Intelligent Modeling of Scheduling Robotic Flexible Assembly Cells Using Fuzzy Logic

Khalid Abd, Kazem Abhary, Rome Marian
School of Advanced Manufacturing and Mechanical Engineering
University of South Australia
Mawson Lakes, SA 5095, South Australia
AUSTRALIA
Abdkk001@mymail.unisa.edu.au,{Kazem.Abhary & Romeo.Marian}@unisa.edu.au

Abstract: - This paper is concerned with scheduling robotic flexible assembly cells (RFACs) using fuzzy logic (FL) technique. A new scheduling rule is developed and evaluated called fuzzy sequencing rule (FSR). A simulation program is used to examine the performance of the existing scheduling rules and the proposed rule with respect to multiple performance measures. Four performance measures considered simultaneously are: maximum of tardiness, mean tardiness, percentage of tardy products, and percentage of improvement. A case study is presented to prove the efficiency of the proposed rule (FSR). The simulation results indicate that FSR gives superior performance compared with existing scheduling rules.

Key-Words: - Assembly cells, Fuzzy Sequencing Rules, Robotics

1 Introduction

Today's enterprises are forced to develop systems flexible to manufacture products and responding to unpredictable demands with minimal reconfiguration. One class of such systems are Robotic Flexible Assembly Cells (RFACs) [1, 2]. The potential benefits of RFACs are the flexibility to assemble a variety of products, as well as the ease to reconfigure[3]. Nevertheless, two robots (or more) operating simultaneously in the same work environment require a complex scheduling policy to prevent collisions between robots and other equipments in the cell [4]. Scheduling RFACs requires finding a way to determine how to use cell resources in an optimal manner to assemble multi-products. Few studies have been devoted to scheduling RFACs. These studies can be categorised into three approaches. First, the studies which applied heuristic approaches to solve scheduling problems such as Lee and Lee [5], Nof and Drezner [6], Lin et al. [7], Pelagagge et al. [8], Sawik [9], Jiang et al.[10] and Rabinowitz et al.[11]. Second, the studies which investigated simulation as an approach to scheduling RFACs, for instance, Glibert et al.[12], Hsu and Fu [13] and Basran et al. [14]. Third, only two studies, Brussel et al.[15] and Dell Valle and Camacho [16], who 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 [17], scheduling RFACs for concurrent assembly of multi-products has been proposed using common scheduling rules.

Scheduling rules are used for preparing the sequence of jobs in job shop. The rules are employed to improve the system performance such as minimise completion time, minimise tardiness or maximise throughput [18]. There are several scheduling rules. The common rules are listed below.

1. SPT (short processing time): select job with minimum processing time first.
2. LPT (long processing time): select job with maximum processing time first.
3. EDD (earlier due date): jobs are sequenced according to their due dates.
4. RAND (random): jobs are sequenced randomly.
5. CR (critical ratio): select job with minimum critical ratio first.
6. MST (minimise slack time): jobs are sequenced according to urgency of a job by its slack time.

The drawback of common rules is satisfied only for on one objective function. For example EDD, and CR are efficient at delivering on due date, but may leads to decrease throughput unlike SPT and LPT.

The aim of this paper is to propose a new scheduling rule based on fuzzy logic for scheduling RFACs in a multi-product assembly environment, and then validate the performance of suggested rule using simulation program.

2 Fuzzy logic

Fuzzy logic (FL) was introduced first by Zadeh in 1965 [19]. FL is a nonlinear mapping of an input data vector into a scalar output. Generally, a fuzzy logic system (FLS) consists of four main components [20] knowledge base, fuzzification, inference engine and defuzzification and as depicted in Figure 1.

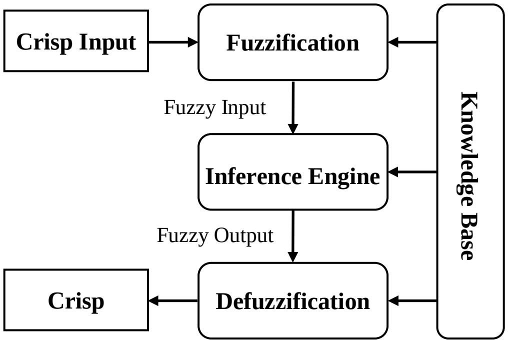

Fig.1: Fuzzy logic system configuration

2.1 Knowledge Base

The important component in a FLS is the knowledge base. To establish a knowledge base, three steps are prepared.

2.1.1 Linguistic Variables

A linguistic variable is the procedure to describe variables in terms of words instead of the values. In general, a linguistic variable is decomposed into a set of terms called linguistic terms, for example, 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.

2.1.2 Membership functions

A membership function (MF) embodies a fuzzy set Ã graphically. Membership functions are considered the core of fuzzication and defuzzication components in a FLS. Figure 2 shows the most well-known of these shapes, i.e. triangular and trapezoidal [20]. These membership functions are employed in this study.

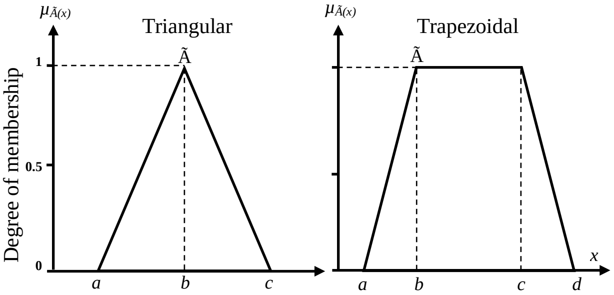

Fig. 2: Two examples of fuzzy numbers, triangular and trapezoidal

2.1.3 Fuzzy rules

A fuzzy rule is structured to control the output variable. These rules can be provide by experts or may be extracted from numerical data. A fuzzy rule has two parts: the antecedent and the consequent as follow IF <antecedent> THEN <consequent>.

2.2 Fuzzification

The fuzzification is the first component in a FLS which represents the process of converting the real world variables (crisp input data) into linguistic variables (fuzzy values). This process can be done using the membership functions of input variables.

2.3 Inference engine

The inference engine maps from fuzzy input to fuzzy output using IF-THEN type fuzzy rules. These rules reflect a human reasoning mechanism.

2.4 Defuzzification

The defuzzification translates the fuzzy output into a crisp value. The defuzzification process can be achieved using the membership functions of output variable. Several methods have been proposed for defuzzification process. One of common method for this process is Center of Gravity (COG) [20].

3 Proposed model

The proposed fuzzy model is developed to combine all input fuzzy variables in one Scheduling rule. In this model, four major steps are performed to generate FSR.

3.1 Definition of fuzzy variables

The input fuzzy variables include processing time, batch size, due date and number of required station; the output fuzzy variable represents a job priority. In this study, each product is considered as independent job.

Processing time: this input variable represents the summation time of all required tasks needed to complete the product. These tasks are assembly, part move, pickup and release.

Batch size: generally, any flexible system can process different jobs. Each job is processed in a different amount called a batch size, which is depending on the customer requirements.

Due date: this input variable denotes the deadline of production of each job. In other words, the job must be completed prior to the required time by the customer; otherwise the company might be facing a penalty for late completion.

Number of required station: the last input fuzzy variable gives high priority to the product which is requiring more number of stations.

Product priority: this variable represents the fuzzy output of the proposed model, illustrating the priority status of a product to be selected for the next assembly operation in RFACs.

3.2 Defining the linguistic variables

The next step is to define the linguistic input/output variables. Let us suppose that due date, batch size and processing time have three linguistic variables, number of required station has two linguistic variables, while the output variable product priority has seven linguistic variables, as shown in Table1.

Table 1: Definitions of fuzzy variable

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

3.3 Construction of membership functions

In this study, the input/output variables are constructed from different types of membership functions. For example, both processing time and batch size are constructed as triangular shape; number of required station is built from trapezoidal shape. While, due date and product priority are constructed from triangular and trapezoidal, as depicted in Figure 3.

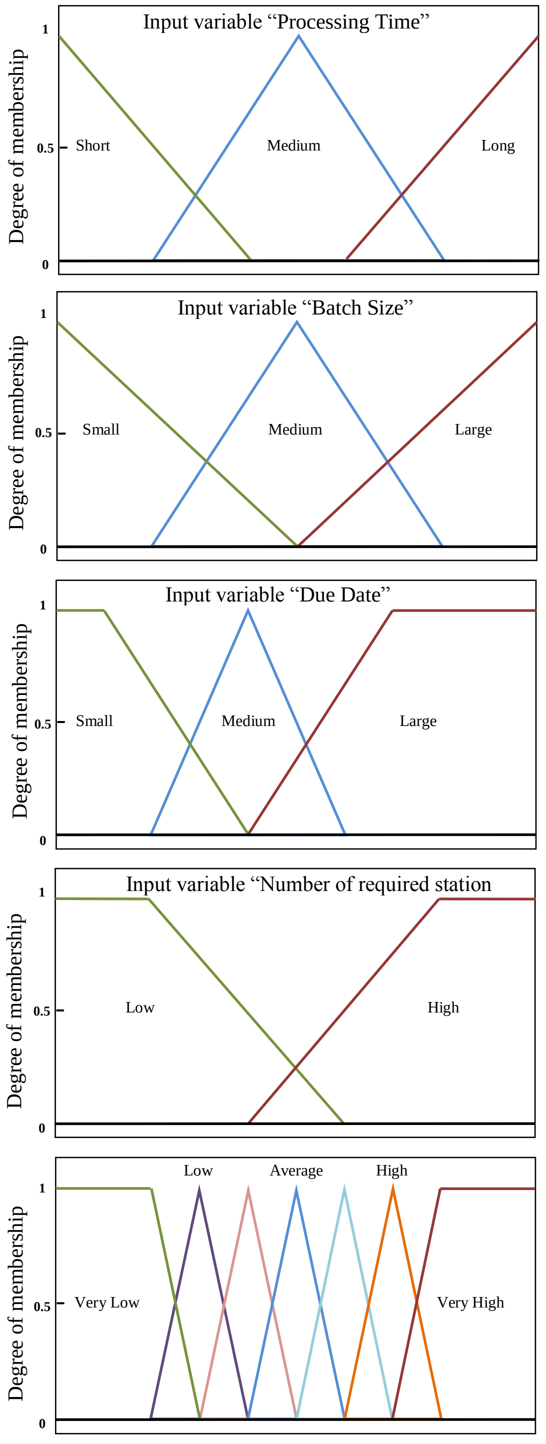

Fig. 3: Membership functions for fuzzy input/output variables

3.4 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).

4 Performance Measures

Four performance measures are suggested. The following notations are used to formulate the mathematical expressions of performance measures.

B	Batch index (h = 1, . . . ,n)
P	Product index (i= 1, . . .,p)
D_i	Due date of product i
N_i	Batch size of product i
C_i	Completion time of product i
C_h	Completion time for the whole batch h
$\overline{TD}_i$	Tardiness of product i
N_{TD}	Number of tardy products
$C_{h,ESR}$	C_h of Existing Sequinning Rule
$C_{h,FSR}$	C_h of Fuzzy Sequinning Rule

1. Maximum tardy products (MaxTD)

$$\mathrm{MaxTD}\left(\max_{1\le i\le p}[C_i - D_i]\right)\forall h \qquad (1)$$

2. Mean tardiness ($\overline{TD}$)

$$\overline{TD} = \frac{1}{\sum_h^n \sum_i^p N_i}\sum_i^p [C_i - D_i]\,\forall h \qquad (2)$$

3. Percentage of tardy products (% T)

$$\%\,T = \frac{N_{TD}}{\sum_{i=1}^{p} N_i}\times 100 \qquad (3)$$

4. Percentage of improvement (% I)

$$\%\,I = \left(\frac{C_{h,ESR} - C_{h,FSR}}{C_{h,ESR}}\right)\times 100 \qquad (4)$$

5 Case study

The RFACs studied in this paper consists of the following physical resources, depicted in Figure 4.

1. Robots (R1 and R2) for fetching the assembled parts and placing them at assembly stations (S1, S2 and S3) where components are assembled.
2. Part feeder (PF) supply parts to the cell
3. An input and output conveyors (IC& OC) supply base part and conveying out a final product.

Six different cell phone types are considered to be assembled in the RFACs. The required station along with assembly operations time for each product type and is shown in Table2.

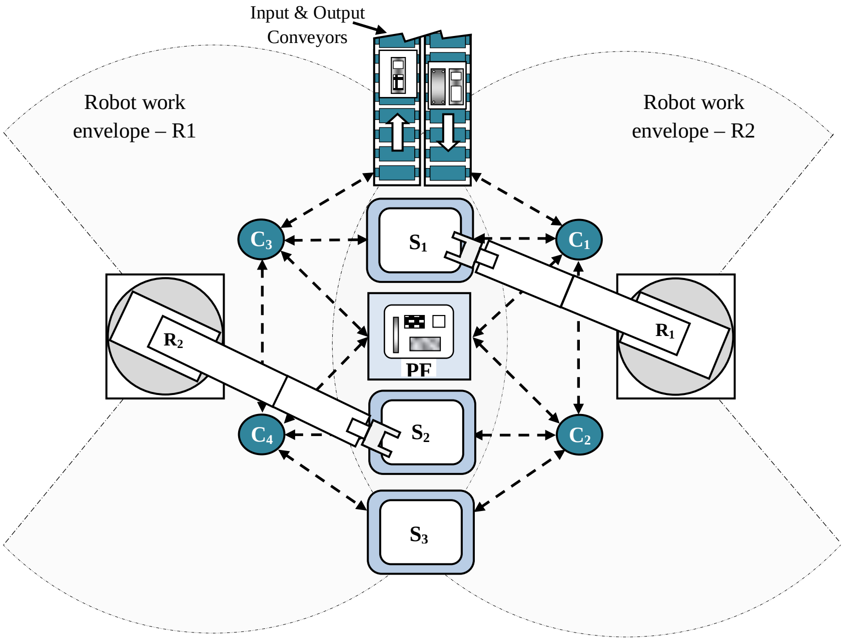

Fig. 4: A robotic flexible assembly cell

Table 2 : Assembly operations requirements

Description	Assembly Station	Time of Assembly operations					
		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 move time (s)		23	17	17	23	17	17
Robot gripper pickup & release time (s)		6	4	4	6	4	4
Total Processing time (s) for each product		60	47	49	68	43	50

In order to simulate RFACs, three customer' orders are assumed and labelled as order #1, 2 and 3, shown in Table3. Order#1 and #3 consist of six types of cell phone, and order#2 is composed of only five types of products. Batch size (BS) and due date (DD) for each product types are given in Table3.

Table 3: Orders data

No	*Orders #1*		*Orders #2*		*Orders #3*	
	BS	**DD(s)**	**BS**	**DD(s)**	**BS**	**DD(s)**
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

6 Result and discussion

The RFACs was simulated with different experiments, using SIMPROCESS simulation software [21]. Each experiment is performed with different scheduling rule. Experiments numbered 1 to 6 were run with existing scheduling rules; Experiment 7 was run using developed rule (FSR).

The simulation results indicate that use of common rules does not assured to obtain the acceptable results regarding all system performance criteria. For example, SPT and LPT rules obtain unacceptable mean tardiness and maximum tardiness of products, while, SPT achieves acceptable percentage of tardy products. Additionally, EDD rule gives best mean tardiness and maximum tardiness of products, and a good percentage of tardy products. On the contrary, EDD performs poorly performance regarding completion time.

From these results shown in the Figure 5, it can be concluded that FSR are generally better than all other existing rules. The first aspect to discuss is the performance of FSR with respect to completion time criterion. FSR has the best performance, FSR is best by 9% from EDD and CR, and more than 7% of the time compared with MST, RAND and LPT, and 4.2% from SPT, as shown in figure 5-d. For the mean tardiness and maximum tardiness of products criterion, FSR and EDD rule have the best performance followed by MST and CR, while RAND, SPT and LPT have the worst performance, as given in figure 5-a, b. In terms of percentage of tardy products criterion, FSR showing the best performance (5.88%) followed EDD rule (11.76%), SPT performs acceptable performance (17.65%), and LPT has the worst performance (52.94%), as shown in figure 5-c.

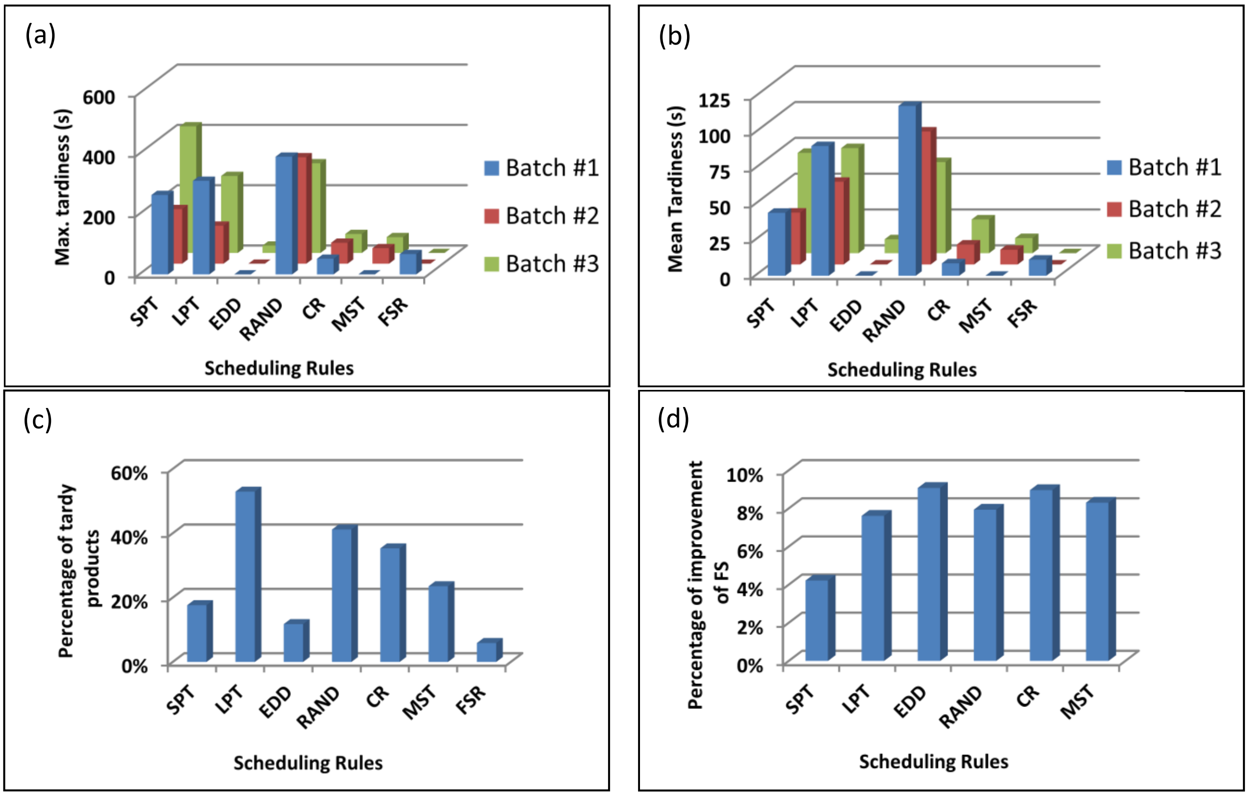

Figure 5: simulation results for common scheduling and FSR

7 Conclusions

In this paper, the new scheduling rule based on fuzzy logic is presented for scheduling RFACs in a multi-product assembly environment. The proposed rule named fuzzy sequencing rule (FSR) is formulated from four fuzzy input variables: processing time, due date, batch size and number of required stations. The product priority is the fuzzy output variable, illustrating the priority status of a product to be selected for the next assembly operation in a cell. To examine the effectiveness of the FSR, several experiments were performed via simulation. Some of these experiments are implemented by using simple scheduling rules such as SPT, LPT and EDD. The other experiments are applied via combination of scheduling rules CR and MST. The simulation results of these experiments demonstrate that the performance of FSR is more efficient compared of common.

References

[1] R. M. Marian, A. Kargas, L. H. S. Luong, and K. Abhary, "A framework to planning robotic flexible assembly cells," in *32nd International Conference on Computers and Industrial Engineering*, Ireland, 2003, pp. 607-615.

[2] K. Abd, K. Abhary, and R. Marian, "A scheduling framework for robotic flexible assembly cells," in *the 10th Global Congress on Manufacturing and Management*, Bangkok, Thailand 2010, pp. 111-116.

[3] S. B. Mohamed, D. J. Petty, and R. Rigby, "A cell management system to support robotic assembly," *The International Journal of Advanced Manufacturing Technology*, vol. 19, pp. 598-604, 2001.

[4] 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, 2003.

[5] J. K. Lee and T. E. Lee, "Automata-based supervisory control logic design for a multi-robot assembly cell" *International Journal Computer Integrated Manufacturing*, vol. 15, pp. 319-334, 2002.

[6] S. Y. Nof and Z. Drezner, "The multiple-robot assembly plan problem," *Journal of Intelligent and Robotic Systems* vol. 7, pp. 57-71, 1993.

[7] H. C. Lin, P. J. Egbelu, and C. T. Wu, "A two-robot printed circuit board assembly system," *International Journal of Computer Integrated Manufacturing*, vol. 8, 1995.

[8] P. M. Pelagagge, and G. Cardarelli, "Design criteria for cooperating robots assembly cells," *Journal of Manufacturing Systems*, vol. 14, pp. 219-229, 1995.

[9] T. Sawik, "Integer programming models for the design and balancing of flexible assembly systems," *Mathematical and Computer Modelling* vol. 21, pp. 1-12, 1995.

[10] K. Jiang, L. D. Seneviratne, and S. W. E. Earles, "Scheduling and compression for a multiple robot assembly workcell," *production Planning & Control*, vol. 9, pp. 143-154, 1998.

[11] G. Rabinowitz, and A. Mehrez, "A scheduling model for multi-robot assembly cells," *International Journal of Flexible Manufacturing Systems* vol. 3, pp. 149-180 1991.

[12] P. R. Glibert, D. Coupez, and Y. M. Peng, "Scheduling of a multi-robot assembly cell," *Computer Integrated Manufacturing Systems,* vol. 3, pp. 236-245, 1990.

[13] H. Hsu and L. C. Fu, "Fully automated robotic assembly cell: scheduling and simulation," in *IEEE International Conference on Robotics and Automation* National Taiwan University, 1995, pp. 208-214.

[14] D. Barral, J.-P. Perrin, and E. Dombre, "Flexible agent-based robotic assembly cell," New Mexico, 1997.

[15] H. Van Brussel, F. Cottrez, and P. Valckenaers, "SESFAC: A scheduling expert system for flexible assembly cell," *Annals of The CIRP*, vol. 39, pp. 19-23, 1990.

[16] C. Del Valle and E. F. Camacho, "Automatic assembly task assignment for a multirobot environment," *Control engineering practice,* vol. 4, pp. 915-921, 1996.

[17] K. Abd, K. Abhary, and R. Marian, "Scheduling and performance evaluation of robotic flexible assembly cells under different dispatching rules," *International Conference on Mechanical, Industrial, and Manufacturing Engineering*, Melbourne, Australia, 2011, pp. 192-197.

[18] Y. B. Canbolat, E. Gundogar "Fuzzy priority rule for job shop", *scheduling*, *Journal of Intelligent Manufacturing,* 15, 527- 533, 2004.

[19] L. A. Zadeh, Fuzzy sets. Information and Control, 8, pp. 338–353, 1965.

[20] J. M. Mendel, Fuzzy logic systems for engineering: A tutorial. Proceedings of the IEEE, 83, 345–377, 1992.

[21] S. Swegles, Business process modeling with SIMPROCESS. *Winter Simulation Conference,* Piscataway, NJ, 606-610, 1997.