored in the knowledge base of the Fuzzy controller.[7]

An example of a Fuzzy rule is shown in Table 1

Table 1: Fuzzy rule

```
/* rule to determine the
   importance of orders */

IF      Time_demand(very low)
    FUZZY_AND Priority(normal)
      FUZZY_AND Date(soon)
THEN   Importance(normal);
```

- Transformation of the Fuzzy scheduling decision into crisp scheduling data (defuzzification) (Example; to determine concrete dates for operations).

The principal advantage of Fuzzy scheduling is the possibility to focus on the significant scheduling decisions.

3. FUZZY GLOBAL SCHEDULING

In order to solve the global scheduling problem with a Fuzzy-based system (controller) the following sub-problems need to be addressed:

- Modeling and transformation of scheduling data into a knowledge representation that can be handled by a Fuzzy controller to carry into fuzzification.
- Processing of the Fuzzy scheduling knowledge for making a decision meansof given rules and integration of Fuzzy arithmetic to deal with imprecise or vague data.
- Transformation of the Fuzzy scheduling decision into crisp scheduling data to carry into defuzzification.

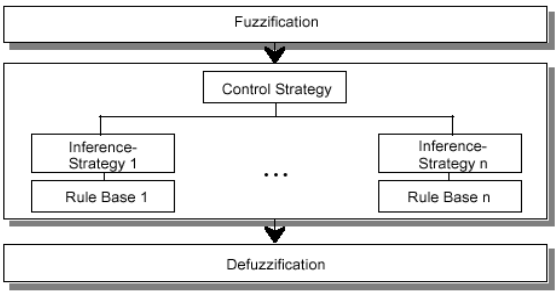

Figure.3: Structure of the Fuzzy controller

Figure 3 visualizes the schematic architecture of the Fuzzy controller we use to solve the global predictive scheduling tasks. The data for the controller is made available in an appropriate format by Fuzzification.

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

The rule bases together with the inference strategy determine the method for generating a predictive global schedule. The controller is designed so generally that it can

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

be also used for obtaining the solution of other scheduling problems, the job shop scheduling problems and local scheduling problems, just by changing the data, the rules and the control strategy. Therefore we provide functionality for an easy change or extension of the Fuzzy knowledge base. [8]

4. SIMULATION ANALYSIS

The system consists of multiple machine agents each of which is associated with a machine. Machine agents coordinate with each other to decide the machine to which the incoming job should be routed. A machine agent keeps up-to-date information of its associated machine: queue length (in the form of total processing time and total number of jobs), machine status (IDLE, BREAKDOWN, SETUP, MAINTENANCE and RUNNING), and the type of operation being processed.

It starts by presenting the graphs of all status (IDLE, BREAKDOWN, SETUP, MAINTENANCE and RUNNING) in comparison with their average response time and number of Experiments was performed Table 2 in the two techniques Rational Multi-Agent and Fuzzy inference. Fuzzy inference uses the prevailing conditions at the job shop to dynamically select the most appropriate machine [9]. After that it presents the total effect response time without RUNNING State and the total effect response time with RUNNING State. Finally it presents the graph of program totaltime of the two techniques. (Note: Experiment 1 has 2 jobs and 2 Machines) [4]

Table 2: Design of Experiments

Number of Experiments	Number of jobs	Number of Machines
5	6	6
10	11	11
15	16	16
20	21	21
25	26	26

5. GRAPHS ANALYSIS

5.1 IDLE State Graph

EE

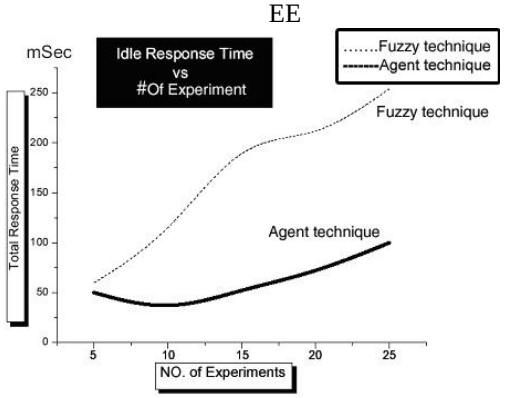

Figure 4: Idle response time

This graph in Figure 4 shows make the view clear, one of our objectives is to make the most critical state (IDLE) is the fastest one at all, that graph show the great difference between the response time of IDLE state in Multi-Agent technique and Fuzzy, that due to the higher priority that was assigned to handles the IDLE state operations. Experiment number is not a matter; it makes no difference if we increase the Experiment number, the results is known that the Multi-Agent technique will give lower response time. It is noticeable that: the difference of response time between the two techniques on the graph is relatively large. The total difference is 526.06667, since

Total difference = Summation (Fuzzy response time – Multi-Agent response time) = 526.06667. (as shown in Table 3)

Table 3: Idle State.

Response Diff.	Experiment number
10	5
76.2	10
153.36667	15
132.5	20
154	25

Table 3 shows that the more the Experiment number increases the more the difference between the Fuzzy and Multi-Agent techniques increase.

5.2 BREAKDWON State Graph

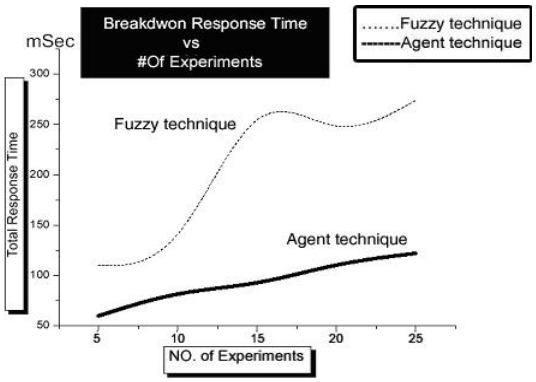

Figure 5: Breakdown response time

This graph in Figure 5 indicates the BREAKDWON state that has the second greatest priority still also has a smaller response time in Multi-Agent technique rather that Fuzzy technique that due to the Second high priority that was assigned to handles the BREAKDWON state operations. We note from the Table 4 that the difference increasing is smaller than the IDLE state. But both are relatively near since they have the two greatest priorities.

Table 4: Breakdown State.

Response Diff.	Experiment number
50	5
65	10
206.6666667	15
117.5	20
152	25

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

Other two states are not considered, since the most important states are the highest and lowest priorities.

5.3 SETUP State Graph

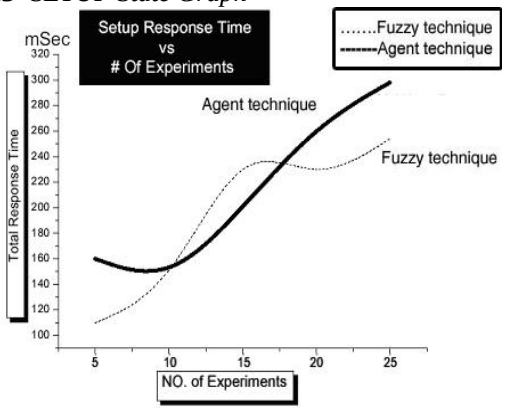

Figure 6: Setup response time

This graph in Figure 6 indicates the SETUP state. We note that the response time in Multi-Agent technique rather that Fuzzy technique is higher in the beginning of the experiment and then begins to decline gradually when repeating experiments to score less time and then come back to rise again. (as shown in Table 5)

Table 5: Setup State

Response Diff.	Experiment number
-50	5
10	10
50.6666667	15
-20	20
-49	25

5.4 Maintenance State Graph

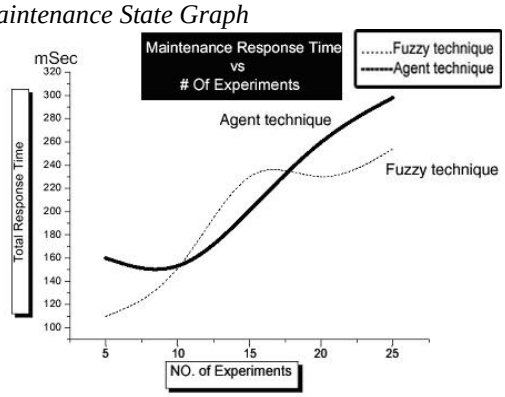

Figure.7: Maintenance response time

This graph in Figure 7 indicates the MAINTENANCE state. We note that the response time in Multi-Agent technique rather that Fuzzy technique is higher in the beginning of the experiment and then begins to decline gradually when repeating experiments to score less time and then come back to rise again(as shown in Table 6)

Table.6: Maintenance State

Response Diff.	Experiment number
-50	5
10	10
50.6666667	15
-20	20
-49	25

5.5 RUNNING State Graph

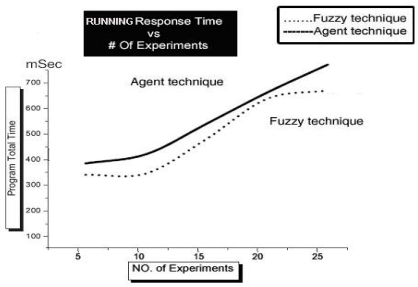

Figure.8: Running response time

The graph in Figure 8 shows that the RUNNING state in both techniques Fuzzy and Multi-Agent, we can get the meaning of the graph directly. The response time of the RUNNING state in Multi-Agent is greater than Fuzzy from first point to the last one. That due to the low priority that was assigned to the RUNNING state, since it must be stored in Database but in how long it's not a matter. Experiment number is not a matter; it makes no difference if we increase the experiment number, the result is known that the Multi-Agent technique will give greater response time. It is noticeable that the difference of response time between the two techniques on the graph is relatively large. (as shown in Table 7).

Table.7: Running State.

Response Diff.	Experiment number
-50	5
-85	10
-66.6666667	15
-27.5	20
-114	25

5.6 Total Effect Response Time

5.6.1 without RUNNING State

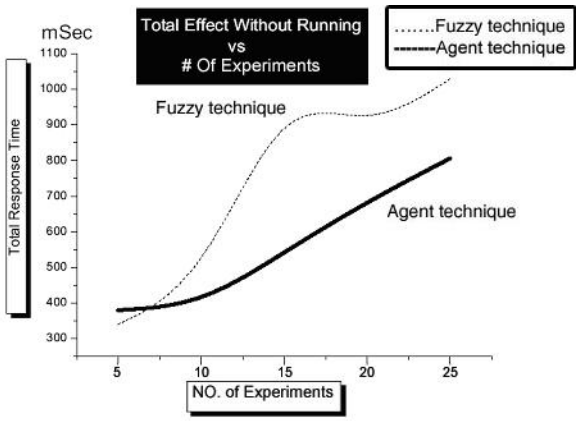

Figure 9: Total Effect Time without Running

The graph in Figure 9 shows the total effect response times of all states except the RUNNING state it's clear that the Multi-Agent technique is greater than the Fuzzy technique, since the all the states except the RUNNING state have higher priorities than the priorities assigned to the Fuzzy technique. The difference increases as the number of experiments number increases.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

5.6.2 with RUNNING State

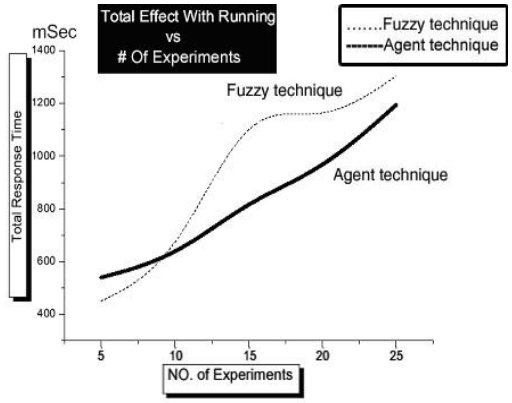

Figure.10: Total Effect Time with Running

Although the graph in Figure 10 shows we add the response time of RUNNING state the Multi-Agent technique still smaller than Fuzzy technique for reasons we described above, but the only effect is the difference becomes smaller, that because the RUNNING state response time in Multi-Agent is greater than in the Fuzzy technique

5.7 Program Total Time Graph

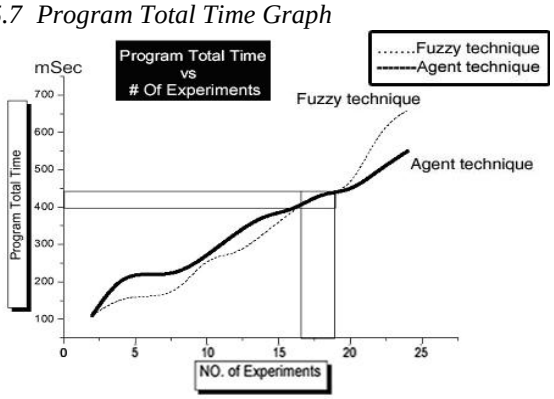

Figure.11: Program Total Times

- Function of Multi-Agent technique:

Y = A + B1*X + B2*X^2 + B3*X^3

Parameter	Value	Error
A	-0.68702	2.12613
B1	0.01805	0.02398
B2	-1.65406E-5	8.14843E-5
B3	2.2969E-8	8.43301E-8

- Function of Multi-Agent technique:

Y = A + B1*X + B2*X^2 + B3*X^3

Parameter	Value	Error
A	-1.63913	1.79707
B1	0.03474	0.0196
B2	-5.66166E-5	5.77329E-5
B3	4.02454E-8	4.99138E-8

The graph represented in Figure 11 shows the program total time (which is the difference between the time of first read of data from files and the time of last write the response to the file). Starting from the 5th experiment we can notice that the total time of Multi-Agent program technique total time is greater than itself in Fuzzy technique, that due to the scheduling (that means to switch from one Agent to another) that consider an additional time over each Agent operations.

Agent scheduling time still makes the Program total time of Multi-Agent technique is larger until a particular point in which the Program total time of Fuzzy technique is equal to the Multi-Agent technique, after that Fuzzy technique Program total time becomes greater than the Multi-Agent technique. The first reason of the occurrence is the processor waste time, when the program requests any system services such as requesting IO operation, the current program is blocked until the system provides the requested service. In Fuzzy technique, at each iteration, the program tries to read data from the file using IO service, so it moves to blocked state, then the program waits until the system provides the requested service.

Note: Increasing the number of experiments causes increasing in the programing at the blocked state and useless processing time. So it increases the program total time in the Fuzzy technique in semi-linear manner.

In the Multi-Agent technique when the one Agent is in the blocked state another Agent makes use of the processor time, which enhance the program total time.

The second reason is, in the Multi-Agent technique the Program total time is consisting of the time of executing all Agents plus Agents scheduling.

From Figure 11, It could be understand that the time of executing all Agents in Multi-Agent technique after 10 experiments is smaller than the Fuzzy technique, but after adding the Agents scheduling time makes Total time Program is greater than Fuzzy, counting until the Fuzzy Program total time is equal to the executing of all Agents plus time scheduling in Multi-Agent technique, because the difference from Figure 11 becomes equals to time scheduling then it becomes greater. At point (390msec, 17 Experiments) to point (440msec, 18 Experiments), after that the Program total time of the Multi-. Agent becomes smaller than the Fuzzy technique.

5.8Comparison between Fuzzy and Multi-Agent Results

From the results, the program total time using Multi-Agent technique by SPT Scheduling is shorter than Fuzzy technique by SPT Scheduling. By SPT Scheduling there exist two techniques Fuzzy and Multi-Agent. The Fuzzy program total time is shorter than until reaching point(390msec, 17 Experiments) to point(440msec, 18 Experiments), it becomes equal in this range,after thatthe Program total time of the Multi-Agent becomes is shorter than the Fuzzy technique.

6. CONCLUSION

Reactive schedule has become one of the most important techniques used in the industry to overcome the gap between planned scheduling and the real parameters effectives scheduling implementation, such as sudden machine failure, Barbour absence and shortage of certain raw material. One of the most important issues in the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

success of this technique is to capture data and update the model in the database at its consequence responses. Therefore a multithreading technique was used to enhance data capture instead of the conventional sequential method which increases its performance by 256.5%.

The SPT and multi-agent method were used to find optimal reactive scheduling (RS).

A comparison was preformed between multi-agent (MA) system and Fuzzy technique it was found for more than eighteen machines the performances by the MA system was much better the Fuzzy.

From the results, the program total time using Multi-Agent technique by SPT Scheduling is shorter than Fuzzy technique by SPT Scheduling.

By SPT Scheduling there are two techniques Fuzzy and Multi-Agent. The Fuzzy program total time is shorter until reaching point(390msec,17Experiments)to point (440msec, 18 Experiments), it becomes equal in this range, after that the Program total time of the Multi-Agent becomes is shorter than the Fuzzy technique.

REFERENCES

[1] Ethan Hodys ,"A Scheduling Algorithm For A Real-Time Multi-Agent System", Master Of Science In Computer Science, University Of Rhode Island, 2000.

[2] AmundTveit, Norwegian University of Science and Technology"A survey of Agent-Oriented Software Engineering", First NTNU CSGSC,May 8, 2001. Available at: http://www.elcomag.com/amund/

[3] B.Frankovic, T.T. Dang,"Agent based Scheduling in Production systems", International Journal of Production Research, Volume 40, Number 15, P: 3669 - 3679 / October 15, 2002.

[4] V. Subramaniam, T. Ramesh and A.S. Raheja "Exploiting the flexibility of multiple job routes in a Dynamic Job Shop",Republic of Singapore 2002.

[5] Haizheng Zhang, and Victor Lesser , "A Queueing Theory Based Analysis of an Agent Control Mechanism in peer-to-peer Information Retrieval Systems", ACM 1595931643,November 4, 2005, Bremen, Germany.

[6] Surekha P, S.Sumathi, Solving Fuzzy based Job Shop Scheduling Problems using Ga and Aco , Journal of Emerging Trends in Computing and Information Sciences , VOL. 1, NO. 2, Oct 2010.

[7] emion Lei, Scheduling fuzzy job shop with preventive maintenance through swarm-based neighborhood search, Technology, June 2011, Volume 54, Issue 9-12, pp 1121-1128, Date: 31 Oct 2010.

[8] Vikas S. Jadhav1 and V. H. Bajaj2{1Research Student, 2Professor}Department of Statistics, Dr.B.A.M.University, Aurangabad-431004(M.S) INDIA. International Journal of Statistical and athematika,Selection of Job Shop Scheduling Problem Using Fuzzy Linguistic Variables , ISSN: 2277- 2790 E-ISSN: 2249-8605, Volume 3, Issue 1, 2012 pp 01-05 .

[9] Xiaojuan Wang, Wenfeng Li, Ying Zhang,An improved multi-objective genetic algorithm for fuzzy flexible job-shop scheduling problem, International Journal of Computer Applications in Technology archive, Volume 47 Issue 2/3, June 2013, Pages 280-288.