

INTELLIGENT MODEL OF SCHEDULING RFACS - PART II: APPLICATION

ABD, K.; ABHARY, K. & MARIAN, R.

Abstract: *This chapter is a continuation of the chapter titled “Intelligent Model of Scheduling RFACS - Part I: Methodology and Strategy” with the emphasis on application of the developed methodology to the scheduling of the RFACS. Based on the methodology and strategy presented in Part I, a new model for multi-objective scheduling problems in RFACS was developed. This model was based on combining a fuzzy-based mathematical model with simulation software tools. In this chapter, a hypothetical case study of RFACS is presented to prove the effectiveness of the developed methodology, and then compare the results that are obtained by the new methodology with the common scheduling rules. Simulation results show that the performance of the proposed methodology outperforms the most popular scheduling rules from literature.*

Key words: *Robotic cells, scheduling, fuzzy logic, simulation*



Authors' data: MEng, Dipl.-Ing. Abd, K[hali]; A/Prof Dr.-Ing. Abhary, K[azem]*, Dr.-Ing. Marian, R[omeo], *University of South Australia, School of Engineering, Mawson Lakes Campus, Mawson Lakes, SA 5095, Australia, khalid.abd@mymail.unisa.edu.au, kazem.abhary@unisa.edu.au, romeo.marian@unisa.edu.au

This Publication has to be referred as: Abd, K[hali]; Abhary, K[azem] & Marian, R[omeo] (2013) Intelligent Model of Scheduling Rfacs - Part II: Application, Chapter 44 in DAAAM International Scientific Book 2013, pp. 737-750, B. Katalinic & Z. Tekic (Eds.), Published by DAAAM International, ISBN 978-3-901509-94-0, ISSN 1726-9687, Vienna, Austria

DOI: 10.2507/daaam.scibook.2013.44

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 (Marian et al. 2003), and flexibility due to superior ease of modification and reconfiguring (Makino 1989). 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 these difficulties, efficient scheduling of RFACs is required.

As mentioned in the previous work presented in Part I, 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 a fuzzy-based mathematical model with simulation software (Abd et al. 2013).

This chapter is a continuation study of Part I on how to to validate the developed methodology via an extensive example application, and then comparison of the results that are obtained by the developed methodology with the common scheduling rules. The remainder of this chapter 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 chapter consist of three main components (Abd et al. 2012a, Abd et al. 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 & OC) supply the base parts and carry out the final products, depicted in Figure 1.

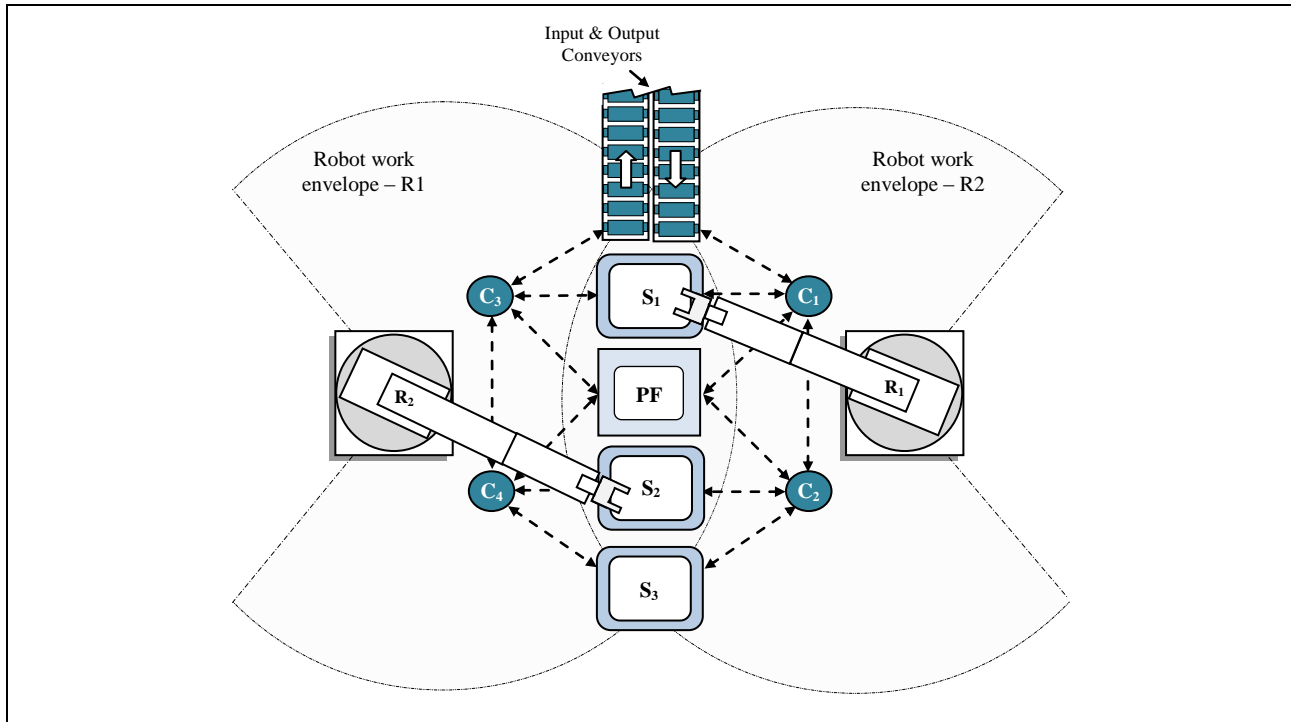


Fig. 1. A robotic flexible assembly cell

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 $\{C_1, C_2, \dots, C_4\}$ 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, \dots, 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. Table 4 shows the value of the membership functions for input data.

Path description	Position	Travel Time
Robot move from resource to control point	$S_1, PF \rightarrow C_1, C_3$ $S_2, S_3, PF \rightarrow C_2, C_4$	0.5
Robot move from control point to resource	$C_1, C_3 \rightarrow S_1, PF$ $C_2, C_4 \rightarrow S_2, S_3, PF$	1
Robot move between control point and conveyor	$C_1, C_3 \leftrightarrow IC$ $C_1, C_3 \leftrightarrow OC$	1.5
Robot move between two control points	$C_1 \leftrightarrow C_2$ $C_3 \leftrightarrow C_4$	0.5

Tab. 1. Transportation time for robots between cell resources

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 & release time (sec.)		6	4	4	6	4	4

Tab. 2. Assembly operations requirements

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	

Tab. 3. Membership functions for input data and priority values

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 subsection 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 5.

System variable	Linguistic variable	Linguistic Value	Term Set
Input	Processing Time	Short Medium Long	S M L
	Batch Size	Small Medium Large	S M L
	Due Date	Short Medium Long	S M L
	Number of Required Stations	Low High	L H
Output	Job Priority	Very Low Low Below Average Average Above Average High Very High	VL L BA A AA H HV

Tab. 4. Input and output variables with their fuzzy values

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.

Processing time is constructed as a triangular shape, as shown in Figure 3.10. 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.

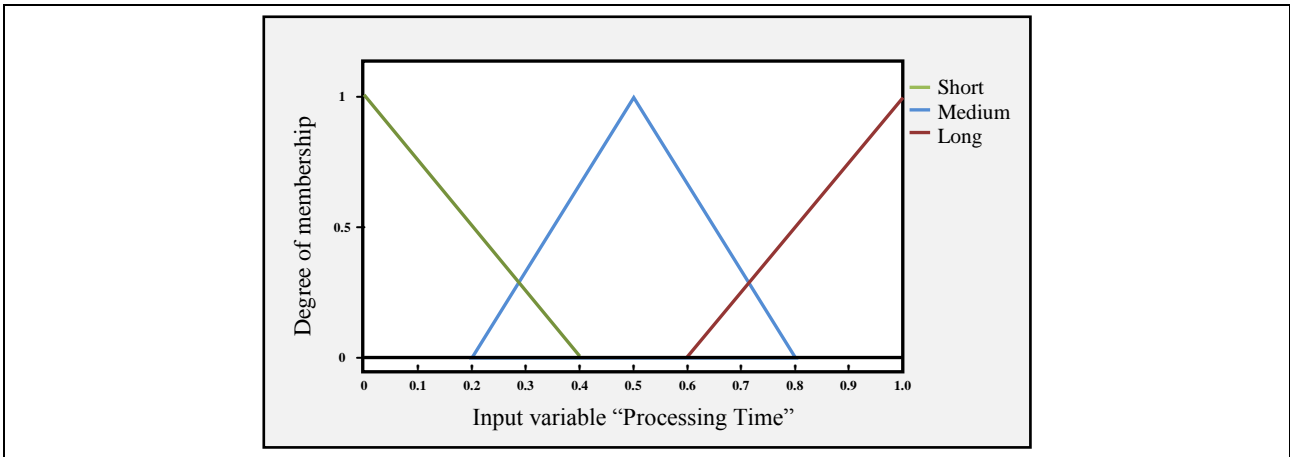


Fig. 2. Membership function for normalised “Processing Time” input

Batch size is also constructed as a triangular shape. The batch size value is measured and evaluated through the required quantity of each job. There are three categories in the quantity of batch size, namely small, medium and large as shown in Figure 3.

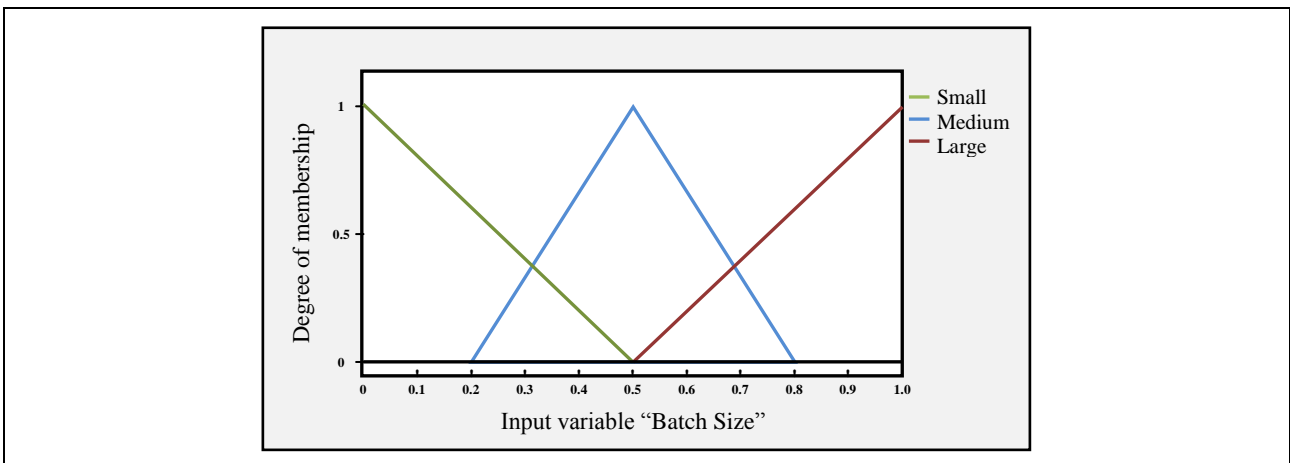


Fig. 3. Membership function for normalised “Batch Size” input

Due date is built from both triangular and trapezoidal shapes. The due date value is measured based on the deadline of production for each job. The categories in the assessed value of due date are *small*, *medium* and *large*, as depicted in Figure 4.

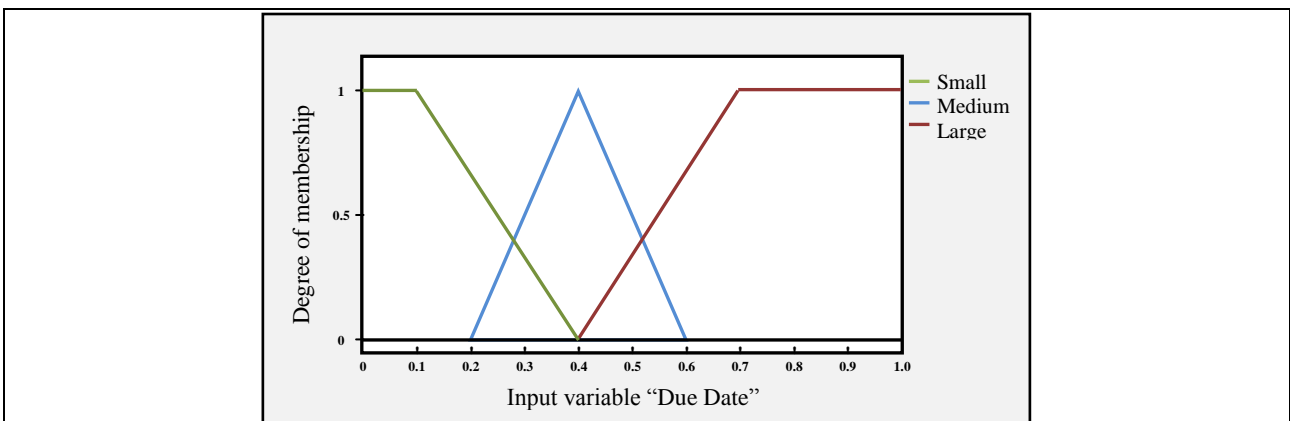


Fig.4. Membership function for normalised “Due Date” input

Number of required stations is constructed as a trapezoidal shape. The number of required stations value is measured and assessed based on the number of essential stations required for completing the job tasks. There are two categories in the number of required stations, namely *low* and *high* as shown in Figure 5.

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 Figure 6.

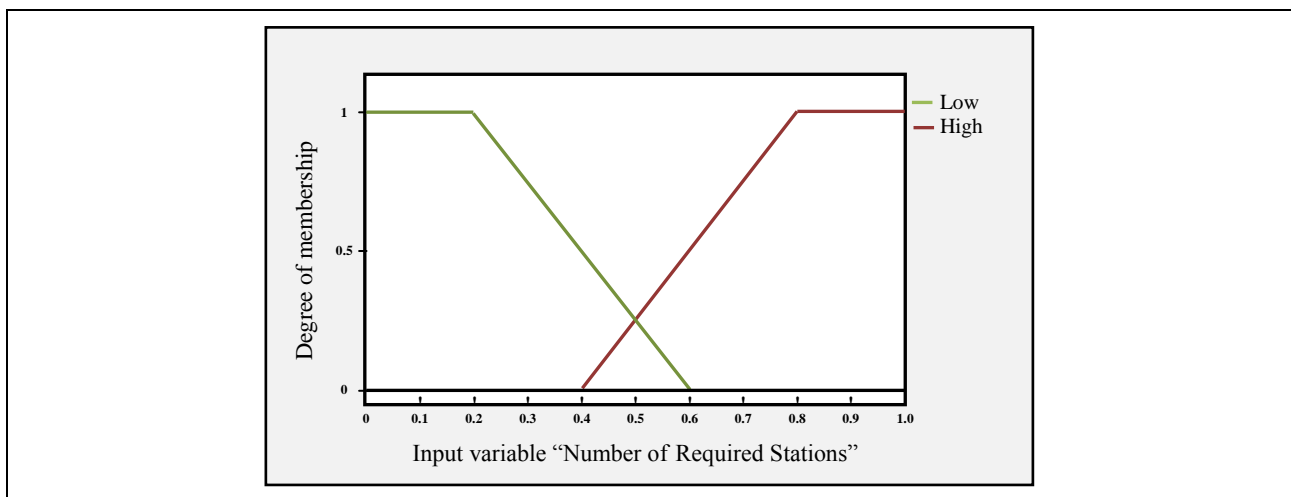


Fig. 5. Membership function for normalised “Number of Req. Stations” input

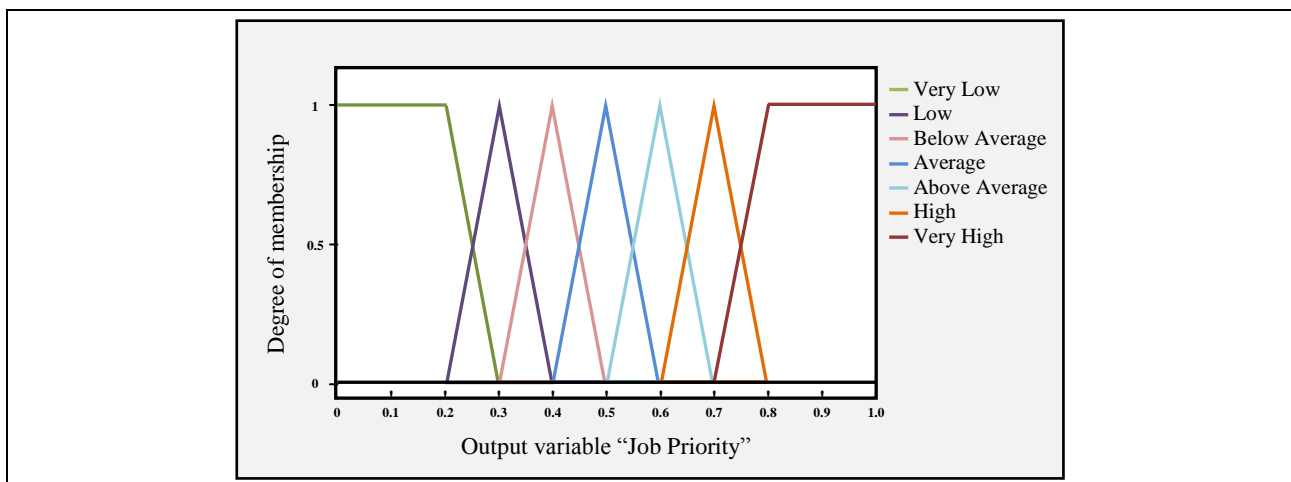


Fig. 6. Membership function for normalised “Job Priority”

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 ($3 \times 3 \times 3 \times 2 = 54$). Table 5 demonstrates the 54 rules. 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.

Batch Size “Small”

Number of Stations “High”		Processing time			Number of Stations “Small”		Processing time		
		Short	Medium	Long			Short	Medium	Long
Due date	Short	VH	VH	H	Due date	Short	AA	AA	A
	Medium	H	H	AA		Medium	A	A	BA
	Long	H	AA	A		Long	A	BA	BA

Batch Size “Medium”

Number of Stations “High”		Processing time			Number of Stations “Small”		Processing time		
		Short	Medium	Long			Short	Medium	Long
Due date	Short	VH	H	AA	Due date	Short	A	A	BA
	Medium	AA	AA	A		Medium	BA	L	L
	Long	AA	A	A		Long	BA	BA	VL

Batch Size “Large”

Number of Stations “High”		Processing time			Number of Stations “Small”		Processing time		
		Short	Medium	Long			Short	Medium	Long
Due date	Short	AA	AA	A	Due date	Short	BA	BA	L
	Medium	A	BA	BA		Medium	L	VL	VL
	Long	A	BA	BA		Long	L	VL	VL

Tab. 5. Fuzzy rules matrix

4. Implementation of Fuzzy Approach for the Scheduling of RFACs

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., 2007, 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 Figure 7. The GUIs are dynamically connected, and the altering of any GUI can affect the other GUIs.

In the fuzzy logic toolbox, the fuzzy inference system (FIS) editor handles the information related to the variables of inputs and output, such as variables' names and their numbers. In this research, four fuzzy inputs variables, processing time, batch size, due date and number of required stations, are defined. The job priority is the fuzzy output variable, representing the priority status of a product to be selected for the next assembly operation in the RFACs. The membership function editor is used to construct the shapes of all the input/output variables. The membership function shape of each fuzzy variable was extensively described in section 3.2. The rule editor is for editing the list of fuzzy rules that are used to control the output variable. In section 3.3, the fuzzy rule is constructed based on the number of linguistic variables for inputs/output (see section 3.1 for definition). The surface viewer allows the user to visualise the relation between input fuzzy variables and the output of a fuzzy system

in a three-dimensional graph, the X-axis and Y-axis in the 3D graph represent any two selected input variables, and the Z-axis represents the output of a fuzzy system.

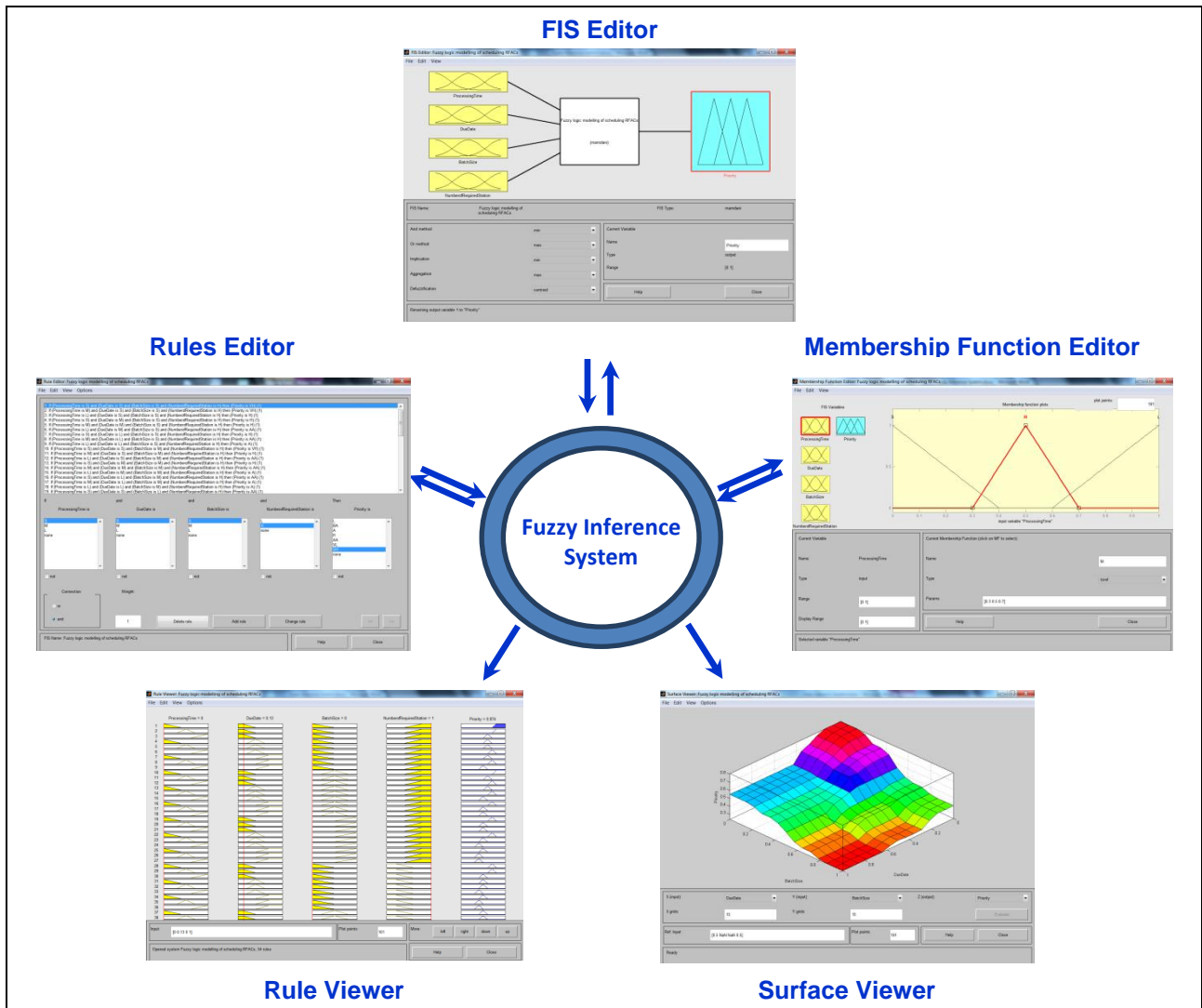


Fig. 7. Fuzzy system and its integral components in MATLAB software

In this research, since the number of input variables is four, the number of generated 3D graphs is six. For instance, Figure 8 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.

The rule viewer, which displays a graphical representation of the values of the input variables and the output of a fuzzy system through all the fuzzy rules, is shown as an example in Figure 9. The output (job priority) in this figure can be interpreted easily as follows: IF the μ_T^i is (0), the μ_D^i is (0.13), the μ_N^i is (0), and the μ_S^i is (1) THEN *priority* will be (0.87).

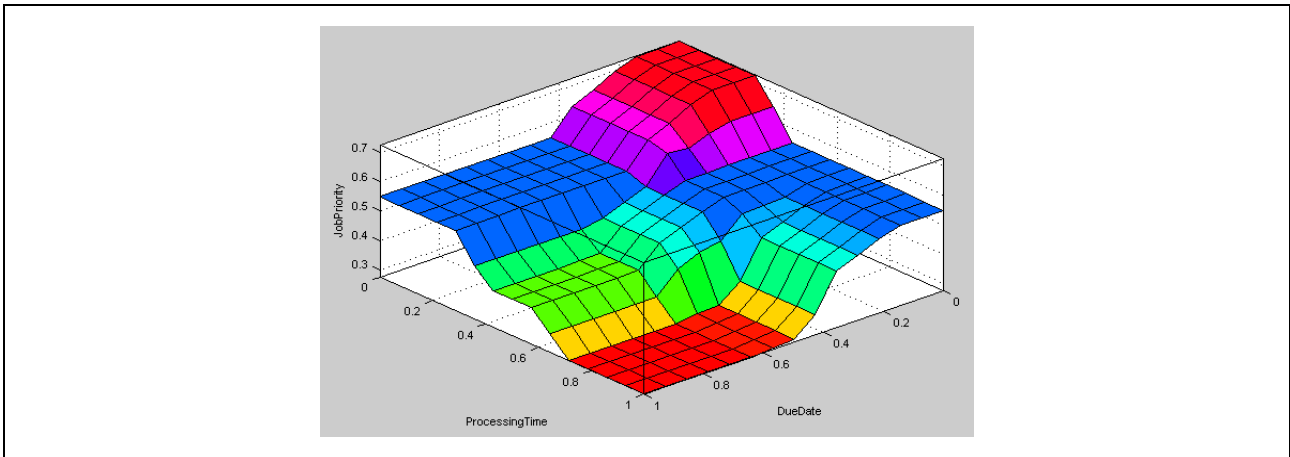


Fig. 8. Output surfaces of the FIS for processing time and due date

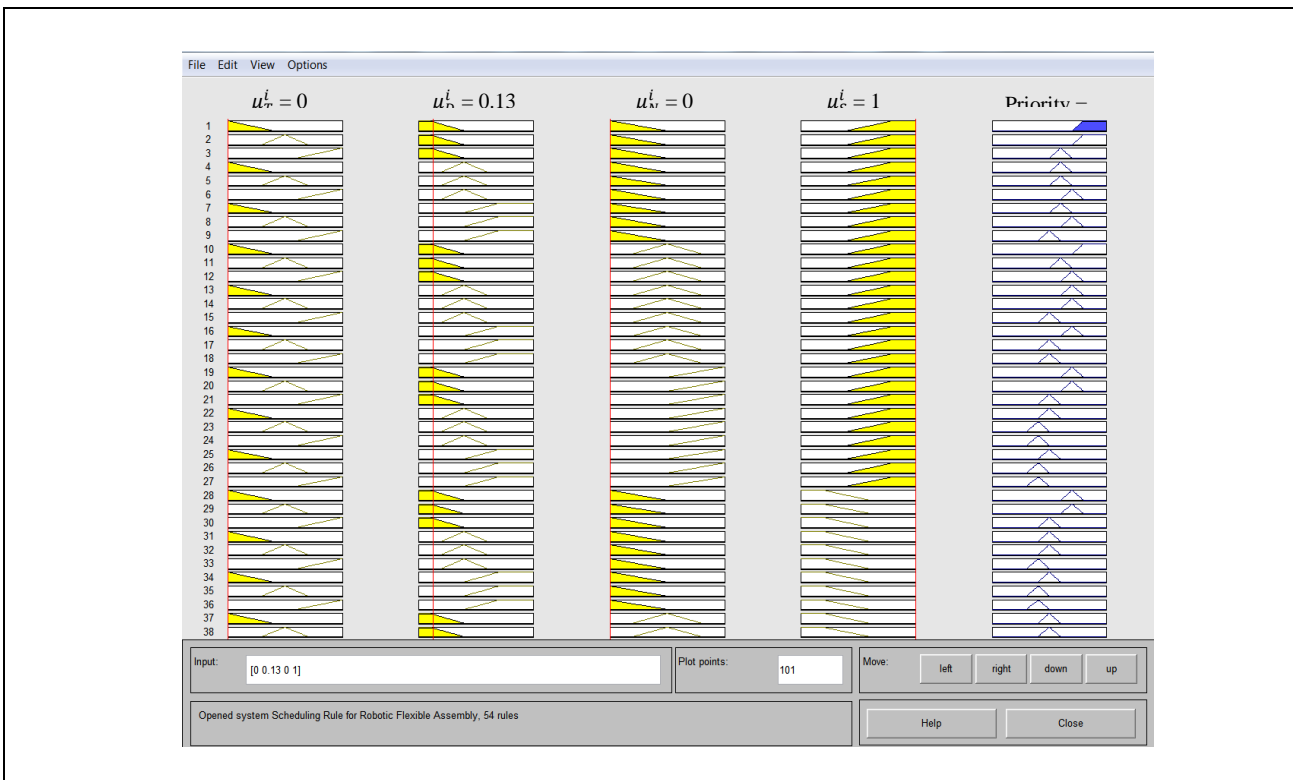


Fig. 9. Rule viewers of the FIS

5. Simulation Results and Discussion

In this section, the experimental design is set. Each experiment is performed with different scheduling rule. Seven experiments are implemented. Experiments numbered 1 to 6 are run with existing scheduling rules; Experiment 7 is run using the developed rule. These rules are: short processing time (SPT), long processing time (LPT), random (RAND), earlier due date (EDD), critical ratio (CR), minimise slack time (MST) and fuzzy sequencing rule (FSR). The selected rules generate different sequences of product flow to the system.

The results of the simulation study are 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. As mentioned earlier, three customer orders, as shown in Table 3, are assumed in order to simulate RFACs.

The simulation results of the overall performance measures of the three different orders and the average results are presented in Figure 10 to 14. 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.

Figure 10 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.

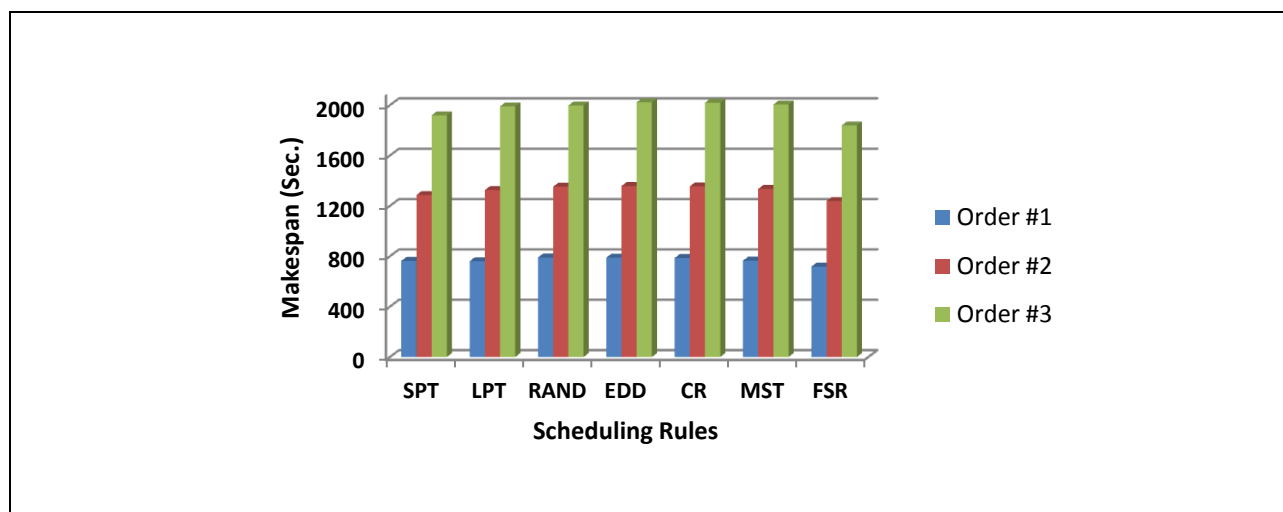


Fig. 10. Makespan for all orders

Robots idle time is an 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. Figure 11 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 Figure 12. 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.

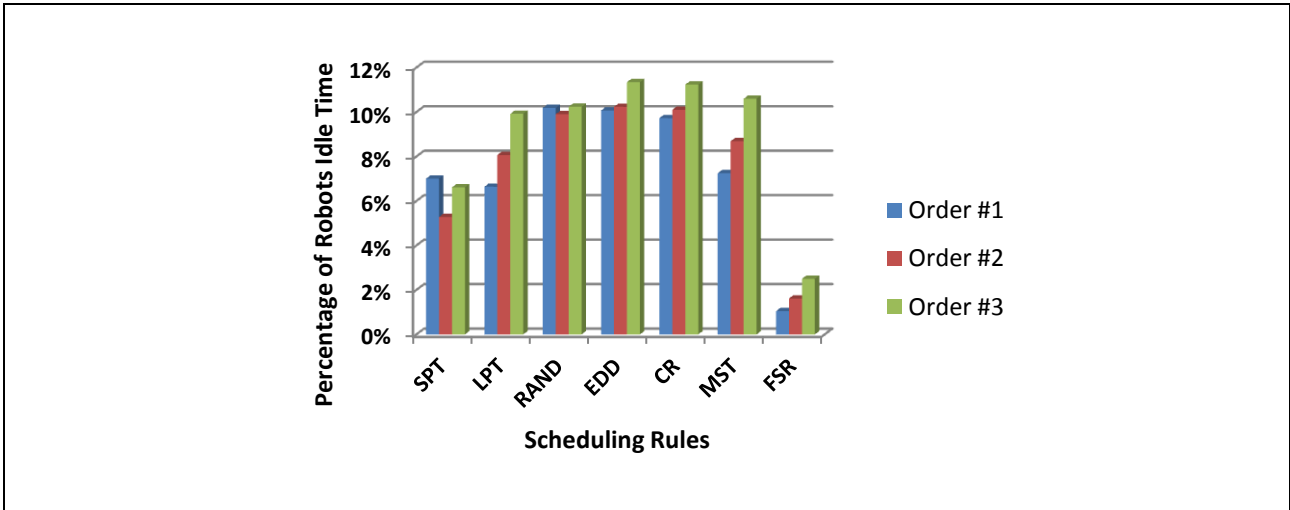


Fig. 11. Percentage of robots idle time for all orders

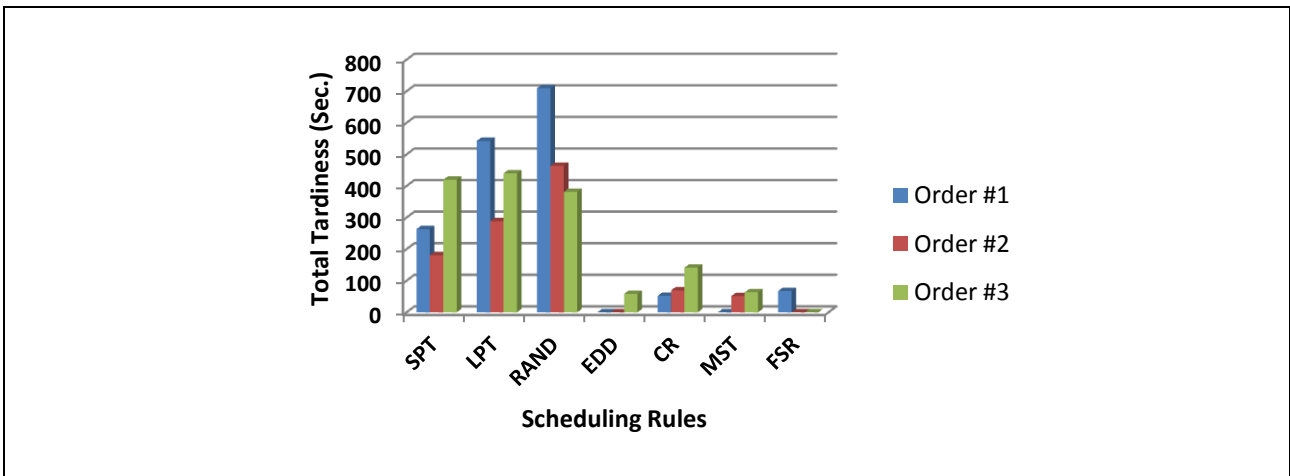


Fig. 12. Total tardiness for all orders

Maximum tardiness is another criterion related to the jobs' due dates. Maximum tardiness represents the largest difference between the final fulfilment date for jobs and the requested delivery date. Figure 13 shows the maximum tardiness of the scheduling rules on all three orders. The EDD rule emerges as the best for minimising the maximum tardiness of the scheduling rules. FSR obtains a slightly lower performance compared with EDD. MST and CR rank third and fourth respectively. LPT, SPT and RAND are observed to be the worst performing rules for this measure, again for the reason that the due date is ignored by LPT, SPT and RAND; also LPT and SPT focus only on the processing time of jobs. Like total tardiness and maximum tardiness, the percentage of tardy jobs is criterion-related to the jobs' due dates. This criterion is defined as the ratio of the number of tardy jobs to the number of jobs. The overall percentages of tardy jobs for scheduling rules on different orders are

presented in Figure 14. According to this Figure, FSR obtains the best results for minimising the percentage of tardy jobs. EDD ranks second, and still obtains good results among all the other selected rules from the previous literature. LPT and RAND perform poorly in minimising the percentages of tardy jobs in almost all orders. The poor performance of these rules is expected because they do not deal with the due date or batch size of jobs.

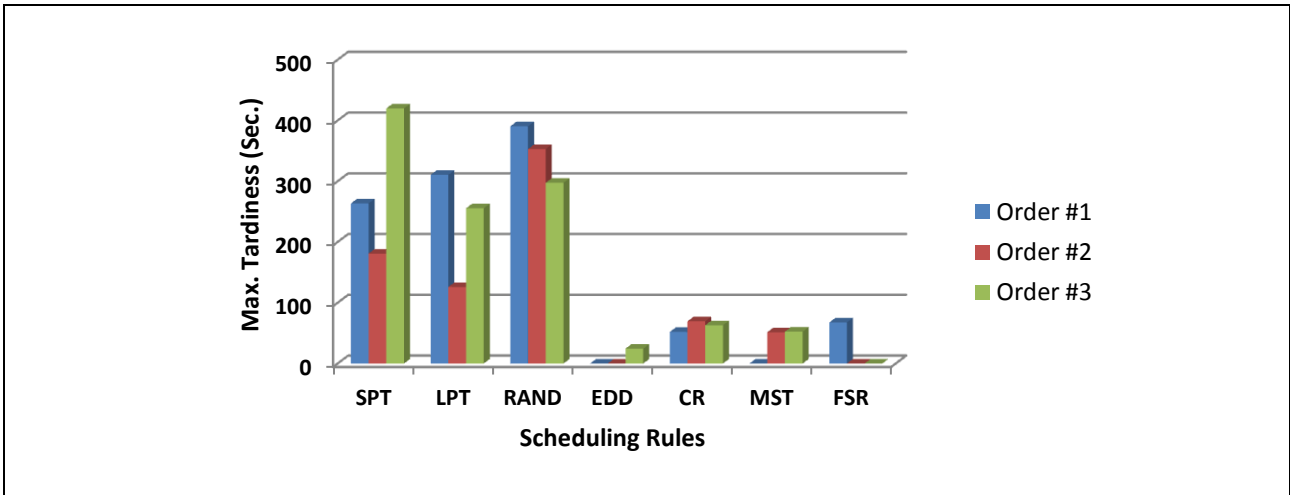


Fig. 13. Maximum tardiness for all orders

In terms of time based measures, the simulation results show that the developed rule (FSR) outperforms all the other selected rules from literature. SPT and LPT obtain acceptable performance. EDD is observed to be the worst performing rule for time based measures. With respect to due date based measures, it can be seen that the EDD rule appears to be the best in minimising these measures; on the other hand it performs poorly in minimising time based measures. Also, FSR proves very effective in minimising the due date based measures. 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.

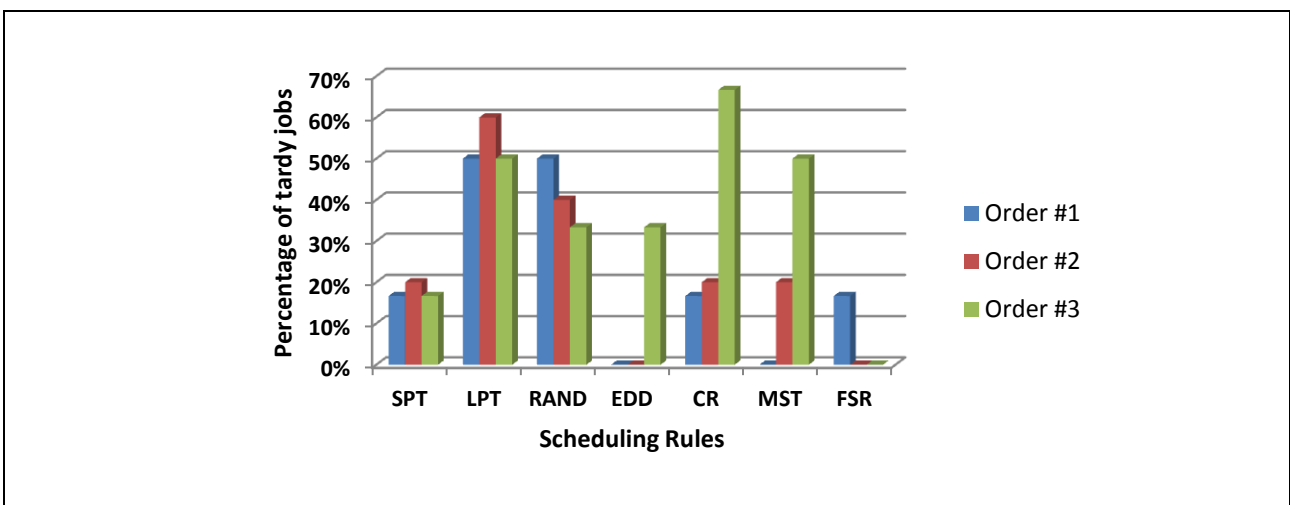


Fig. 14. Percentages of tardy jobs for all orders

6. Conclusion

In order to demonstrate the effectiveness of the developed methodology presented in Part I, a hypothetical case study of RFACs was used. The scenario presented in the case study 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; nevertheless it was worst with time based objectives.

For future work, two further stages are planned. The first is to integrate the Taguchi method with a fuzzy logic approach to achieve multi-objective optimisation of scheduling problems in RFACs. This can be done by developing a multiple performance characteristics index (MPCI) based on fuzzy logic approach to derive the optimal solution. The other stage is to expand the proposed study to be more applicable to real scheduling problems, by taking into account robot breakdowns during the scheduling of RFACs. Three parameters may be considered in the scope of this area of study: (1) the number of robots subject to breakdown (2) breakdown frequency which represents how many times each robot will break down during the scheduling horizon (3) repair duration which represents the mean time required to repair the robot after its breakdown.

7. References

- Abd, K., Abhary, K. & Marian, R. 2013. Intelligent Model of Scheduling RFACs - Part I: Methodology and Strategy, In: *DAAAM International Scientific Book*
- Abd, K., Abhary, K. and Marian, R., 2012a. Intelligent modeling of scheduling robotic flexible assembly cells using fuzzy logic, *12th WSEAS International Conference on Robotics, Control and Manufacturing Technology*. Rovaniemi, Finland. 201-207
- Abd, K., Abhary, K. and Marian, R., 2012b. Efficient scheduling rule for robotic flexible assembly cells based on fuzzy approach. *45th CIRP Conference on Manufacturing Systems*, 3, 483-488
- Marian, R. M., Kargas, A., Luong, L. H. S. & Abhary, K. 2003. A framework to planning robotic flexible assembly cells *32nd International Conference on Computers and Industrial Engineering*. Limerick, Ireland
- Mathworks 2009. Fuzzy Logic Toolbox User's Guide. http://www.mathworks.com/access/helpdesk/help/pdf_doc/fuzzy/fuzzy.pdf.
- Mohamed, S.B., Petty, D.J., Harrison, D.K. and Rigby, R., 2001. A cell management system to support robotic assembly. *International Journal of Advanced Manufacturing Technology*, 19, 598-604
- Nof, S. Y. and Chen, J. 2003. Assembly and disassembly: an overview and framework for cooperation requirement planning with conflict resolution. *Journal of Intelligent and Robotic Systems*, 37, 307-320
- Sivanandam, S. N., Sumathi, S. & Deepa, S. N. 2007. *Introduction to Fuzzy Logic using MATLAB*, New York, Springer
- Xing, L. N., Chen, Y. W., Wang, P., Zhao, Q. S. & Xiong, J. 2010. A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing*, 10, 888-896