A Methodology for Scheduling Robotic Flexible Assembly Cells Using Fuzzy Logic and Simulation

K. Abd, K. Abhary, R. Marian

Abstract— **Due to a developing global economy, today's companies are facing a greater challenge than ever to employ flexible manufacturing systems (FMS) capable of dealing with unexpected events and meeting customers' requirements. One of these systems is called robotic flexible assembly cells (RFACs). There has been relatively little work on the scheduling RFACs, even though overall scheduling problems of FMS have attracted significant attention. This study aims to develop an efficient methodology for scheduling RFACs. The proposed scheduling methodology is divided into three modules: pre-processing, scheduling and simulation. Three performance measures are considered: makespan and robots idle time and total tardiness. Simulation results show that the proposed methodology outperforms the common scheduling rules.**

***Index Terms*—Robotic cells, scheduling, fuzzy logic, simulation**

I. Introduction

Robotic Flexible Assembly Cells (RFACs) are highly modern systems, structured with industrial robot(s), assembly stations and an automated material handling system, all monitored by computer numerical control [1-3]. The design of RFACs with more than one robot offer many advantages over single robot. For example, efficiency due to a reducing work environment [4], increased robustness to assemble a variety of products using the same resources [2], and additional flexibility due to superior ease of modification and reconfiguring [5]. Accordingly, employing multi-robots in the RFACs offer the advantages of increased productivity in a shorter cycle time with lower production costs [6]. Nevertheless, there are certain difficulties that have arisen with this design. For example, two robots (or more) operating simultaneously in the same work environment require a complex control system to prevent collisions between robots and other equipment's in the cell [7]. Also, industrial robots must be employed effectively [6].

This work was supported in part by the School of Advanced Manufacturing and Mechanical Engineering, University of South Australia.

Khalid Abd is a PhD student, School of Engineering, University of South Australia, Mawson Lakes, SA 5095, South Australia, AUSTRALIA (e-mail: abdkk001@mymail. unisa. edu.au).

A/Prof. Kazem Abhary is an Associate Professor in Mechanical & Manufacturing Engineering with the School of Engineering, University of South Australia, Mawson Lakes, SA 5095, South Australia, AUSTRALIA (e-mail: kazem.abhary@ unisa.edu.au).

Dr. Romeo Marian is Senior Lecture/Program Director with the School of Engineering, University of South Australia, Mawson Lakes, SA 5095, South Australia, AUSTRALIA (e-mail: romeo.marian@unisa.edu.au).

To overcome the above difficulties, an efficient scheduling methodology of RFACs is required.

Few studies have been done on the problem of scheduling in the RFACs. These studies may be categorised into three groups. First, the studies which applied heuristic approaches to solve scheduling problems such as Lee and Lee [8], Nof and Drezner [9], Lin et al. [10], Pelagagge et al. [11], Sawik [12], Jiang et al. [13] and Rabinowitz et al. [14]. Second, the studies which investigated simulation as an approach to scheduling RFACs, for instance, Glibert et al. [15], Hsu and Fu [16] and Basran et al. [17]. Third, only two studies, Brussel et al. [18] and Dell Valle and Camacho [19], implemented expert systems approaches to solve scheduling problems. Based on the previous studies, the major limitation is that these studies are arranged to assemble only one product type. In our recent study [20]-[22], scheduling RFACs for concurrent assembly of multi-products has been proposed using common scheduling rules.

Scheduling rules are employed to improve the system performance such as minimise makespan, minimise tardiness or maximise throughput [18]. Unfortunately, the common rules are not satisfied to optimise most of performance measures. Therefore, an efficient scheduling strategy is required to satisfy multiple performance measures in the RFACs. The aim of this study is to develop a new scheduling methodology based on fuzzy logic and simulation for scheduling RFACs in a multi-product assembly environment.

II. Proposed Methodology

The scheduling of the RFACs requires finding a way which determines how to use cell resources in an optimal manner to assemble multi-products. Let us consider an assembly cell in which a set of tasks are performed using a set of resources to assemble multi-products concurrently.

- Tasks represent any physical activities that are carried out by utilising resources. The tasks can be categorised into four types: move, tool-change, pick-up and assembly.
- Multi-products of the same family group usually involve similar operations; however, there are some differences in the assembly operations and the operational sequences among these products.

The developed methodology has three major steps: pre-processing, scheduling and simulation and module. The architecture of the proposed methodology is illustrated in Fig. 1. The next sub sections will present these three modules in more detail.

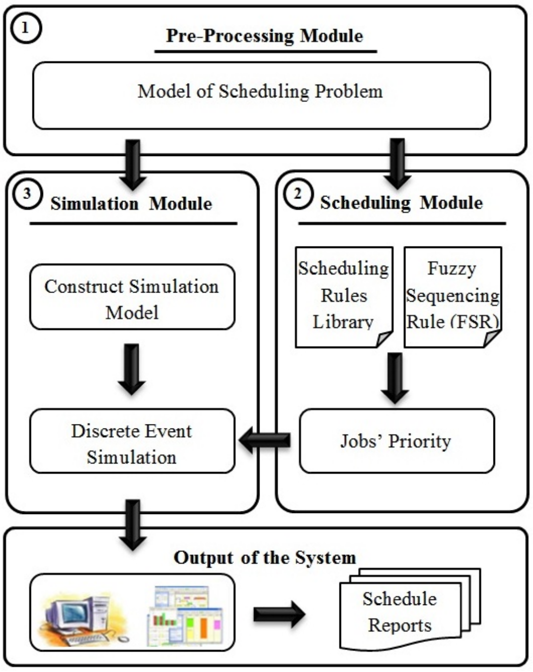

Fig. 1. Architecture of proposed methodology for RFACs scheduling

A. Pre-processing module

The aim of the pre-processing module is to describe all the required components of the scheduling problems in the RFACs. These components are: parameters, objective functions, constraints and decision variables.

1) Parameters

The required parameters for the scheduling process can be categorised into two types: system structure parameters and jobs parameters.

The parameters of system structure depend on the configuration of the system. For example, RFACs generally consist of main resources and tools that are used to perform the jobs. These resources are: robots for fetching the assembled parts and placing them at a number of assembly stations (S_1, $S_2, \ldots$ S_n); parts feeder (PF) for supplying parts to the cell; gripper changing station (GC); input conveyor (IC) for supplying the base parts; and output conveyor (OC) for conveying out a final product when assembly processes are completed.

Jobs parameters represent inputs data. In other words, input variables that have fixed values. Processing time, batch size and due date are selected as the common input variables in the scheduling problems. The number of required stations is another suggested variable in this study.

2) Objective functions

The objective function is a value to be minimised or maximised in any optimisation problems. Several objectives functions are used to evaluate the system's performance under different scheduling strategies. Examples of objective function include makespan, system utilisation, lateness/tardiness, production cost. In this study, to evaluate the RFACs' performance under different scheduling policies, three objectives functions, namely makespan, percentage of robots idle time and total tardiness, are to be minimised. The following notations are used to formulate the mathematical expressions of the objectives.

P	Products index ($P = 1, 2, \ldots, i$).
Q	Part index ($Q = 1, \ldots, j$)
R	Robots index ($R = 1, 2, \ldots, k$)
S	Resource index ($S = 1, \ldots, l$)
OP	Assembly operation index ($OP = op_{1i}, op_{2i}, \ldots, op_{mi}$)
T_{mi}	Time of assembly operation *m* of product *i*
$T_{(s\to l)i}$	Time taken by robot to travel between two resources ($s \to l$), to assemble product *i*
T_{ji}	Time of tool change to transfer/assemble component *j* of product *i*
D_i	Due date of product *i*
N_i	Batch size of product *i*
C_i	Completion time of product *i*

The makespan is the maximum completion time of the last job processed by robots. The minimisation of this objective results in an efficient utilisation of system resources. The makespan can be represented as:

$$C_{max} = \max_{1 \le i \le p}(C_i) \qquad \forall R \tag{1}$$

The robots idle time is the waiting time of robots before the start of any actions such as move, tool-change, pick-up and assembly. The percentage of robots idle time can be calculated using the following formula:

$$\%\, I_T = \left(1 - \frac{\sum_{OP=1}^{m} T_{mi} + \sum_{S=1}^{l} T_{(s\to l)i} + \sum_{Q=1}^{j} T_{ji}}{C_{max}}\right) \forall\, i \tag{2}$$

The total tardiness is the sum of the tardiness of all jobs. The minimisation of total tardiness aims to find schedules that satisfy the customers' due dates. Total tardiness can be represented as:

$$TD = \sum_{p=1}^{i} [C_i - D_i\,, 0] \tag{3}$$

3) Constraints

Constraints affect the feasibility of a schedule. To generate a reliable solution to practical problems, a set of constraints must be satisfied. In this research, the RFACs scheduling problem is subject to three resource constraints. First, to fetch and assemble, the hand of each robot should be equipped with the right tool; however, a specific tool may be not available for the two robots concurrently, due to the restricted number of available tools. These are tooling resource constraints. Second, robot arms cannot move from one place to another directly. The reason for this is to avoid collisions with the other robot arms. This is achieved by assigning control points in the cell. Control points $\{C_1, C_2, \ldots, C_4\}$ are set to simplify path planning and avoid collisions. For example, R1 cannot move from S5 to S6 directly; to move from S5 to S6, R1 should move via control point C2. These requirements are called robot move constraints. Third, to prevent collisions between robots in a shared area, more than one robot cannot access the same resource simultaneously. For instance, just one robot R1 or R2 can access transfer table (S4) or tool magazine (S5) or assembly station (S6) or the conveyors IN and OUT. These requirements are named robot access constraints.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

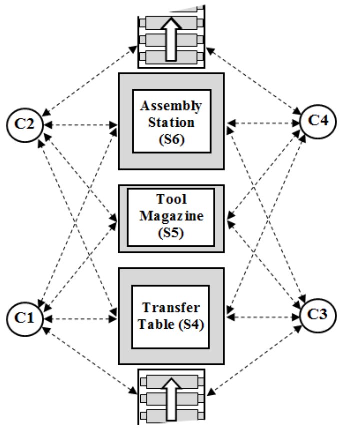

Fig. 2. Robot move and access constraints

4) Decision Variable

In this research, the decision variable represents the job priority, illustrating the priority status of a product to be selected for the next assembly operation in RFACs. The section (scheduling module) will explain how to determine the job priority using scheduling rules.

B. Scheduling module

In scheduling RFACs, when a robot becomes free and more than one job is waiting for processing, the jobs will be scheduled, from the highest priority to the lowest priority. This can be done using scheduling rules. Scheduling rules are used to generate the sequence of job flow to the system. In the proposed methodology, a new scheduling rule is developed for scheduling RFACs. This rule named a fuzzy sequencing rule (FSR) is constructed by combining all the input variables using fuzzy logic.

In this study, the job sequence determination is carried out by evaluating the normalisation of processing time μ_T^i, batch size μ_N^i, due date μ_D^i and number of required stations μ_S^i. The normalisation of the four inputs can be easily defined. For example the (μ_T^i) The normalisation of the total processing time of product *i* is defined as the ratio of the difference between the total processing time of product *i*, and minimum the total processing time to the difference between the maximum and minimum total processing time of the same product, as show in formula 4.

$$\mu_T^i = \frac{[(T_i) - \text{Min}\ (T_i)]}{[\text{Max}\ (T_i) - \text{Min}\ (T_i)]}\,, \qquad 0 \le \mu_T^i \le 1 \qquad (4)$$

The overall normalisations are used to determine which product must be assembled first. The products with low μ_T^i, early μ_D^i, low μ_N^i and high μ_S^i will take earlier position in the job sequence. The sequence of the products is determined by ordering the priority of the jobs from high product priority to small product priority. The job priority can be calculated using fuzzy logic.

Fuzzy logic system (FLS) consists of four main components: knowledge base, fuzzification, inference engine and defuzzification, as shown in Fig. 3.

The most important component in a FLS is the knowledge base. This component stores both the membership functions and the IF-THEN rules base provided by experts. Three steps: linguistic variables, membership functions and fuzzy rule are prepared to establish a knowledge base. The next sub section will describe the previous three steps.

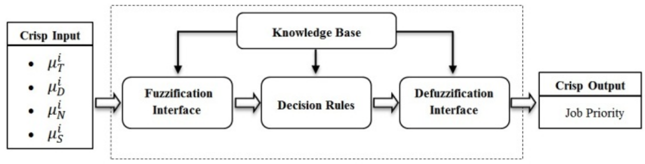

Fig. 3. Fuzzy logic system configuration for job selection

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 I.

TABLE I
DEFINITIONS OF FUZZY VARIABLES

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

2) Construction of membership functions

In this study, the input/output variables are constructed from different types of membership functions. Both processing time and batch size are constructed as triangular shape; number of required station is built from trapezoidal shape. While, due date and job priority are constructed from triangular and trapezoidal. Fig. 4 is a one example of the membership function for processing time.

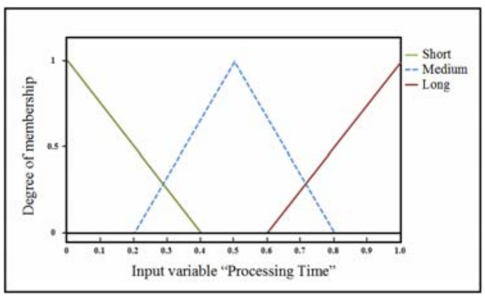

Fig. 4. Membership function of processing time

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3) Definition of fuzzy rules

Fuzzy rules are structured to control the output variable. These rules can be provide by experts or may be extracted from numerical data. Since the variables of processing time, batch size, due date have three states each and number of required station has two states, the total number of fuzzy rules is fifty four (3×3×3×2 = 54).

The generic form of a fuzzy rule can be stated in the form as: 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 fuzzy variable. The fuzzy rules derived are shown as in the example: IF (*Processing Time* is S) and (*Due Date* is S) and (*Batch Size* is S) and (*Number of Required Station* is H) THEN (*Priority* is VH).

C. Simulation module

Once the scheduling parameters, objective functions, constraints and decision variables are determined, the simulation module is defined and constructed. The simulation module is the main part of the proposed methodology that enables the implementation of simulation model of RFACs, to evaluate the system performance under different scheduling strategies. In this study, the simulation software named is SIMPROCESS used to build and simulates the assembling processes. The process of simulation RFACs is achieved through main four stages, using SIMPROCESS software. These stages are shown in Fig. 5.

Fig. 5. Simulation process in SIMPROCESS.

Constructing a computer model of the RFACs is divided into three steps; define the model, construct software model and make a pilot run. Step one is based on the conceptual model that represents all the information related to the system, such as the components of the system and its layout, inputs required, assumption, and output generated. Step two is to construct the proposed model as a computer program; this can be done via encoding the mathematical and logical information of the system in a form that can be achieved by the computer software. After the model is defined and constructed, then a pilot run is done in step three, in order to be sure that the model is working as required, and detect any errors before beginning the simulation process. In SIMPROCESS, verification is an essential tool for checking the validity of the constructed model. Animation is another powerful tool for verifying the constructed model and visualising the process in motion. The second stage in Fig. 5 is running a model to generate the desired solutions. The model is run based on different numbers of experiments. In this research, the design experiments are determined by the output of the scheduling module, which represents the sequence of job flow to the system. The third stage is computing the performance measures. In this research, five performance measures are used. The last stage in Fig. 5 is evaluating alternative scenarios, in order to evaluate the RFACs performance under different scheduling strategies.

III. Case Study

The RFACs studied in this chapter consist of the following three main components, depicted in Fig. 6. First, robots (R1 and R2) fetch the required parts and place them at assembly stations (S1, S2 and S3) where the parts are assembled. Second, part feeder (PF) supplies parts to the cell. Third, input and output conveyors (IC & OC) supply the base parts and carry out the final products.

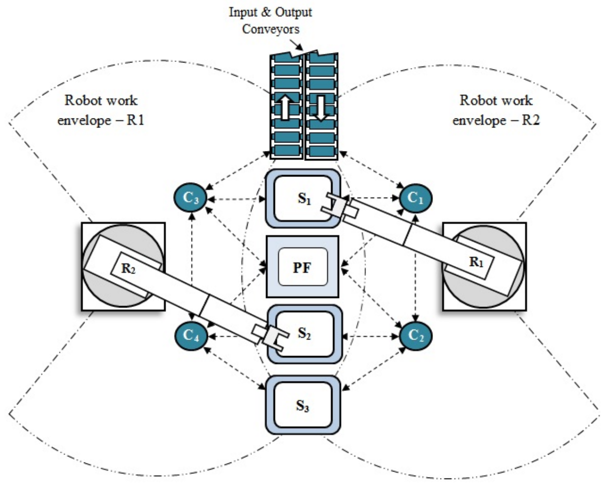

Fig. 6. A robotic flexible assembly cell.

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 $\{C_1, C_2, \ldots, C_4\}$ are set to simplify path planning and avoid collision collisions between robots in the shared area. Table II shows the robot paths and their required time to move between two positions in the cell.

TABLE II
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
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_1←→C_2 C_3←→C_4	0.5

The RFACs described above are assumed to assemble *n* product types. Each product is considered as an independent job. In this model, six products are taken as an example. Table III 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.

TABLE III
ASSEMBLY OPERATIONS REQUIREMENTS

Description	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

In order to simulate RFACs, three customer orders are assumed and labelled as order 1, #2 and #3, shown in Table IV. 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 also given in this Table.

TABLE IV
ORDERS FOR PRODUCT TYPES WITH DIFFERENT PRODUCTION VOLUME

Product Type	*Orders #1*		*Orders #2*		*Orders #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
Prod. volume	28		20		22	

IV. SIMULATION EXPERIMENTAL DESIGN AND RESULTS

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 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 are generated different sequence 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. Three common performance measures, namely makespan, percentage of robots idle time and total tardiness, are used to determine the performance of the RFACs. As mentioned earlier, three customer orders, as shown in Table IV, are assumed in order to simulate RFACs.

The comparisons of all scheduling rules with respect to the five performance measures are shown in Fig. 7 to 10. The following paragraphs 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. 7 shows the makespan results of scheduling rules for different customer orders. From this Fig. 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.

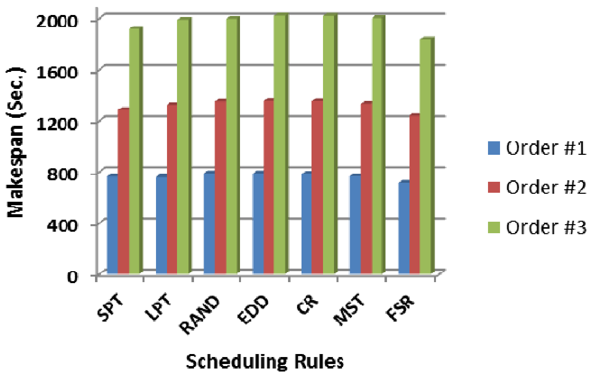

Fig. 7. Makespan

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.

Fig. 8 shows the percentage of idle time of scheduling rules on the three orders. In this Fig., 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

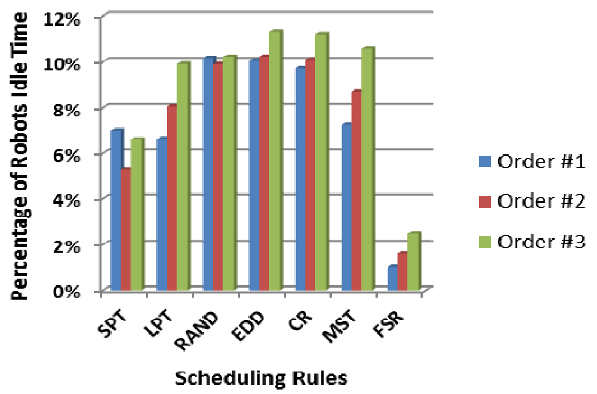

Fig. 8. Percentage of robots idle time

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.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The overall total tardiness of the scheduling rules on the three different orders is depicted in Fig. 9. In this Figure, EDD appear 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

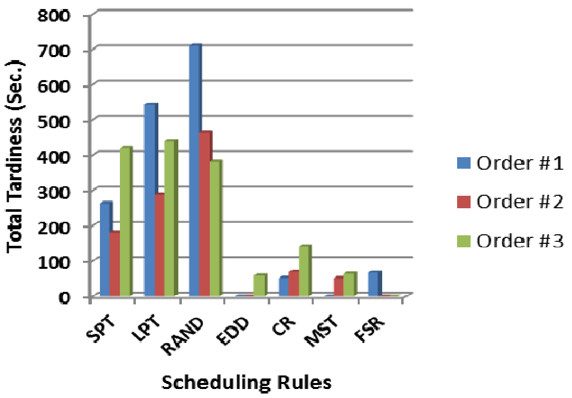

Fig. 9. Total tardiness

V. Conclusion

In this study, a new methodology based on fuzzy logic and simulation is developed for scheduling RFACs, to minimise multi objective functions. The proposed methodology is constructed from three modules: pre-processing, scheduling and simulation module. A new rule, named fuzzy sequencing rule (FSR), is proposed in scheduling module. Several experiments were performed via simulation module to investigate the effectiveness of the FSR.

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 total tardiness, on the other hand performs it poorly in minimising time based measures. Also, the FSR rule proves very effective in minimising the due date based measures.

From the above results and discussion, it can be concluded that FSR are 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 and number of required stations. Additionally, the simulation results indicate that use of common scheduling rules does not guarantee the obtaining of satisfactory results regarding all system performance criteria.

Acknowledgment

The authors gratefully acknowledge the comments and suggestions of the anonymous referees that considerably improved the presentation and clarification of the paper.

References

[1] S. Manivannan, "Robotic collision avoidance in a flexible assembly cell using a dynamic knowledge base," *IEEE transactions on systems, man, and cybernetics*, vol. 23, pp. 766-782, 1993.
[2] R. M. Marian, A. Kargas, L. H. S. Luong, and K. Abhary, "A framework to planning roboticflexible assembly cells," *presented at the 32nd International Conference on Computers and Industrial Engineering*, Limerick, Ireland, 2003.
[3] T. Sawik, Production planning and scheduling in flexible assembly systems. poland: springer, 1999.
[4] S. B. Mohamed, D. J. Petty, D. K. Harrison, and R. Rigby, "A cell management system to support robotic assembly," *The International Journal of Advanced Manufacturing Technology*, vol. 19, pp. 598-604, 2001.
[5] H. Makino, "The comparison between robotic assembly line and robotic assembly cell," *presented at the 10th International Conference on Assembly Automation*, Canazawa, Japan, 1989.
[6] E. K. Xidias, P. T. Zacharia, and N. A. Aspragathos, "Time optimal task scheduling for two-robotic manipulators operating in a three-dimensional environments," *Journal of Systems and Control Engineering*, vol. 224, pp. 845-855, 2010.
[7] S. Y. Nof and J. Chen, "Assembly and disassembly: an overview and framework for cooperation requirement planning with conflict resolution," *Journal of Intelligent and Robotic Systems* vol. 37, pp. 307–320, 2003.
[8] J. K. Lee, T. E.Lee, "Automata-based supervisory control logic design for a multi-robot assembly cell", *International Journal Computer Integrated Manufacturing* 15, pp. 319-334, 2002.
[9] S. Y. Nof and Z. Drezner, "The multiple-robot assembly plan problem", *Journal of Intelligent and Robotic Systems* 7, pp. 57-71. 1993.
[10] H. C. Lin, P. J. Egbelu and C. T. Wu, "A two-robot printed circuit board assembly system", *International Journal of Computer Integrated Manufacturing* 8, pp.21-31. 1995.
[11] P. M. Pelagagge, G. Cardarelli and M. Palumbo, "Design criteria for cooperating robots assembly cells", *Journal of Manufacturing Systems* 14, pp. 219-229. 1995.
[12] T. Sawik , "Integer programming models for the design and balancing of flexible assembly systems", *Mathematical and Computer Modelling* 21, pp. 1-12. 1995.
[13] K.Jiang, L. D. Seneviratne, and S. W. E. Earles, "Scheduling and compression for a multiple robot assembly workcell", *Production Planning & Control* 9, pp. 143-154. 1998.
[14] Rabinowitz, G., Mehrez, A., Samaddar, S., "A scheduling model for multi-robot assembly cells", *International Journal of Flexible Manufacturing Systems* 3, pp. 149-180. 1991.
[15] P. R. Glibert, D. Coupez, Y. M. Peng and A. Delchambre, "Scheduling of a multi-robot assembly cell", *Computer Integrated Manufacturing Systems* 3, pp. 236-245. 1990.
[16] H. Hsu and L. C. Fu, "Fully automated robotic assembly cell: scheduling and simulation", *IEEE International Conference on Robotics and Automation*, pp. 208-214. 1995.
[17] D. Barral, J.-P. Perrin and E. Dombre, Flexible agent-based robotic assembly cell, New Mexico, 1997.
[18] H. Van Brussel, F. Cottrez and P.Valckenaers, "SESFAC: A scheduling expert system for flexible assembly cell", *Annals of The CIRP* 39, pp. 19-23. 1990.
[19] C. Del Valle and E. F.Camacho, "Automatic assembly task assignment for a multirobot environment", *Control engineering practice* 4, pp. 915-921. 1996.
[20] K. Abd, K. Abhary, and R. Marian, "A scheduling framework for robotic flexible assembly cells," *AIJSTPME-Asian International Journal of Science and Technology in Production and Manufacturing Engineering*, vol. 4, pp. 30-37, 2011.
[21] K. Abd, K. Abhary, and R. Marian, "Scheduling and performance evaluation of robotic flexible assembly cells under different dispatching rules," *Advances in Mechanical Engineering*, vol. 1, 2011.
[22] K. Abd, K. Abhary, and R. Marian, "An MCDM Approach to Selection Scheduling Rule in Robotic Flexible Assembly Cells," *presented at the International Conference on Mechanical, Aeronautical and Manufacturing Engineering*, Venice, Italy, 2011.