

INTELLIGENT MODEL OF SCHEDULING RFACS - PART I: METHODOLOGY AND STRATEGY

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

Abstract: *Production scheduling of advanced manufacturing systems has attracted significant attention of both researchers and industrial practitioners in recent years. Due to the complexity in these systems, the generation of production schedules requires an intelligent technique. Many artificial intelligence techniques such as fuzzy logic, genetic algorithms and neural networks have been successfully applied to the scheduling of advanced manufacturing systems. One such system is robotic flexible assembly cells (RFACs). Few studies have been done on the problem of scheduling RFACs. The major limitation is that these studies are limited to the assembly of only one product type. The objective of this chapter is to propose a new intelligent model of scheduling RFACs in a multi-product assembly environment, using fuzzy logic.*

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



Authors' data: MEng, Dipl.-Ing. **Abd, K**[halid]; 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[halid]; Abhary, K[azem] & Marian, R[omeo] (2013) Intelligent Model of Scheduling Rfacs - Part I: Methodology and Strategy, Chapter 43 in DAAAM International Scientific Book 2013, pp. 719-736, 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.2013.43

1. 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 (Manivannan 1993, Marian et al. 2003, Sawik 1999). The design of RFACs with multi robots leads to increased productivity in a shorter cycle time and with lower production costs (Xidias et al. 2010). However, there are certain difficulties that have arisen with this design concept. For example, more than one robot operating simultaneously in the same work environment requires a complex control system to prevent collisions between robots (Nof and Chen 2003), and also to prevent deadlock problems (Lee and Lee 2002). Moreover, industrial robots must be employed as effectively as possible due to high cost of the robots (Xidias et al.2010). To overcome the above difficulties, efficient scheduling of RFACs is required.

Few studies have been done on the problem of scheduling RFACs. These studies may be categorised into three groups (For more detail, see the paper Abd et al. 2011a). The first group applied heuristic methods, while the second group investigated simulation as an approach to scheduling RFACs, and the third group implemented expert systems to solve scheduling problems in RFACs. The major limitation of the previous studies of scheduling RFACs is that they concentrated on assembling only one type of product at a time. The objective of this chapter is to propose a new intelligent model of scheduling RFACs in a multi-product assembly environment, using fuzzy logic.

This chapter is organised as follows. The next section describes an overview of recent studies and how they have been applied fuzzy logic to solving the scheduling problems. In Section 3, a new methodology for the scheduling of RFACs is developed. Finally, the conclusions and areas for further work are presented in Section 4.

2. A Review of the Literature

In manufacturing systems, since the scheduling problems are NP-hard, an efficient approach is required to get best results (Buil et al., 2010, Sridhar et al., 2010). Recently, a fuzzy logic approach has been widely applied to the scheduling problems for both conventional and flexible manufacturing systems (Subramaniam et al., 2000, Vidyarthi and Tiwari, 2001, Domingos and Politano, 2003, Bilkay et al., 2004, Canbolat and Gundogar, 2004, Kumar et al., 2004, Srinoi et al., 2006, Restrepo and Balakrishnan, 2008, Srinoi et al., 2008, Mahdavi et al., 2009). In this section, the literature review will provide the necessary key points for the development of a conceptual methodology for the scheduling of RFACs.

Subramaniam *et al.* (2000) developed a scheduling method, named the fuzzy scheduler, based on fuzzy logic which is used in the job shop environment to evaluate several candidate scheduling rules and then select the most convenient rule. In this study, the fuzzy scheduler is applied to different job shop problems, which are representative of real industrial applications. The results indicated that the fuzzy scheduler performs better than other common scheduling rules. The results also showed that the fuzzy scheduler is as easy to apply as the scheduling rules.

Vidyarthi and Tiwari (2001) presented a fuzzy-based methodology for machine loading in a FMS. They considered processing time, batch size and optional operation processing time as three fuzzy input variables. The first step in this methodology was to evaluate the overall contribution of the job variables, using fuzzy membership functions, and then to determine the job sequencing as the output fuzzy variable. The objective was to maximise the throughput and minimize the system unbalance.

Domingos and Politano (2003) proposed a procedure based on fuzzy logic for scheduling a FMS. In this study, processing time, due date and work load were the input fuzzy variables, and the part priority was the output fuzzy variable. Three common objective functions, namely average tardiness, percentage of tardy jobs and average flow time, were to be minimised. They compared their procedure with common scheduling rules via simulation software. The simulation results indicated that the proposed procedure improved the performance of the FMS under multi objectives.

Kumar *et al.* (2004) developed a fuzzy based algorithm to solve the scheduling problems of a FMS. They applied fuzzy membership functions to evaluate the overall contribution of each job type to the objectives according to the attributes, and then determine the job sequencing. They used processing time, batch size and required tool slots as the main attributes. Two objectives functions, maximising of throughput and minimising of system imbalance, were considered in this study. The computational results showed that the developed algorithm gives better solutions than those obtained by heuristic approaches.

Bilkay *et al.* (2004) proposed a fuzzy logic-based decision-making algorithm for generating the sequence of part types to be processed. They generated a sequence based on priority value of the part types, from higher priority to lower priority value. The algorithm assigns priorities for part types according to four input variables: processing time, batch size, due date and the required tool slots. The results showed that fuzzy logic improves the system efficiency and is suited for scheduling problems that have multiple conflict objectives.

Canbolat and Gundogar (2004) applied a fuzzy logic approach to solve a multi criteria scheduling problem for a job shop environment. The suggested approach combined three scheduling rules in a new rule named fuzzy priority rule (FPR). The new rule is compared with other traditional scheduling rules such as SPT, EDD, CR, etc., using a simulation program. The simulation results showed the superiority of the FPR over traditional rules in mean flow time and mean tardiness.

Srinoi *et al.* (2006) developed a new approach based on fuzzy logic to generate a scheduling model for solving the resource allocation problem in flexible manufacturing systems. They defined four fuzzy input variables of the model: processing time, due date, setup time and machine priority; the output variable of the model is the job priority. They conducted several experiments to prove the effectiveness of the developed approach. The experimental results indicated that the fuzzy logic approach is a powerful technique for scheduling problems in FMS, based on multi criteria objectives.

Restrepo and Balakrishnan (2008) proposed fuzzy-based methodology to solve multi-objective scheduling problems for robotic flexible manufacturing cells. The proposed methodology compared two scheduling rules namely SPT and EDD. Simulation results showed that the existing scheduling rules are satisfied only when

considering single objectives such as maximizing the throughput time or minimizing the tardiness cost. The results indicated also that the fuzzy-based methodology is able to give a better performance than SPT and EDD.

Srinoi *et al.* (2008) developed a fuzzy-based mathematical model to deal with scheduling in FMS, based on multi-performance measures. They used processing time, machine priority, due date and setup time as input fuzzy variables, while the job priorities are the output variable. The simulation results pointed out the superiority of the suggested model in most performance measures.

Mahdavi *et al.* (2009) presented a fuzzy approach to solve the scheduling problems of a FMS. They defined four fuzzy input variables: processing time, workload, setup time and travelling time. In this study, the output fuzzy variable was the optimal route selection to satisfy multi-conflicting objectives. They used the MATLAB fuzzy logic toolbox to determine the route selection. The numerical results showed that the presented approach is easily applicable to finding the optimal flexible routing in FMS.

Based on the previous studies, five key points can be extracted:

- Most of the above studies showed that the use of fuzzy logic and simulation tools can be suitable to optimise the scheduling problems for both conventional and flexible manufacturing systems.
- The scheduling problems may be divided into three main sub-problems: part type selection, machine loading and resource allocation. Most of the studies dealt with part type selection problems.
- The majority of the listed studies took into account the processing time; due date and batch size were the main fuzzy criteria.
- Two decision types, parts routing and parts sequencing, can be identified in the above studies. Most of the studies focused on the parts sequencing decision.
- Nearly all the studies reviewed above employed more than two performance measures to evaluate the quality of the schedules.

This chapter attempts to use the key points mentioned above to develop an intelligent methodology for the scheduling of RFACs. Therefore, the proposed methodology will include the following: the technique that will be used is fuzzy logic in combination with a simulation tool; the scheduling problem is product type selection; the fuzzy criteria are processing time, due date and batch size; the decision type is products sequencing; and the scheduling output is evaluated using multi performance measures. The next section will describe the proposed methodology in more detail.

3. Proposed Methodology for the Scheduling of RFACs

The main purpose of this chapter is to develop a methodology that will allow the user to model the scheduling of RFACs in an optimal way. 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.

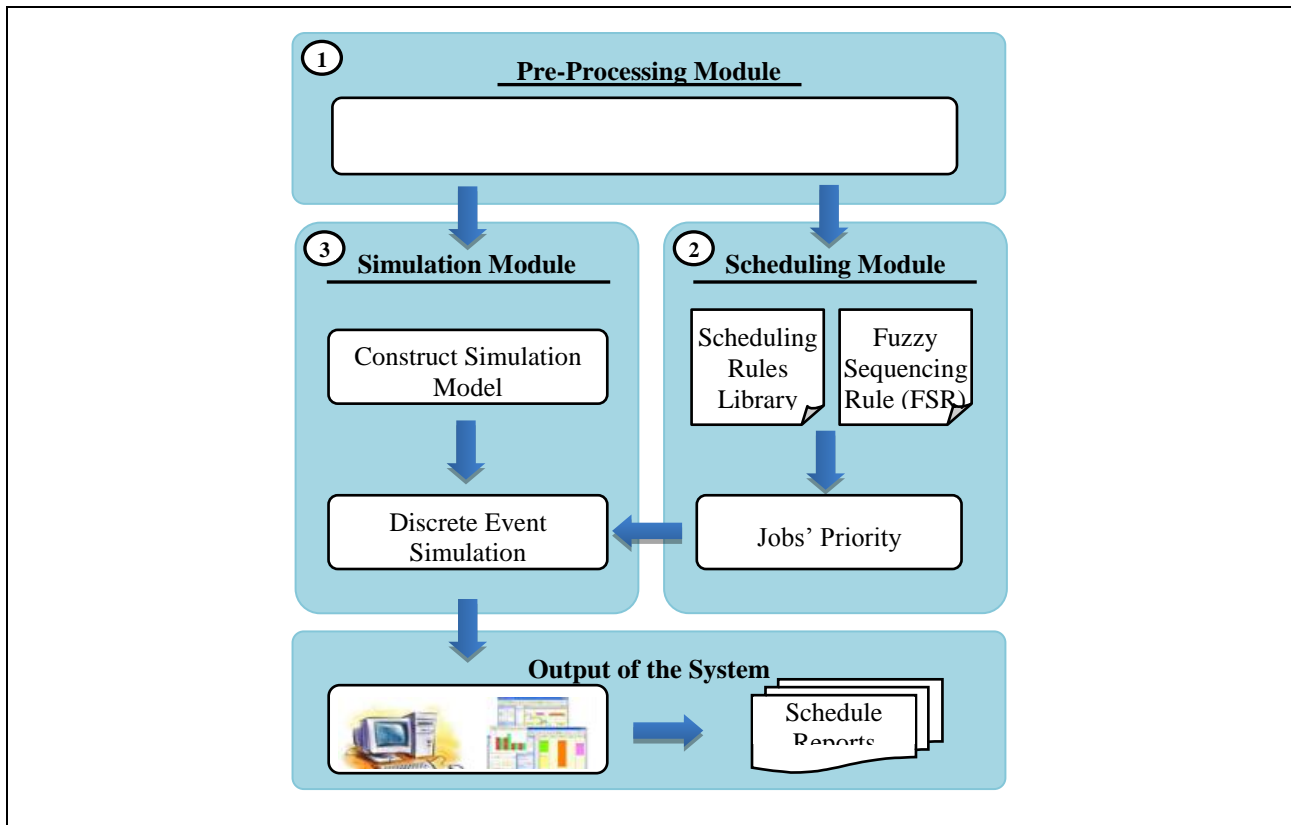


Fig. 1. Architecture of proposed methodology for RFACs scheduling

In this section, the proposed methodology for the scheduling of RFACs is described. This methodology has three major modules: (1) pre-processing module: this module helps to define the components of the scheduling problem model. For example, this module determines the system's inputs/output, identifies the objectives and describes the characteristics of RFACs (2) scheduling module: this module is the core of the proposed methodology, which allows the user to generate the schedule for assembling multi products (3) simulation module: this enables the user to build the RFACs as a computer model, and then simulates the model under different scenarios, depending on the outcome of the scheduling module. The architecture of the proposed methodology is illustrated in Figure 1. The next sections will present these three modules in more detail.

3.1 Pre-processing Module

The aim of the pre-processing module is to describe all the required components of the scheduling problem model in the RFACs. These components are: parameters, decision variables, constraints and objective functions, as shown in Figure 2. The next paragraphs give the necessary information about these components.

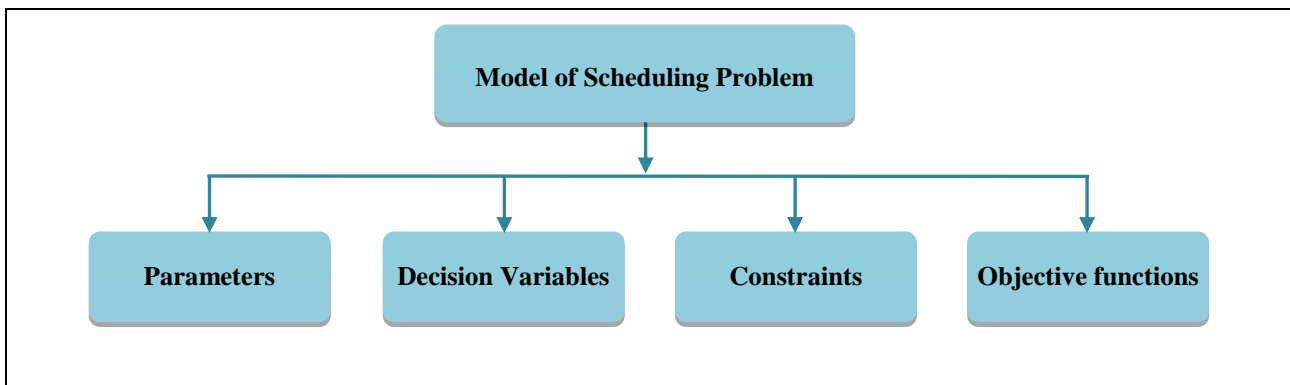


Fig. 2. Model of scheduling problem

3.1.1 Parameters

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

The system structure parameters depend on the configuration of the system. In other words, they reflect the physical characteristics 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 (AS_1, AS_2, \dots, AS_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 (Marian et al., 2003, Abd et al., 2012).

Jobs parameters represent inputs data for a system: in other words, input variables that have fixed values. In this chapter, processing time, batch size and due date are selected as the input variables in the scheduling problems (Bilkay et al., 2004). Also, the number of required stations is suggested as another variable in this research.

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

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

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

Number of required stations: the last input variable gives high priority to the product which requires the greater number of stations.

3.1.2 Decision Variables

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

3.1.3 Constraints

Constraints define 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: tooling resource constraints, robot movement constraints and robot access constraints (Abd et al., 2011a, Abd et al., 2011b).

- 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 simultaneously, due to the restricted number of available tools. These are tooling resource constraints.
- 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 $\{C_1, C_2, \dots, C_4\}$ are set to simplify path planning and avoid collisions. For example, R1 cannot move from S_5 to S_6 directly; to move from S_5 to S_6 , R1 should move via control point C_2 . These requirements are called robot movement constraints, as shown in Figure 3.
- To prevent collisions between robots in a shared area, more than one robot cannot access the same resource at the same time. For instance, just one robot R_1 or R_2 can access transfer table (S_4) or tool magazine (S_5) or assembly station (S_6) or the conveyors IN and OUT. These requirements are named robot access constraints, as shown in Figure 3.

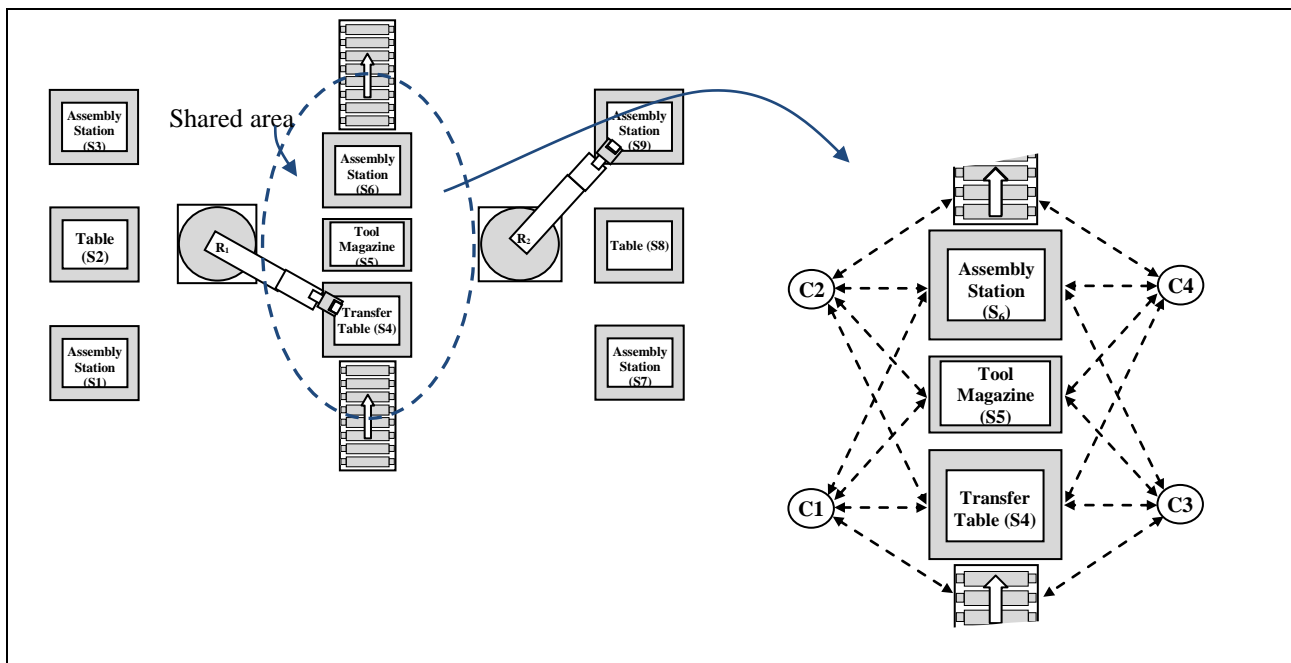


Fig. 3. Robot move and access constraints

3.1.4 Objective functions

The objective function is a value to be minimised or maximised in any optimisation problems. Examples of objective functions include makespan, system utilisation, lateness/tardiness, production cost. In the scheduling area, several objective functions are used to evaluate the system's performance under different

scheduling strategies. (Ramasesh, 1990) categorised the objective functions into four types: time-based objectives, work-in-process objectives, due-date-based objectives and cost-based objectives.

In this research, five objective functions, namely makespan, percentage of robots idle time, total tardiness, maximum tardiness and percentage of tardy jobs, are to be minimized, to evaluate the RFACs' performance under different scheduling policies. These objectives are classified into two categories: time based objectives and due date based objectives. The makespan and percentage of robots idle time are in the first category while total tardiness, maximum tardiness and percentage of tardy jobs fall into the second category. The following notations are used to formulate the mathematical expressions of the objectives.

P	Products index ($P = 1, 2, \dots, i$).
Q	Part index ($Q = 1, \dots, j$)
R	Robots index ($R = 1, 2, \dots, k$)
S	Resource index ($S = 1, \dots, l$)
OP	Assembly operation index ($OP = op_{1i}, op_{2i}, \dots, op_{mi}$) of product i
T_{mi}	Time of assembly operation m of product i
$T_{(s \rightarrow l)i}$	Time taken by robot to travel between two resources ($s \rightarrow l$), 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
U_i	Indicator for whether product i is tardy or not

Makespan (C_{max})

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 (Danping and Lee, 2010, Li and Chen, 2011). The makespan can be represented as:

$$C_{max} = \max_{1 \leq i \leq p} (C_i) \quad \forall R \quad (1)$$

Percentage of robots idle time (% I_T)

The robots idle time is the time the robots must sit idle (Desal, 1997): in other words, the waiting time of robots before the start of any actions such as movement, tool-change, pick-up and assembly. The percentage of robots idle time can be calculated using the following formula:

$$\% I_T = \left(1 - \frac{\sum_{OP=1}^m T_{mi} + \sum_{S=1}^l T_{(s \rightarrow l)i} + \sum_{Q=1}^j T_{ji}}{C_{max}} \right) \times 100 \quad \forall i \quad (2)$$

Total tardiness (**TD**)

The total tardiness is the sum of the tardiness of all jobs (Restrepo and Balakrishnan, 2008, Berrichi and Yalaoui, 2013). The minimisation of total tardiness aims to find schedules that satisfy the customers' due dates. Total tardiness can be represented as:

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

Maximum tardiness (**MaxTD**)

The maximum tardiness is the largest difference between the completion time and the due date committed to for all jobs (Tavakkoli-Moghaddam et al., 2005, Baptiste and Schieber, 2003). The maximum tardiness can be represented using the following formula:

$$\text{MaxTD} = (\max_{1 \leq i \leq p} [C_i - D_i]) \quad (4)$$

Percentage of tardy jobs (**%N_U**)

The tardy job is the sum of jobs which are overdue divided by the number of jobs (Domingos and Politano, 2003, Jayamohan and Rajendran, 2000). The percentage of tardy jobs can be expressed using the following formula:

$$\%N_U = \frac{\sum_{P=1}^i U_i}{\sum_{P=1}^i N_i} \times 100, \quad U_i = \begin{cases} 1, & \text{if } C_i > D_i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

3.2 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. These rules are used to generate the sequence of job flow to the system. In this research, each product is considered as an independent job. The algorithm of the scheduling module is depicted in Figure 4. In the proposed methodology, the scheduling module contains two types of rules which allow the decision maker to determine the job sequencing.

The first type is the rules that have been commonly used in scheduling for solving scheduling problems. The following is a list of the common scheduling rules used in this thesis.

- Short Processing Time (SPT): select job with minimum processing time first.
- Long Processing Time (LPT): select job with maximum processing time first.
- Random (RAND): jobs are sequenced randomly.
- Earlier Due Date (EDD): jobs are sequenced according to their due dates.

- Critical Ratio (CR): select job with minimum critical ratio first.
- Minimise Slack Time (MST): jobs are sequenced according to their urgency.

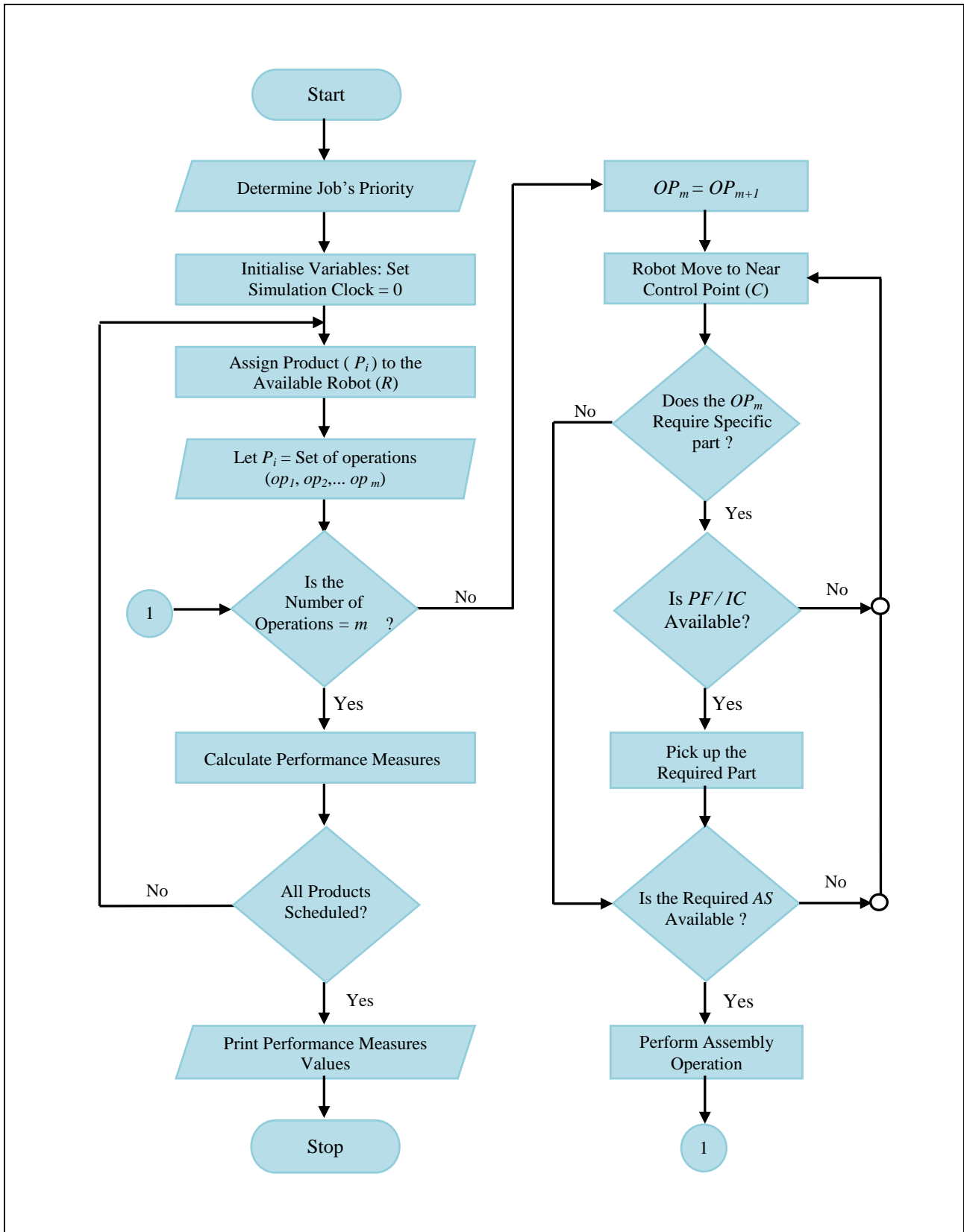


Fig. 4. Flow chart of the algorithm of the scheduling module

The second type is a rule developed for scheduling RFACs in a multi-product assembly environment, called a fuzzy sequencing rule (FSR) which is constructed by combining all the input variables using fuzzy logic technique. In this research, the job sequence determination is carried out by evaluating the normalisation of each job variable such as processing time, batch size, due date and number of required stations. The normalisation of the four inputs to the system can be defined, using the following notations:

μ_T^i Normalisation of the processing time T of product i

μ_D^i Normalisation of the due date D of product i

μ_N^i Normalisation of the batch size N of product i

μ_S^i Normalisation of the number of required stations S for product i

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 the minimum total processing time to the difference between the maximum and minimum total processing times of the same product, as shown in equation 6.

$$\mu_T^i = \frac{[(T_i) - \text{Min}(T_i)]}{[\text{Max}(T_i) - \text{Min}(T_i)]}, \quad 0 \leq \mu_T^i \leq 1 \quad (6)$$

According to the formula 6, it can be seen that the product with minimum processing time has a normalised value of 0 and the product with maximum processing time has a normalised value equal to 1.

μ_D^i : The normalisation of the due date of product i is defined as the ratio of the difference between the due date of product i and the minimum due date to the difference between the maximum and minimum due dates, as expressed in equation 7.

$$\mu_D^i = \frac{[(D_i) - \text{Min}(D_i)]}{[\text{Max}(D_i) - \text{Min}(D_i)]}, \quad 0 \leq \mu_D^i \leq 1 \quad (7)$$

From the previous formula, it can be concluded that the product with minimum due date has a normalised value of 0 and the product with maximum due date has a normalised value of 1.

μ_N^i : The normalisation of the batch size of product i is defined as the ratio of the difference between the batch size of product i and the minimum batch size to the difference between the maximum and minimum batch size of product i , as shown in equation 8.

$$\mu_N^i = \frac{[(N_i) - \text{Min}(N_i)]}{[\text{Max}(N_i) - \text{Min}(N_i)]}, \quad 0 \leq \mu_N^i \leq 1 \quad (8)$$

Again, formula 8 shows that the product with minimum due date has a normalised value equal to 0 and that with maximum due date has a normalised value 1.

μ_S^i : The normalisation of the number of required stations for product i is defined as the ratio of the difference between the maximum number of required stations and the number of required stations for product i to the difference between the maximum and minimum numbers of required stations, as expressed in equation 9.

$$\mu_S^i = \frac{[\text{Max}(S_i) - (S_i)]}{[\text{Max}(S_i) - \text{Min}(S_i)]}, \quad 0 \leq \mu_S^i \leq 1 \quad (9)$$

When the number of required stations is considered, as defined in formula 9, the product with the minimum number of required stations has a normalised value of 1 and the product with the maximum number of required stations has a normalised value of 0.

The overall normalisations are combined 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 high priority. In this part of the research, a mathematical model is developed to calculate the jobs' priority, using fuzzy logic. Section 3.4 will explain the implementation of the fuzzy-based mathematical model. As mentioned in chapter two, fuzzy logic systems (FLS) consist of four main components: knowledge base, fuzzification, inference engine and defuzzification, as shown in Figure 5.

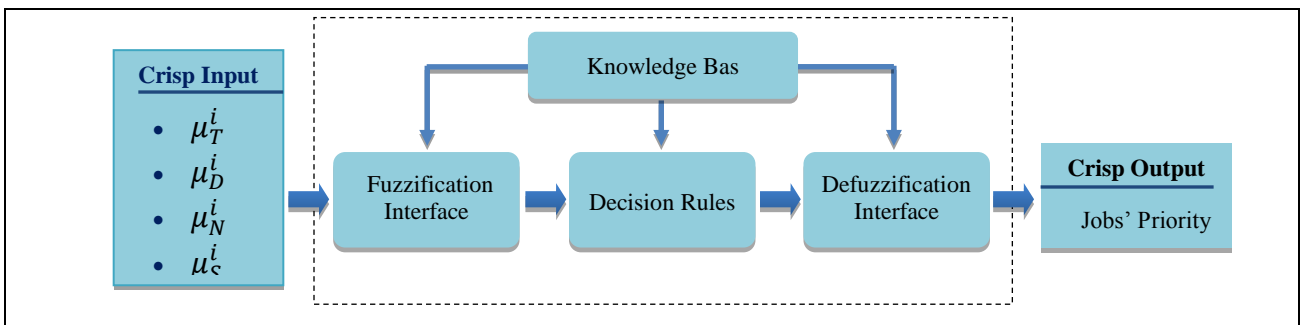


Fig. 5. Fuzzy logic system configuration for job selection

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 rules are prepared to establish a knowledge base (Abd et al., 2012, Abd et al., 2102), as depicted in Figure 6. The next sub section will describe the previous three steps.

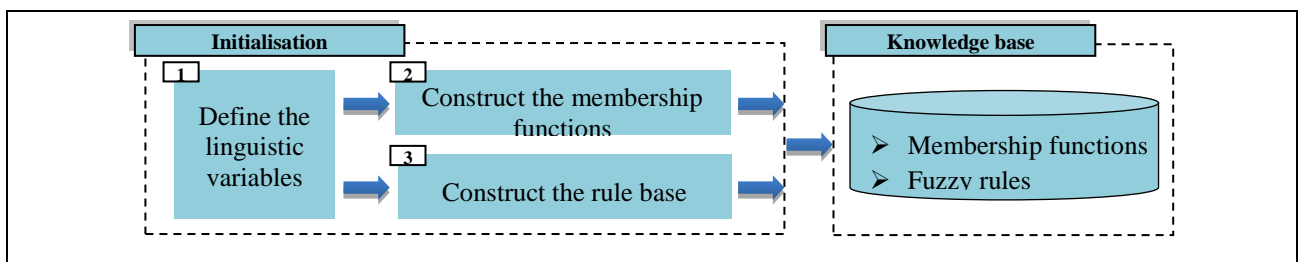


Fig. 6. Knowledge base construction of fuzzy logic approach

Linguistic Variables

A linguistic variable is the procedure to describe variables in terms of words instead of their values. In general, a linguistic variable consists of a set of terms called linguistic terms, denoted by T. 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. These terms are called a fuzzy set of the processing time. Hence, a linguistic variable of processing time could be T [processing time] = [Short, Medium, Long].

Membership functions

A membership function (MF) embodies a fuzzy set \tilde{A} graphically. The values of the membership functions are between 0 and 1, denoted by $\mu_{\tilde{A}}(x)$ where x is an element of \tilde{A} ; these values are called degree of membership. Figure 7 shows the most well-known of membership functions shapes, triangular and trapezoidal (Mendel, 1995).

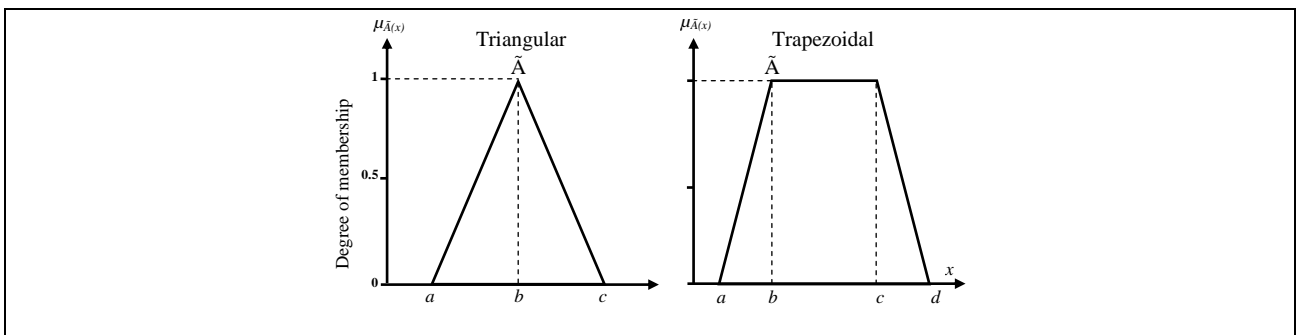


Fig. 7. Two examples of fuzzy numbers, triangular and trapezoidal

A triangular fuzzy number can be defined by triplet (a, b, c). A trapezoidal fuzzy number is represented by four points (a, b, c, d) as shown in equations 10 and 11.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ \frac{c-x}{c-b}, & b \leq x \leq c, \\ 0, & x > c, \end{cases} \quad (10)$$

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \frac{d-x}{d-c}, & c < x \leq d, \\ 0, & x > d, \end{cases} \quad (11)$$

Fuzzy rules

A fuzzy rule is structured to control the output variable. These rules can be provided by experts or may be extracted from numerical data. A fuzzy rule has two parts, the antecedent and the consequent: IF <antecedent> THEN <consequent>. For instance, IF x is A THEN y is B; where x and y are variables and A and B are linguistic variables determined by fuzzy sets.

3.3 Simulation Module

Once the scheduling parameters, objective functions, constraints and decision variables are determined, the simulation module is defined and constructed. In this module, a computer simulation model of the RFACs is built, to evaluate the system performance under different scheduling strategies. In this research, simulation software called SIMPROCESS is used to build and simulate the assembling processes (Swegles, 1997, CACI, 2006). The process of simulation RFACs is achieved through main four stages, using SIMPROCESS software. These stages are shown in Figure 8.

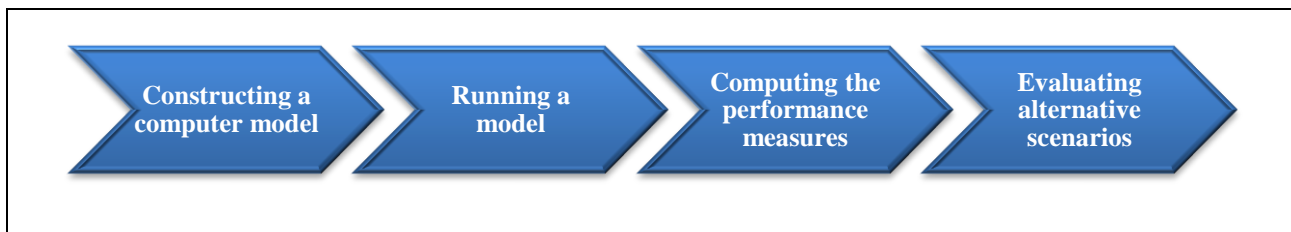


Fig. 8. Simulation process in SIMPROCESS

Constructing a computer model of the RFACs is divided into three steps; *define the model, construct a 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, assumptions, 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 to 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 Figure 8 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 Figure 8 is evaluating alternative scenarios, in order to evaluate the RFACs' performance under different scheduling strategies.

The overall architecture of the proposed methodology is depicted in Figure 9.

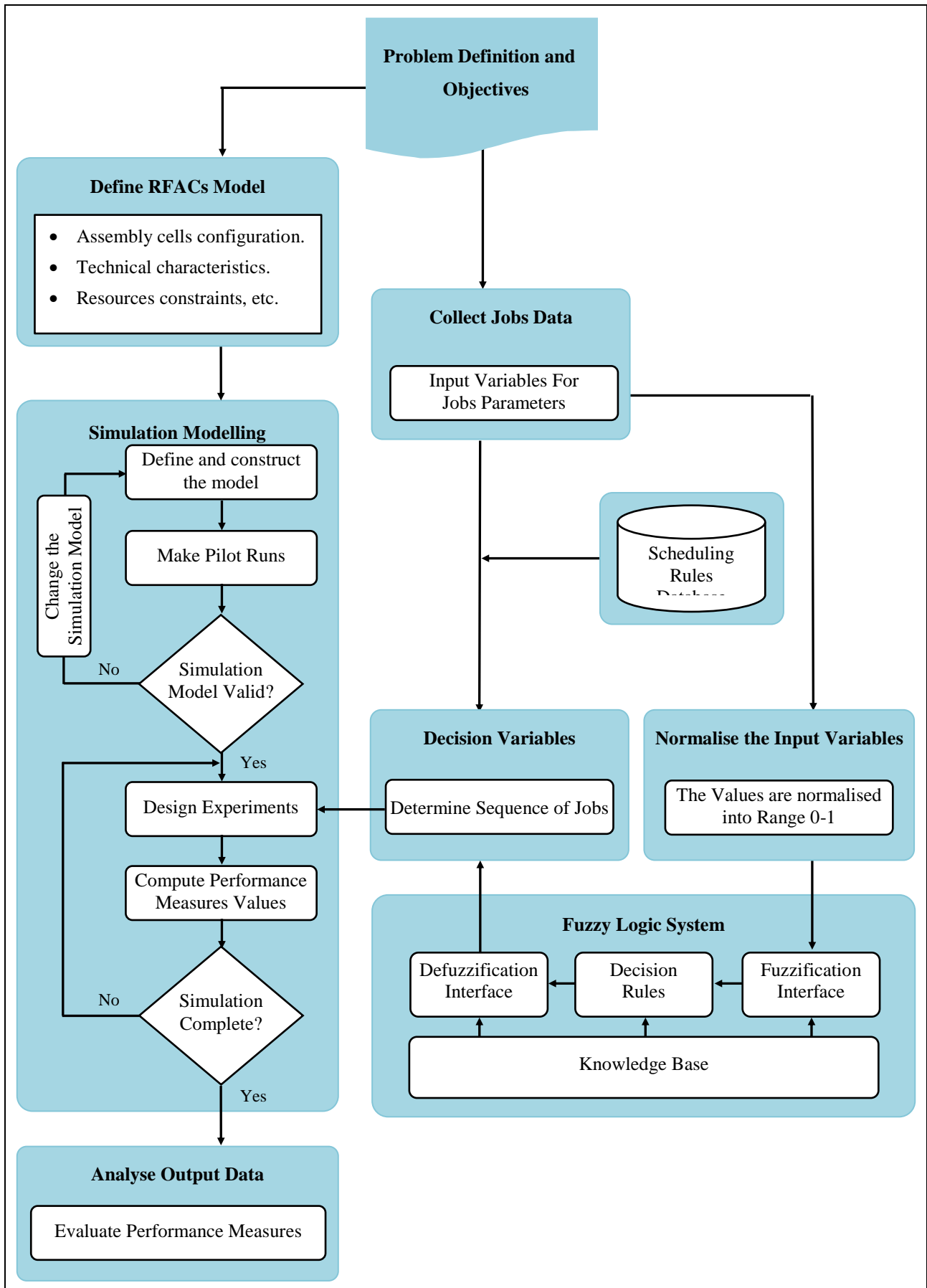


Fig. 9. Proposed strategy for scheduling problems in RFACs

4. Conclusion

In this chapter, a new methodology for scheduling RFACs has been developed with the objective of minimising makespan, robots idle time, total tardiness, maximum tardiness and number of tardy jobs. The developed methodology was divided into three main modules, namely pre-processing, scheduling and simulation.

In the first one, the required components of modelling the scheduling problems in the RFACs were defined and described. These components were: parameters, objective functions, constraints and decision variables.

In the second module, the schedule for assembling multi products was generated via a new and sophisticated scheduling rule, namely fuzzy sequencing rule (FSR). FSR was constructed using a fuzzy-based mathematical model. This model used the membership functions to find the contribution of each product type to the output (job's priority) and then generate the sequence of products flow to the RFACs. The sequence generation was determined by normalisation of each job variable such as processing time, batch size, due date and number of required stations.

In the last module, a computer simulation model of the RFACs was built in SIMPROCESS, and then simulated under different scenarios, depending on the outcome of the scheduling module.

The methodology developed in this chapter has been applied for scheduling RFACs, which will be presented in the companion chapter – Part II.

5. References

- Abd, K., Abhary, K. & Marian, R. 2011a. A scheduling framework for robotic flexible assembly cells. *AIJSTPME-Asian International Journal of Science and Technology in Production and Manufacturing Engineering*, 4, 30-37
- Abd, K., Abhary, K. & Marian, R. 2011b. Scheduling and performance evaluation of robotic flexible assembly cells under different dispatching rules. *Advances in Mechanical Engineering*, 1
- Abd, K., Abhary, K. & Marian, R. 2012. Intelligent Modeling of Scheduling Robotic Flexible Assembly Cells Using Fuzzy Logic *12th WSEAS International Conference on Robotics, Control and Manufacturing Technology*. Rovaniemi, Finland
- Abd, K., Abhary, K. & Marian, R. 2102. Efficient Scheduling Rule for Robotic Flexible Assembly Cells Based on Fuzzy Approach. *45th CIRP Conference on Manufacturing Systems*
- Baptiste, P. & Schieber, B. 2003. A note on scheduling tall/small multiprocessor tasks with unit processing time to minimize maximum tardiness. *Journal of Scheduling*, 6, 395–404
- Berrichi, A. & Yalaoui, F. 2013. Efficient bi-objective ant colony approach to minimize total tardiness and system unavailability for a parallel machine scheduling problem. *The International Journal of Advanced manufacturing Technology*
- Bilkay, O., Anlagan, O. & Kilic, S. E. 2004. Job shop scheduling using fuzzy logic. *International Journal of Advanced Manufacturing Technology* 23, 606–619

- Buil, R., Piera, M. A. & Luh, P. B. 2010. Improvement of lagrangian relaxation convergence for production scheduling. *IEEE transactions on automation science and engineering*, 9, 137-147
- CACI 2006. User's Manual: Simprocess. CACI Products Company, La Jolla, CA.
- Canbolat, Y. B. & Gundogar, E. 2004. Fuzzy priority rule for job shop scheduling. *Journal of Intelligent Manufacturing*, 15, 527- 533
- Chan, T. S. F. & Chan, K. H. 2004. A comprehensive survey and future trend of simulation study on FMS scheduling. *Journal of Intelligent Manufacturing* 15,87–102
- Danping, L. & Lee, C. K. M. 2010. A review of the research methodology for the re-entrant scheduling problem. *International Journal of Production Research*, 49, 2221–2242
- Desal, N. K. 1997. *Scheduling algorithm for flexible manufacturing cells*. Master of Science, University of Manitoba
- Domingos, J. C. & Politano, P. R. 2003. On-line scheduling for flexible manufacturing systems based on fuzzy logic. *IEEE International Conference on Systems, Man and Cybernetics*, 5, 4928 – 4933
- Jayamohan, M. S. & Rajendran, C. 2000. New dispatching rules for shop scheduling: A step forward. *International Journal of Production Research*, 36, 563–586
- Kumar, R. R., Singh, A. K. & Tiwari, M. K. 2004. A fuzzy based algorithm to solve the machine-loading problems of a FMS and its neuro fuzzy Petri net model. *International Journal of Advanced Manufacturing Technology*, 23, 318-341
- Lee, J.-K. and Lee, T.-E. 2002. Automata-based supervisory control logic design for a multi-robot assembly cell. *International Journal Computer Integrated Manufacturing*, 15, 319-334
- Li, Y. & Chen, Y. 2011. An effective TPA-based algorithm for job-shop scheduling problem. *Expert Systems with Applications*, 38, 2913–2918
- Mahdavi, I., Fekri Moghaddam Azar, A. H. & Bagherpour, M. Applying fuzzy rule based to flexible routing problem in a flexible manufacturing system. *IEEE International Conference on Industrial Engineering and Engineering Management*, 2009. 2358-2364
- Manivannan, S., 1993. Robotic collision avoidance in a flexible assembly cell using a dynamic knowledge base. *IEEE Transactions on Systems, Man, and Cybernetics*, 23, 766-782
- 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.
- Mendel, J. M. 1995. Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*, 83, 345–377
- 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
- Ramasesh, R. 1990. Dynamic job shop scheduling: a survey of simulation research. *OMEGA: International Journal of Management Science*, 18, 43-57

- Restrepo, I. M. & Balakrishnan, S. 2008. Fuzzy-based methodology for multi-objective scheduling in a robot-centered flexible manufacturing cell. *Journal of Intelligent Manufacturing*, 19, 421-432
- Sawik, T., 1999. Production planning and scheduling in flexible assembly systems. Poland, Springer
- Sridhar, S., Prabakaran, T. & Saravanan, M. 2010. Optimisation of sequencing and scheduling in hybrid flow shop environment using heuristic approach. *International Journal of Logistics Economics and Globalisation*, 2, 331-351
- Srinoi, P., Minyong, P., Shayan, E. & Ghotb, F. 2008. Routing and sequencing determination in flexible manufacturing system using a fuzzy logic approach *Asian International Journal of Science and Technology in Production and Manufacturing*, 1, 127-138
- Srinoi, P., Shayan, E. & Ghotb, F. 2006. A fuzzy logic modelling of dynamic scheduling in FMS. *International Journal of Production Research*, 44, 2183–2203.
- Subramaniam, V., Ramesh, T., Lee, G. K., Wong, Y. S. & Hong, G. S. 2000. Job shop scheduling with dynamic fuzzy selection of dispatching rules. *International Journal of Advanced Manufacturing Technology*, 16, 759–764
- Swegles, S. 1997. Business process modeling with SIMPROCESS. *Winter Simulation Conference*. Piscataway, NJ
- Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M. & Azaron, A. 2005. Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert. *Applied Mathematics and Computation*, 167, 1430-1450
- Vidyarthi, N. K. & Tiwari, M. K. 2001. Machine loading problem of FMS: a fuzzy-based heuristic approach. *International Journal of Production Research*, 39, 953–957
- Xidias, E. K. Zacharia, P. T. and Aspragathos, N. A. 2010. Time-optimal task scheduling for two robotic manipulators operating in a three-dimensional environments. *Journal of Systems and Control Engineering*, 224, 845-855
- 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