APPLICATION OF A FUZZY-SIMULATION MODEL OF SCHEDULING ROBOTIC FLEXIBLE ASSEMBLY CELLS

[1,2]Khalid Abd, [1]Kazem Abhary and [1]Romeo Marian

[1]School of Engineering, University of South Australia, Mawson Lakes, 5095, South Australia
[2]Department of Production Engineering and Metallurgy,
School of Industrial Engineering, University of Technology, Baghdad, Iraq

ABSTRACT

This study describes the application of the developed model to the scheduling of the Robotic Flexible Assembly Cells (RFACs). The present study is a continuation of the paper titled "Development of a Fuzzy-Simulation Model of Scheduling Robotic Flexible Assembly Cells". This new model for multi-objective scheduling problems in RFACs was based on combining a fuzzy-based mathematical model with simulation software tools. In this study, a hypothetical case study of RFACs is presented to demonstrate the effectiveness of the developed model and then compare the results that are obtained by the new methodology with the common scheduling rules. The simulation results show that the performance of the proposed methodology outperforms the most popular scheduling rules from previous research.

Keywords: Robotic Cells, Scheduling, Fuzzy Logic, Simulation

1. INTRODUCTION

Due to increasing competition in the developing global economy, today's companies are facing greater challenge than ever to employ Flexible Manufacturing Systems (FMS) capable of dealing with unexpected events and meeting customers' requirements. Therefore, FMSs have attracted significant attention by both researchers and industrial practitioners in recent years. One class of such systems is called Robotic Flexible Assembly Cells (RFACs). RFACs is an integrated system, includes at least two of robotic assembly stations linked by an automated material handling system, all controlled by a central computer (Mohamed *et al*., 2001; Nof and Chen, 2003).

The design of RFACs with more than one robot offers many advantages. For example, efficiency due to a reduced work environment (Mohamed *et al*., 2001), increases robustness enabling the assembly of a variety of products using the same resources and flexibility due to superior ease of modification and reconfiguring (Marian *et al*., 2003). Accordingly, employing multi-robots in the RFACs offers the

advantage of increased productivity in a shorter cycle time with lower production costs (Xidias *et al*., 2010). Nevertheless, two robots (or more) operating simultaneously in the same work environment require a complex scheduling policy to prevent collisions between robots and other equipment in the cell (Nof and Chen, 2003). To overcome the difficulties, efficient scheduling of RFACs is required.

As mentioned in the previous work presented in paper titled "Development of a New Model of Scheduling Robotic Flexible Assembly Cells", The objective was to propose a new intelligent model of scheduling RFACs in a multi-product assembly environment, using fuzzy logic and simulation software.

The stated objective was achieved via two main steps. First, the relevant literature for the use of fuzzy logic approaches for scheduling problems in both conventional and flexible manufacturing systems were reviewed and then extracted the key points that be considered when developed a conceptual methodology for scheduling RFACs. Second, developed a new methodology for multi-objective scheduling problems in RFACs. This methodology was based on combining

Corresponding Author: Khalid Abd, School of Engineering, University of South Australia, Mawson Lakes, 5095, South Australia

a fuzzy-based mathematical model with simulation software. This study is on how to validate the developed model via an extensive example application and then comparison of the results those are obtained by the developed methodology with the common scheduling rules. The remainder of this study is organised as follows: The next section presents a case study of hypothetical RFACs to authenticate the developed methodology. In section 3, a fuzzy-based mathematical model is implemented to combine all input fuzzy variables in a new scheduling rule. In section 4, the new rule is applied using the MATLAB fuzzy logic toolbox. The simulation results for both most popular scheduling rules and new scheduling rule are provided in section 5. The conclusions and future research direction are given in section 6.

2. EXAMPLE APPLICATION OF SCHEDULING RFACS

The RFACs studied in this study consist of three main components (Abd *et al*., 2012a; 2012b): (1) Robots (R_1 and R_2) fetch the required parts and place them at assembly stations (S_1, S_2 and S_3) where the parts are assembled. (2) Part Feeder (PF) supplies parts to the cell. (3) Input and Output Conveyors (IC and OC) supply the base parts and carry out the final products, depicted in **Fig. 1**.

To provide a reliable solution to practical cases, six assumptions are considered in the simulation model. First, the optimum assembly sequence of each product is given in advance. Second, each product uses some or all of the cell resources. Third, each robot can perform only one task at a time. Fourth, each robot has multi-purpose end effectors. Fifth, no interruption, such as resources breakdown, occurs in the system. Sixth, the processing time of each task is deterministic and is known in advance.

In this system, four control points {C_1, C_2,…,C_4} are set to simplify path planning and avoid collisions between robots in the shared area. **Table 1** shows the robot paths and their required time to move between two positions in the cell.

The RFACs described above are assumed to assemble n product types (P_1, P_2,…,P_n). Each product is considered as an independent job. In this model, six products are taken as an example. **Table 2** shows the details of required stations along with the assembly operations time for each product type. This table also includes parts pick up and release times for the robots assembling the products.

In order to simulate RFACs, three customer orders are assumed and labelled as order #1, #2 and #3, as shown in **Table 3**. Orders #1 and #3 consist of six types of cell phone and order #2 is composed of only five types of products. Batch size and due date for each product type are given in **Table 3**.

3. A FUZZY LOGIC MODEL FOR SCHEDULING RFACS

A fuzzy-based mathematical model is developed to combine all input fuzzy variables in one scheduling rule. The input fuzzy variables include processing time, batch size, due date and number of required stations; the output fuzzy variable represents job priority. In this model, three major steps are considered. The next sub-section will explain in detail how these steps are performed.

3.1. Defining the Linguistic Variables

The first step is to define the linguistic inputs/output variables. Each linguistic variable is divided into a set of linguistic terms. For instance, if processing time is interpreted as a linguistic variable, to qualify the processing time, terms such as short, medium and long processing time are used in a real industry context. In this model, let us suppose that processing time, due date and batch size have three linguistic variables, number of required stations has two linguistic variables, while the output variable, product priority, has seven linguistic variables, as shown in **Table 4**.

3.2. Constructing Membership Functions

As mentioned earlier, triangular and trapezoidal shapes are the most well-known of membership functions. In this model, the input/output variables are constructed from triangular and/or trapezoidal shapes. For example, processing time is constructed as a triangular shape, as shown in **Fig. 2**.

The values of this membership function are between 0 and 1. The processing time value is evaluated based on time of all required tasks needed to complete the job.

The product priority, which represents the fuzzy output of the suggested model, takes both triangular and trapezoidal shapes. The product priority is assessed and measured based on the priority status of products to be sequenced, from the highest product priority to the lowest order priority. The categories in the assessed value of product priority are very low, low, below average, average, above average, high and very high, as depicted in **Fig. 3**.

Table 1. Transportation time for robots between cell resources

Path description	Position	Travel time
Robot move from resource to control point	S_1, PF→C_1, C_3, S_2, S_3, PF → C_2, C_4	0.5
Robot move from control point to resource	C_1, C_3→S_1, PF, C_2, C_4→ S_2, S_3, PF	1.0
Robot move between control point and conveyor	C_1, C_3 ←→ IC, C_1, C_3 ←→ OC	1.5
Robot move between two control points	C_1ß→C_2, C_3←→C_4	0.5

Table 2. Assembly operations requirements

Description	Assembly station	Time of assembly operations (sec)					
		P_1	P_2	P_3	P_4	P_5	P_6
Insert lens on front cover	S_1	4	3	3	4	3	4
Insert keypad on front cover	S_1	5	4	5	6	4	6
Assemble PC board with front cover	S_2	6	8	10	9	8	9
Insert antenna on back cover	S_3	9	0	0	9	0	0
Assemble back cover with front cover	S_2	7	11	10	11	7	10
Robot gripper pickup and release time (sec)		6	4	4	6	4	4

Table 3. Membership functions for input data and priority values

Product type	Order #1		Order #2		Order #3	
	Batch size	Due date	Batch size	Due date	Batch size	Due date
P_1	3	450	2	1200	4	1500
P_2	6	650	6	1300	5	1900
P_3	5	800	5	1400	3	1650
P_4	3	600	3	1000	3	1700
P_5	5	400	4	1100	3	1850
P_6	6	500	-	-	4	2000
Production volume	28		20		22	

Table 4. Input and output variables with their fuzzy values

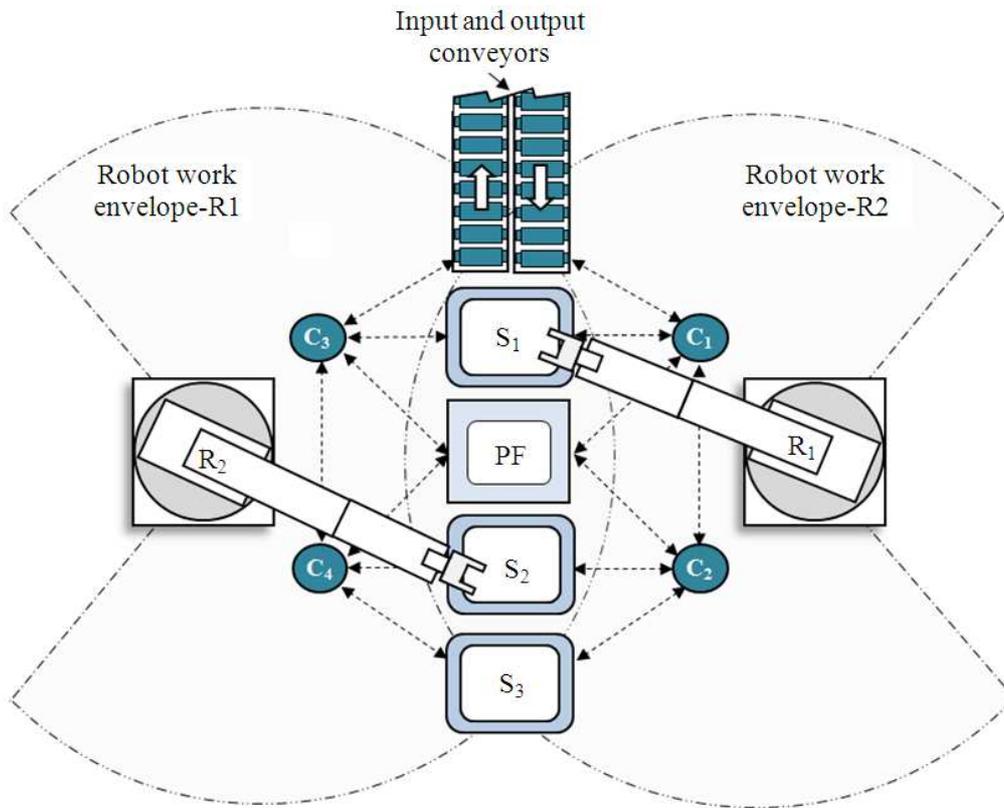| System variable	Linguistic variable	Linguistic value	Term set
Input	Processing time	Short	S
		Medium	M
		Long	L
	Batch size	Small	S
		Medium	M
		Large	L
		Short	S
	Due date	Medium	M
		Long	L
	Number of required stations	Low	L
		High	H
Output	Job priority	Very low	VL
		Low	L
		Below average	BA
		Average	A
		Above average	AA
		High	H
		Very high	HV

Fig. 1. A robotic flexible assembly cell
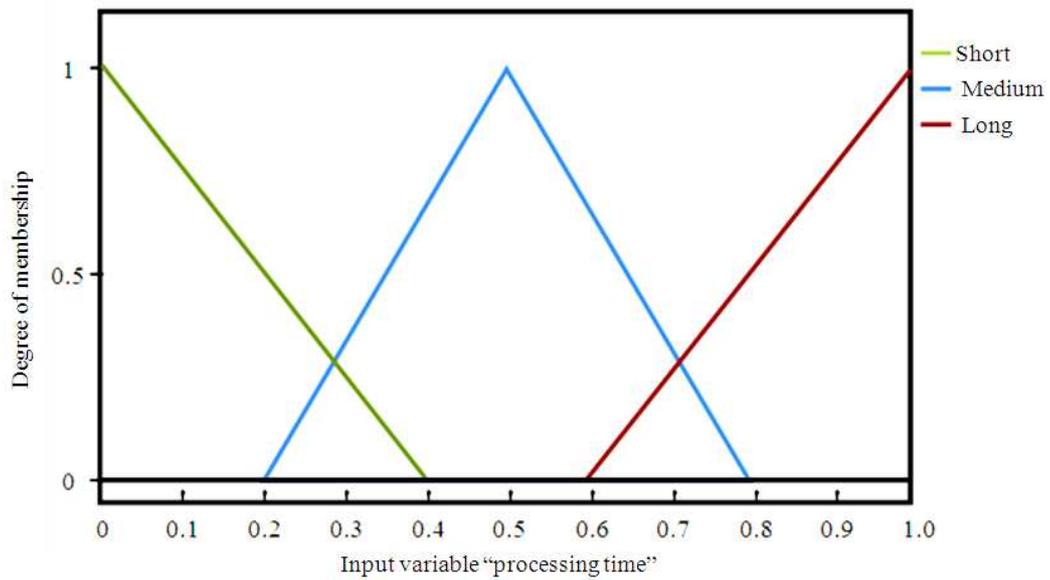

Fig. 2. Membership function for normalised "processing time" input
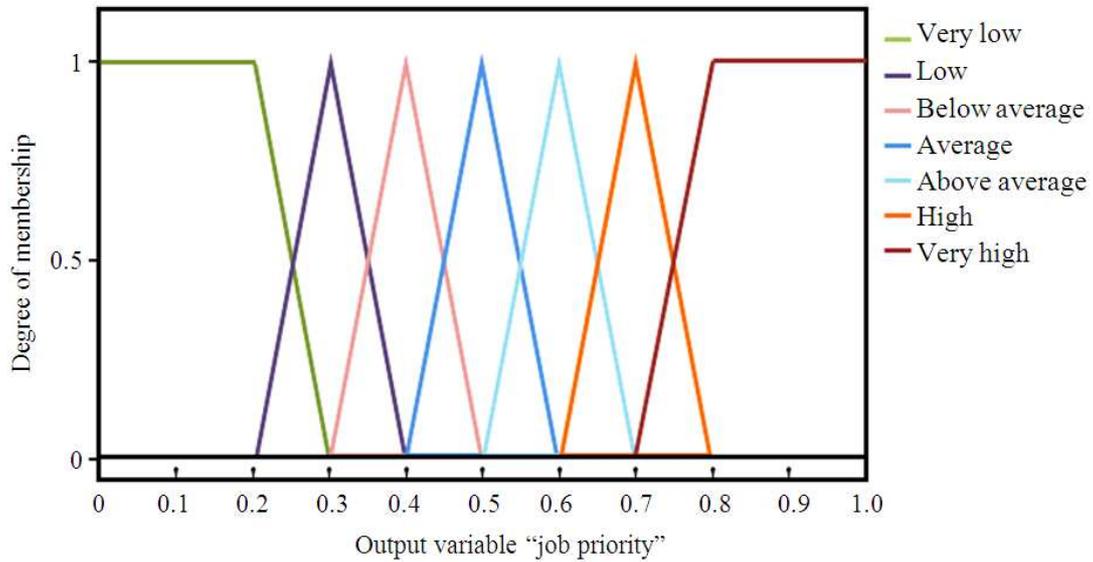
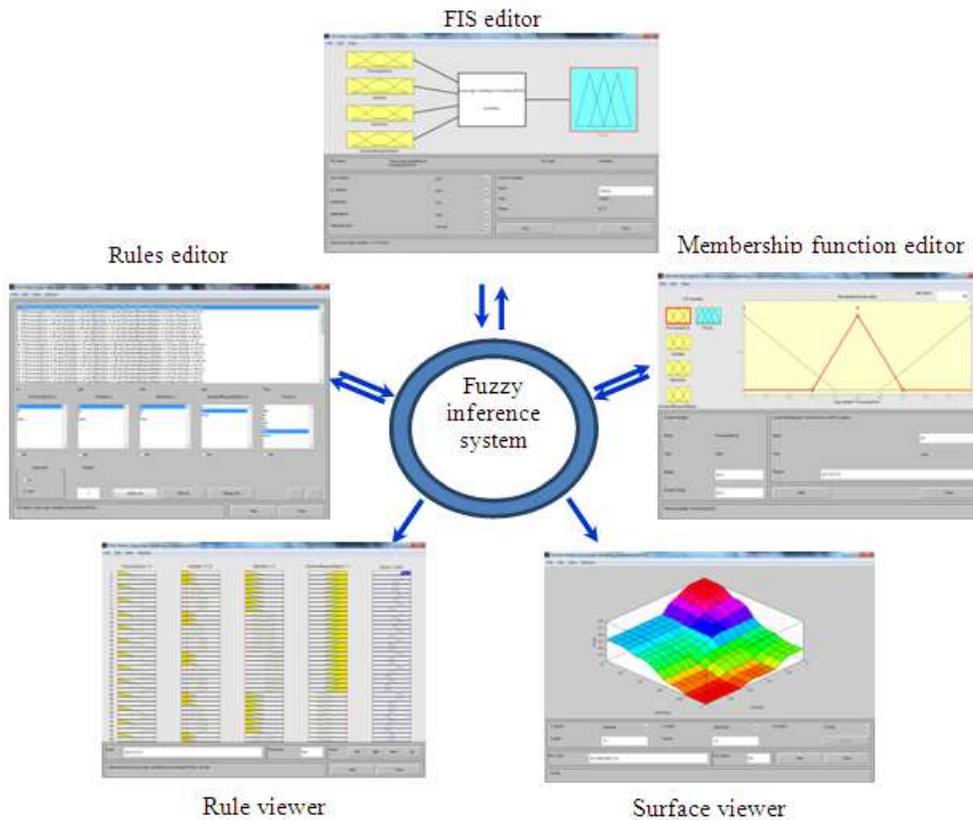**Fig. 3.** Membership function for normalised "Job Priority"

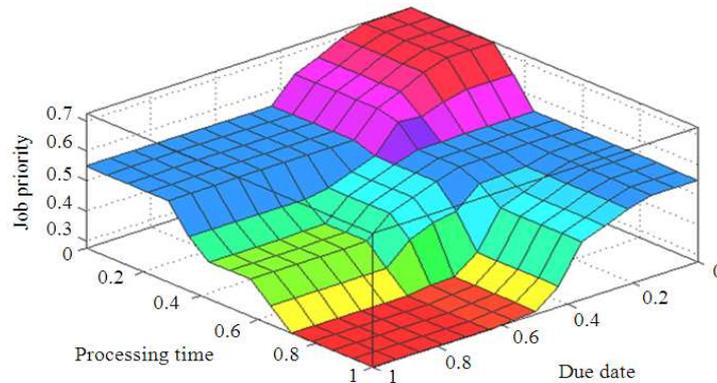**Fig. 4.** Fuzzy system and its integral components in MATLAB software

Fig. 5. Output surfaces of the FIS for processing time and due date

3.3. Constructing Fuzzy Rules

Fuzzy rules are structured to control the output variable. These rules can be provided by experts or may be extracted from numerical data. Since the variables of processing time, batch size and due date have three states each and the number of required stations has two states, the total number of fuzzy rules is fifty four ($3\times3\times3\times2 = 54$). The generic form of a fuzzy rule can be stated in the following form: IF (Processing Time is ▬) and (Due Date is ▬) and (Batch Size is ▬) and (Number of Required Stations is ▬) THEN (Priority is ▬). The black boxes represent the linguistic variables for each of the fuzzy variables.

4. IMPLEMENTATION OF FUZZY APPROACH

In this section, the scheduling proposed rule (FSR) is implemented using the MATLAB fuzzy logic toolbox. The fuzzy logic toolbox consists of five Graphical User Interface tools (GUIs) for building, editing and observing any fuzzy inference system (Sivanandam *et al.*, 2006; Mathworks, 2009). These tools are: the Fuzzy Inference System (FIS) editor, the membership function editor, the rule editor, the rule viewer and the surface viewer, as shown in **Fig. 4**. The GUIs are dynamically connected and the altering of any GUI can affect the other GUIs.

In this research, since the number of input variables is four, the number of generated 3D graphs is six. For instance, **Fig. 5** illustrates the priority resulting from the interaction of processing time and due date. From this figure it can be seen that the short processing time and small due date values give a high score of product priority. Moreover, it can be seen that the processing

time has a slightly higher influence than the due date on the product priority.

5. SIMULATION RESULTS AND DISCUSSION

In this section, the results of the simulation study are presented and discussed. The discussion will focus on analysing the results and comparing the RFACs performance based on the proposed rule (FSR) and existing scheduling rules. Five common performance measures, namely makespan, percentage of idle time, total tardiness, maximum tardiness and percentage of tardy jobs, are used to determine the performance of the RFACs. The simulation results of the overall performance measures of the three different orders and the average results are presented in **Fig. 6-8**. The paragraphs following discuss and analyse each performance measure individually.

One of the important measures of manufacturing system performance is makespan. Makespan represents the maximum completion time for the entire set of jobs. Shorter makespan results in due dates of customer orders being met, as well as a decrease in the direct production cost. **Fig. 6** shows the makespan results of scheduling rules for different customer orders. From this figure it can be seen that the developed Rule (FSR) obtains the best results for minimising the makespan, compared with the other scheduling rules. SPT and LPT rank second and third respectively. CR and EDD are the worst in minimising the makespan objective, for the reason that CR and EDD concentrate only on due dates of jobs and ignore the other variables such as processing time and batch size.
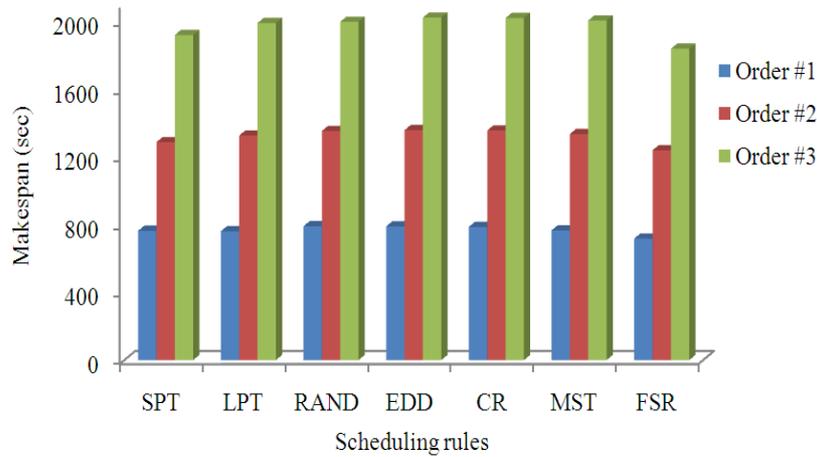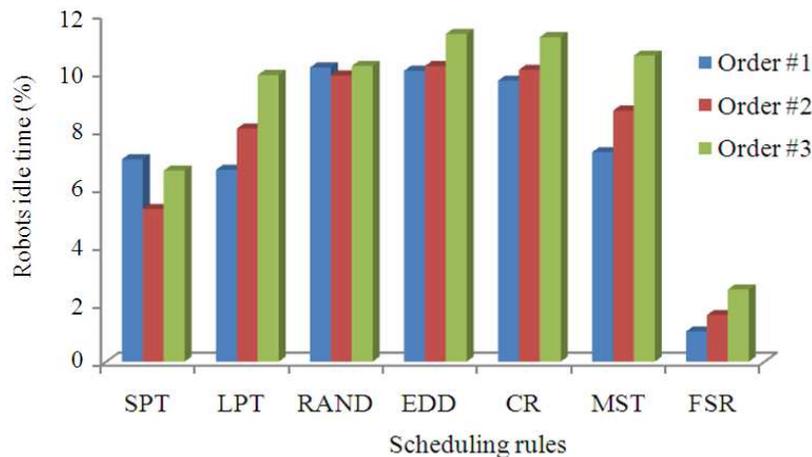
Fig. 6. Makespan for all orders

Fig. 7. Percentage of robots idle time for all orders

Robots idle time is an another important time based measure for scheduling evaluation. Since robots are a costly investment, it is vital to use them efficiently by reducing the idle time. This criterion enables a clear evaluation as to whether the robots are used in an efficient way. **Fig. 7** shows the percentage of idle time of scheduling rules on the three orders. In this figure, FSR emerges as the best rule among all seven scheduling rules, followed by SPT and LPT. SPT and LPT give good results for this measure. EDD appears to be the worst rule for minimising the robots' idle time. The reason for the poor performance of this rule is that the EDD rule concentrates only on the due date for the complete set of jobs and ignores the variable of

processing time. Total tardiness is another performance measure typically used in scheduling evaluation. This criterion represents the summation of jobs that fail to meet the due date. A higher total tardiness may result in loss of customers and competitiveness, as penalty for the late completion. The overall total tardiness of the scheduling rules on the three different orders is depicted in **Fig. 8**. In this figure, EDD appears to be the best rule among all seven scheduling rules. The second rank goes to the FSR. The difference between the results of EDD and FSR is insignificant. SPT, LPT and RAND are the worst in minimising the total tardiness criteria. This is because the due date variable is ignored by these rules.
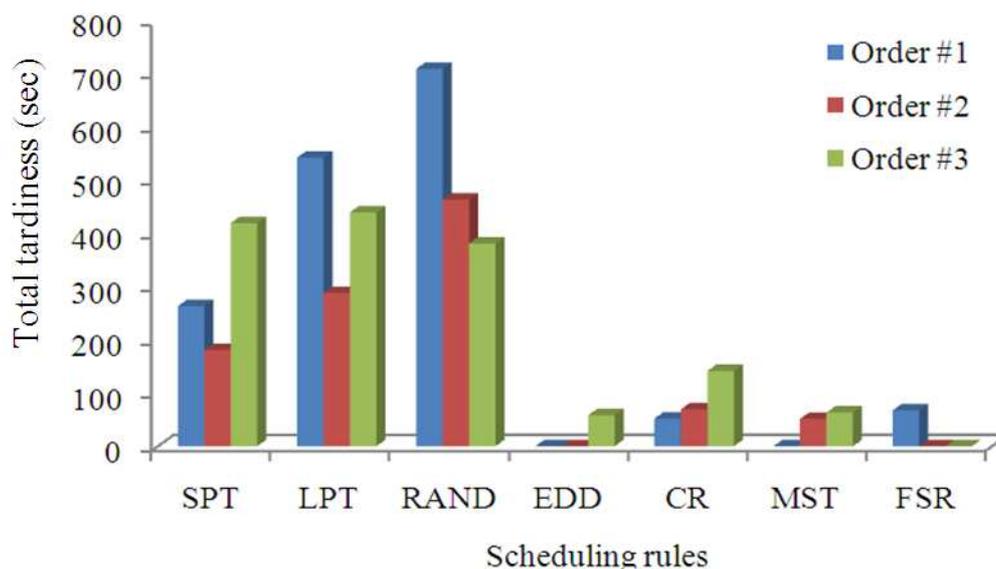
Fig. 8. Total tardiness for all orders

From the above results and discussion, it can be concluded that FSR is generally better than all other common scheduling rules. This is because the developed rule is constructed by combining all input variables such as processing time, due date, batch size.

6. CONCLUSION

To demonstrate the effectiveness of the developed model, a hypothetical case study of RFACs was used. The scenario presented is realistic and of an average level of difficulty. The simulation results revealed that the performance of the proposed fuzzy-based mathematical model was more efficient compared to the heuristic scheduling rules (SPT, LPT, RAND, EDD, CR and MST), based on all five of above the mentioned objectives. In addition, the results concluded that the heuristic scheduling rules were not suitable for finding an acceptable schedule regarding multi-objective criteria. For example EDD was efficient at due date based objectives; but it was least efficient with time based objectives.

7. ACKNOWLEDGMENT

8. REFERENCES

Abd, K., K. Abhary and R. Marian, 2012a. Intelligent modeling of scheduling robotic flexible assembly cells using fuzzy logic. Proceedings of the 12th WSEAS International Conference on Robotics, Control and Manufacturing Technology, (CMT' 12), World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA., pp: 202-207.

Abd, K., K. Abhary and R. Marian, 2012b. Efficient scheduling rule for robotic flexible assembly cells based on fuzzy approach. Proc. CIRP, 3: 483-488. DOI: 10.1016/j.procir.2012.07.083

Marian, R., A. Kargas, L. Luong and K. Abhary, 2003. A framework to planning robotic flexible assembly cells. Proceedings of the 32nd International Conference on Computers and Industrial Engineering, (CIE' 03), Gemini International Limited, Limerick, Ireland, pp: 607-615.

Mathworks, 2009. Fuzzy Logic Toolbox User's Guide. MathWorks, Inc.

Mohamed, S.B., D.J. Petty, D.K. Harrison and R. Rigby, 2001. A cell management system to support robotic assembly. Int. J. Adv. Manufact. Technol., 19: 598-604. DOI: 10.1007/s001700170037

Nof, S.Y. and J. Chen, 2003. Assembly and disassembly: An overview and framework for cooperation requirement planning with conflict resolution. J. Intell. Robot. Syst., 37: 307-320. DOI: 10.1023/A:1025466401869

Sivanandam, S.N., S. Sumathi and S.N. Deepa, 2006. Introduction to Fuzzy Logic using MATLAB. 1st Edn., Springer, Berlin, ISBN-10: 3540357815, pp: 444.

Xidias, E.K., P.T. Zacharia and N.A. Aspragathos, 2010. Time-optimal task scheduling for articulated manipulators in environments cluttered with obstacles. J. Syst. Control Eng., 224: 845-855. DOI: 10.1017/S0263574709005748