

## Non-Reshuffle-Based Approach for Rescheduling of Flexible Manufacturing System

Zalmiyah Zakaria\*<sup>1</sup>, Safaai Deris<sup>2</sup>, Muhamad Razib Othman<sup>1</sup>, Shahreen Kasim<sup>3</sup>

<sup>1</sup>Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

<sup>2</sup>Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan, Karung Berkunci 01, 16300 Bachok, Kelantan, Malaysia,

<sup>3</sup>Soft Computing and Data Mining Centre, Faculty of Computer Science And Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

E-mail: \*zalmiyah@utm.my

---

**Abstract**— Scheduling and rescheduling play vital roles in ensuring the effectiveness of the production control in flexible manufacturing system (FMS). The failure of these systems may interrupt the efficiency of the production activities and thus may lessen the profit to be gained by the company. The FMS scheduling problem is considered as dynamic as new orders may get in every day. The new orders need to be immediately desegregated with the existing production schedule by preserving the efficiency and stability of the existing schedule. This research applies the non-reshuffle-based genetic match-up algorithms which admit new orders by manipulating available machine idle times to address rescheduling problem in a FMS that practises the pull strategy. The idea of the match-up approach is to update only a part of the initial schedule and genetic algorithms used to optimise the solution within the rescheduling horizon in such a way in order to preserve the efficiency and stability of the shop floor. The proposed methodology has been tested using different rescheduling parameters. The experiments show that the rescheduling method improves efficiency and stability of the new schedule.

**Keywords**— dynamic scheduling; match-up rescheduling; genetic algorithms; flexible manufacturing system.

---

### I. INTRODUCTION

There are two main manufacturing control strategies that exist, namely push and pull manufacturing [1]. Traditional manufacturing applies push strategy in which the manufacturing starts, regardless of the requirements for that product and then pushes to the following step that can be an internal process or a selling activity in the market (also known as make-to-stock). Meanwhile, customer demand is a pull strategy. The scheduling problem in an advanced manufacturing system such as the flexible manufacturing system (FMS) that practises pull strategy is considered as dynamic as new orders may arrive intermittently at any time. The rescheduling process is needed to accommodate new orders proportionate to the existing production schedule with minimum number of changes and also to preserve the efficiency of the existing schedule. There are three approaches have been proposed to solve FMS rescheduling problem which are [2]: reactive, robust and predictive-reactive scheduling. Reactive scheduling produces a schedule over time as the shop floor state changes using local information of the dispatched jobs. Commonly, dispatching rules or other heuristics are used to prioritise the jobs waiting for processing. This approach is also called on-

line scheduling [3] or dynamic scheduling [4] in the literature. Robust scheduling refers to a schedule that is produced in such a way that it absorbs the disruptions on the shop floor. Some examples of this work are the ones conducted by Davenport *et al.* [5] and Jensen [6]. Predictive-reactive scheduling is an initial schedule which optimises the shop floor performance that is generated and then updated whenever necessary. Yang and Wu [7] and Shi-jin *et al.* [8] are among the previous researchers that involved significantly in predictive-reactive scheduling. In predictive-reactive scheduling, an optimal predictive schedule that is based on certain criteria and constraints is provided at the first stage, and then the schedule is revised during the implementation stage to adapt changes in the shop floor. This revision can be performed in several ways, either by generating a new schedule [7], [8] or by repairing the existing schedule [9]-[11]. Repairing the existing schedule may involve partial or total rescheduling methods. However, generating new schedule and total rescheduling may cause shop floor nervousness as the schedule will be exposed to frequent and huge amount of changes. The bigger number of operations affects the lesser stability of the shop floor. Thus, partially repaired schedule is expected to be more applicable to maintain the stability of the shop floor. Right-shifting and

match-up scheduling are two types of partial rescheduling. Right-shifting is a simple repairing method that is easy to implement, but it may lead to poor performance as all remaining operations are delayed. Match-up scheduling repairs the existing schedule by preserving the initial schedule as much as possible to maintain schedule stability.

There are many techniques that have been used for solving predictive-reactive scheduling such as knowledge-based or expert systems [12], [13], genetic algorithm (GA) [7], [11] and filtered beam search [8]. GA is amongst the very popular heuristic search methods [14]-[18] as it often provides a good performance solution for many types of combinatorial optimisation problems. Since Davis [14] proposed the first GA-based technique to handle scheduling problems in 1985, GA has been widely used in different kinds of scheduling problems such as university and exam timetabling [15], [19], maintenance scheduling [20]-[21], nurse rostering [22] and production scheduling [7], [18] – usually with encouraging results.

Chromosome is a basic component of GA which represents a point of the search space. This point is also referred to as individual, which manifests a solution of a particular optimisation problem. Chromosome representation significantly influences the GA performance and an improper arrangement of the chromosome structure will often cause a poor performance [23]. This research exploits the chromosome representation in order to improve the performance of GA in producing a good solution at a reasonable time.

As rescheduling in FMS involves open rolling horizon, the match-up approach is used to define rescheduling horizon. This is essential in order for the rescheduling process to involve only a minimum number of affected operations within the rescheduling horizon. Match-up algorithms have been studied by Bean *et al.* [9], and Birge and Dempster [24] in the context of single machine and flow shop problems. Akturk and Gorgulu [10] presented another match-up scheduling that partially rescheduled a flow shop when a machine breakdown occurred. Moratori *et al.* [11] proposed a number of match-up strategies based on precedence-based and non-precedence-based collected machine idle-times for a real world job shop scheduling. In this work, a genetic match-up rescheduling methodology that belongs to the predictive-reactive scheduling is presented to cater to an FMS rescheduling problem that is subjected to changes and results in shop floor nervousness due to dynamic environment.

In this study, we investigate the dynamic scheduling problem of a discrete manufacturing involving different product models that also need different processes. The production consists of three main stages that are forming, heat treatment and assembly and under each of these stages, there are several processes, each with very distinct characteristics. Each of the processes is carried out on separate machines or on a single machine centre or workstation. Day-to-day a new order (or orders) may come in to be processed on the shop floor. In this research, both the normal and the rush orders are considered. Currently robust method is applied to absorb any changes in the shop floor. The right-shifting strategy is used when new rush orders arrive and reactive method with priority rules is used

for normal orders. Consequently, practising the robust method resulting low performance of the shop floor as every order only can be delivered to the customer after four months production time. Meanwhile, postponing the remaining operations in order to prioritize the rush orders may affect the stability of the shop floor.

The FMS scheduling problems are more complex than the conventional scheduling problem due to the assortment of machine configurations (flexible machines), large scale dimension (large number of machines and jobs), a wide range of parameters and uncertainties involved [25], [26]. Thus, the problem must be simplified to make it tractable to be solved. In order to simplify the scheduling problem, the related processes are grouped together to match with the located machines in the workstations. Consequent to that, the parts will be assigned to the workstation that matches to the processes needed by that particular part. In addition, the processes for the non-assembly stages (forming and heat treatment) and assembly stage are separated by different jobs and bounded with precedence relation. Accordingly, the jobs required for assembly may only be processed after the non-assembly processes for that particular part have been completed. The FMS scheduling problem as addressed in this paper was taken from the process plan. The process plan is generated for each part to be manufactured in a manufacturing system which specifies operations to be performed and their sequences, required resources, as well as process parameters of each operation.

## II. MATERIAL AND METHOD

The scheduling problem involves the allocation of a number of jobs on machines which are grouped into 18 workstations. Thus, the scheduling problem can be described as follows. There are a set of  $n$  jobs  $J_i = \{J_1, J_2, \dots, J_n\}$  whereby  $i = 1, \dots, n$  and a set of  $m$  workstations  $M_k = \{M_1, M_2, \dots, M_m\}$  where  $k = 1, \dots, m$ . Each job  $J_i$  consists of  $q_i$  operations and has its corresponding release time  $R_i$ , due date  $D_i$  and priority weight  $W_i$ . The priority weight  $W_i$  determines the level of importance of an order (rush or normal order). Each operation  $O_{ij}$ , in which  $j = 1, \dots, q_i$  can be performed on a number of alternative machines with possibly the same or different processing times  $P_{ij}$ . There are  $x$  machine types. A machine type is denoted by  $Y_b$ , in which  $b = 1, \dots, x$ . The operations  $O_{ij}$  are dependent based on their precedence relations. The order or the precedence of the operations for each job is fixed and known beforehand. The FMS scheduler produces an ordered sequence of operations to be processed on each machine with the processing start times which optimise the desired objective functions and satisfy the problem constraints. In the dynamic FMS environment, new jobs may arrive after processing begins. Hence, the objective of problem rescheduling is to integrate the new job into the existing schedule, in such a way that the stability of the running schedule is maintained with at least the same level of schedule performance. The desired schedule performances in pull manufacturing are anticipated to simultaneously minimise the tardiness, the earliness and the mean flow-time costs. Figure 1 shows the relationship between tardiness/earliness cost and completion time in practice [27]. The tardiness cost increases exponentially when the completion time  $C_i$  increases from

the expected due date  $D_i$ , while the earliness cost increases gradually when the completion time  $C_i$  decreases from the expected due date  $D_i$ . Meanwhile the work-in-progress (WIP) cost linearly increases with the growth of the mean flow time of the jobs as shown in Figure 2.

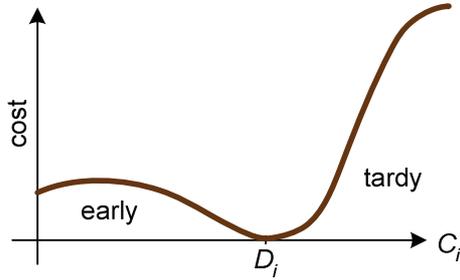


Fig. 1 Earliness and tardiness cost function [27]

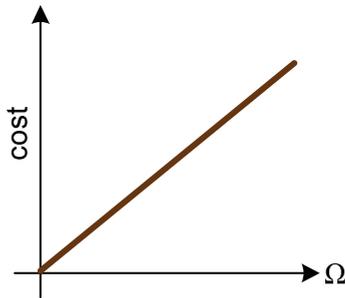


Fig. 2 WIP cost function

The rescheduling methodology consists of four main steps as shown in Figure 3. When new job,  $J_{n+i}$  arrives at time  $t$ , Step 1 to Step 4 below must be followed:

**STEP 1**

- (a) Identify the initial schedule that is currently implemented in the shop floor.
- (b) Determine the rescheduling horizon for the initial schedule,  $[startHorizon, endHorizon]$ .
- (c) Let  $t_i$  be the arrival time of new job  $i$  and  $t_i^+$  be the start time that had been approved by the shop floor's decision maker for adding the new job to the schedule. Therefore, the rescheduling task needs to be performed between time  $t$  and  $t^+$  as the  $t^+$  will be the  $startHorizon$ . Then, determine the  $endHorizon$ .

**STEP 2**

- (a) Identify the idle timeslots within the rescheduling horizon.
- (b) Collect the idle timeslots from each machine.

**STEP 3**

- (a) Construct and initialise the chromosome.

**STEP 4**

- (a) Evolve  $nGen$  times, where  $nGen$  is the number of generations.
- (b) Perform mutation (select, mutate and evaluate).
- (c) Perform crossover (select, crossover and evaluate).

*A. Performance Measures*

The performance measures used in this research are of the main criteria emphasised in pull manufacturing. Both efficiency and stability are considered as performance measures. Efficiency is considered to produce a good quality schedule, while the stability of the existing schedule must be kept by producing the fewest number of changes in terms of timing, operation sequencing and machine assignment.

*1) Schedule Efficiency*

In order to preserve the quality of the existing schedule, efficiency measures involving the mean flow time, the weighted quadratic tardiness and the weighted earliness objectives were used. As a normal practise in pull manufacturing, the tardiness and earliness measures are used to handle the trade-off between meeting due dates and holding excessive inventory [28], respectively while the mean flow-time measure tends to reduce the number or the size of work-in-progress (WIP). The performance measures used in this research are defined as follows:

- (i) The mean flow time of jobs  $\Omega$  can be found using the following formula:

$$\Omega = \frac{\sum_{i=1}^n C_i - R_i}{n} \quad (1)$$

The flow time of job  $i$  is the total time of job  $i$  spent in the shop floor, where  $C_i$  and  $R_i$  represent the completion time and the release time of job  $i$  respectively, and  $i = 1, \dots, n$ .

- (ii) The weighted quadratic tardiness  $F$  is defined in the following way:

$$F = \sum_{i=1}^n W_i T_i^2 \quad (1)$$

where  $W_i$  is the priority weight of job  $J_i$  and  $T_i$  is the tardiness of job  $J_i$ , defined as  $\max [0, C_i - D_i]$ . This objective function considers the importance of meeting the due dates, and the fact that a job becomes more crucial with each time unit after passing its due date [7], [8], [29].

- (iii) The weighted earliness  $\Xi$  is defined in the following way:

$$\Xi = \sum_{i=1}^n W_i E_i \quad (2)$$

where  $W_i$  is priority weight of job  $J_i$  and  $E_i$  is the earliness of job  $J_i$ , defined as  $\max [0, D_i - C_i]$ . This objective function considers the importance of holding excessive inventory, and the fact that a job

becomes crucial with each time unit preceding its due date.

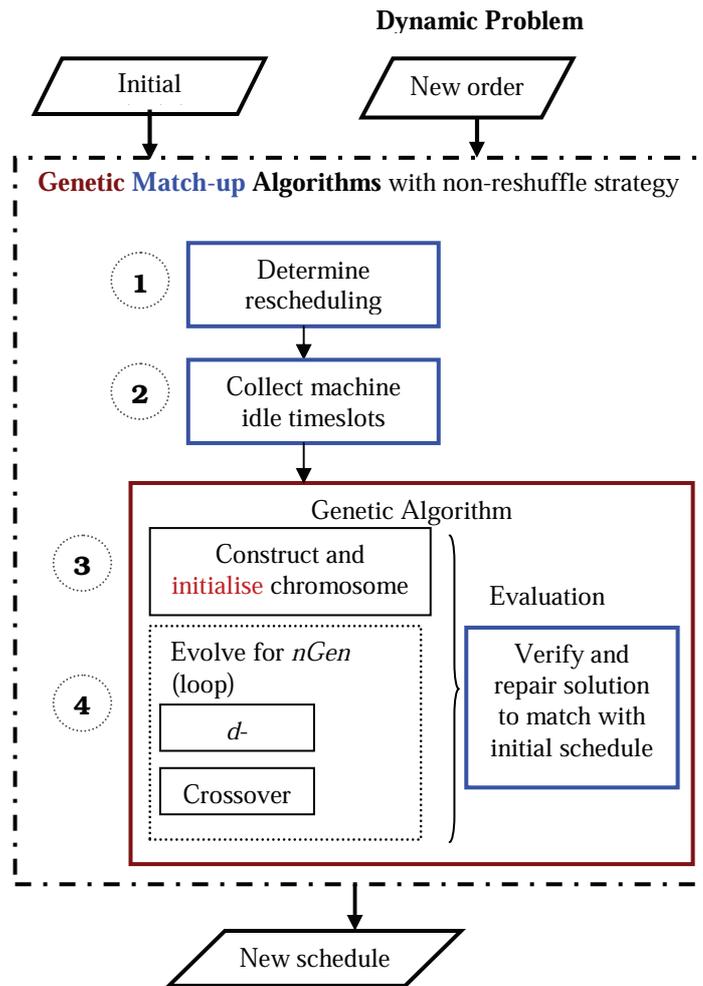


Fig. 3 The Rescheduling Methodology

### 2) Schedule Stability

In order to maintain the stability of the new schedule, the following three conditions are considered: start time deviation, operation sequence deviation and machine migration. Thus, the stability measure is calculated using the following expression:

$$Q = 1 - \frac{\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^m \text{penalty}_{i,j,k}}{\sum_{i=1}^n q_i} \quad (3)$$

where  $\text{penalty}_{i,j,k}$  is the summation of penalties given to operation  $O_{i,j}$  at machine  $k$ . Each operation is penalised as follows:

$$\text{penalty}_{i,j,k} = \begin{cases} 1 & \text{if operation } O_j \text{ is processed on different machine} \\ 0.6 & \text{if the position of operation } O_j \text{ on machine } k \text{ changes in the new sequence} \\ 0.3 & \text{if start time of operation } O_j \text{ on machine } k \text{ changes} \\ 0 & \text{if there is no change} \end{cases} \quad (4)$$

These objectives are incommensurable since they are measured in different units. The mean flow time of the jobs is given in time units, whereas weighted tardiness are calculated as squared of the total weighted tardiness of the jobs, weighted earliness are given as the total of weighted earliness of the jobs, while the stability of the schedule is given in between 0 and 1. However, the objective values have to be evaluated simultaneously to determine the quality of the schedules. Thus, satisfaction grades [30]-[31] are used to reflect the decision maker preferences with respect to the achieved values of the objectives. The values of the objectives are represented by satisfaction grades, which take values between [0, 1], where 0 denotes full dissatisfaction and 1 represents full satisfaction of the obtained objective value. Then, the satisfaction grades of all the objectives are aggregated in an overall satisfaction grade.

### 3) Satisfaction Grades

The objectives (Equation 1 to Equation 4) are transformed into objective of maximising the corresponding satisfaction grades as follows:

- (i) Satisfaction grade of mean flow time of jobs, is given by:

$$S_{\Omega} = \begin{cases} 1 & \text{if } \Omega \leq Min_{\Omega} \\ (Max_{\Omega} - \Omega) / (Max_{\Omega} - Min_{\Omega}) & \text{if } Min_{\Omega} < \Omega < Max_{\Omega} \\ 0 & \text{if } \Omega \geq Max_{\Omega} \end{cases} \quad (5)$$

where  $\Omega$  is the mean flow time of the jobs which is calculated by the expression given in Equation 1,  $Min_{\Omega}$  is the mean processing time of the jobs, and  $Max_{\Omega}$  is the rescheduling period, whereby within the differences between the *startHorizon* and the *endHorizon*. There is full dissatisfaction when  $\Omega \geq Max_{\Omega}$  and full satisfaction when  $\Omega \leq Min_{\Omega}$ , while satisfaction level is linearly decreasing when  $\Omega$  increases from  $Min_{\Omega}$  to  $Max_{\Omega}$ .

- (ii) Satisfaction grade of the weighted quadratic tardiness  $F$  of jobs,  $S_F$ , is given as follow.

$$S_F = \begin{cases} 1 & \text{if } F = Min_F \\ (Max_F - F) / (Max_F - Min_F) & \text{if } Min_F < F < Max_F \\ 0 & \text{if } F > Max_F \end{cases} \quad (7)$$

where  $F$  is the weighted quadratic tardiness of jobs calculated by the expression given in Equation 2.  $Min_F = 0$  represents no tardy jobs and  $Max_F = 0.5n\delta^2$  is the assumption that at most 50% of jobs are tardy. As the completion time of each job for the new schedule are immeasurable at the initial stage of the rescheduling process, the quadratic *endHorizon* has been used as the basis of calculation, where  $\delta$  is the *endHorizon* and  $n$  is the total number of jobs. There is a full dissatisfaction when  $F \geq Max_F$  and full satisfaction when  $F = 0$ , while satisfaction level is linearly decreasing when  $F$  increases from  $Min_F$  to  $Max_F$ .

- (iii) Satisfaction grade of the weighted earliness  $\Xi$  of jobs,  $S_{\Xi}$ , is presented as follow.

$$S_{\Xi} = \begin{cases} 1 & \text{if } \Xi > Max_{\Xi} \\ (Max_{\Xi} - \Xi) / (Max_{\Xi} - Min_{\Xi}) & \text{if } Max_{\Xi} \geq \Xi \geq Min_{\Xi} \\ 0 & \text{if } \Xi < Min_{\Xi} \end{cases} \quad (8)$$

where  $\Xi$  is the weighted earliness of jobs calculated using Equation 3.  $Min_{\Xi} = 0$  represents no early jobs and  $Max_{\Xi} = 0.5n\delta^2$  is the assumption that at most 50% of jobs are early. As the completion time of each job for the new schedule are immeasurable at the initial stage of the rescheduling process, the quadratic *endHorizon* has been used as the basis of calculation, where  $\delta$  is the *endHorizon* and  $n$  is the total number of jobs. There is a full dissatisfaction when  $\Xi \geq Max_{\Xi}$  and full satisfaction when  $\Xi = 0$ , while satisfaction level are linearly decreasing when  $\Xi$  increases from  $Min_{\Xi}$  to  $Max_{\Xi}$ .

- (iv) Satisfaction grade of stability  $Q$  of the new schedule,  $S_Q$ , as presented by:

$$S_Q = \begin{cases} 1 & \text{if } Q = 1 \\ Q & \text{if } 0 < Q < 1 \\ 0 & \text{if } Q = 0 \end{cases} \quad (9)$$

where  $Q$  is the stability of the new schedule compared to the previous one that is calculated in Equation 4. There is a full dissatisfaction when  $Q = 0$  and full satisfaction when  $Q = 1$ , while satisfaction level is linearly increasing when  $Q$  increases from 0 to 1.

The satisfaction grade  $S$  of all objectives, used as the fitness function, is calculated as follows:

$$S = \frac{(w_{\Omega}S_{\Omega} + w_F S_F + w_{\Xi} S_{\Xi} + w_Q S_Q)}{4} \quad (10)$$

where  $w_{\Omega} = 0.5$ ,  $w_F = 1.5$ ,  $w_{\Xi} = 0.5$  and  $w_Q = 1.0$ . The total satisfaction grade  $S$  is divided by four which reflects the number of objectives. The tardiness and the stability are considered to be more important than the mean flow-time and the earliness. This is because the earliness and the flow time only affect the company's inventory and work-in-progress (WIP) respectively, but the tardiness is closely related to customer satisfaction, while the stability may affect the production system. Therefore the aim of rescheduling algorithms is to ensure as many orders as possible that meet the expected delivery date and to maintain the efficiency of the shop floor with the stable schedule, the small number or size of WIP and inventory.

### B. Genetic Match-Up Algorithms

An example of 7x6 FMS scheduling problem is shown in Table I. This problem consists of seven jobs (five jobs for the initial schedule and two newly arrived jobs, namely Job 6 and Job 7) and six different machines, which are sorted into three machine types. Each job consists of two to four operations with different processing times that need to be performed on three different machine types. Table II lists the machine types.

TABLE I  
AN EXAMPLE OF 7X6 FMS SCHEDULING PROBLEM

Job no.	Due date	Operation	Machine type	Processing time
1*	12	1	2	3
		2	3	3
		3	1	4
2*	12	1	1	2
		2	2	4
3	24	1	1	3
		2	3	5
		3	1	5
		4	3	4

4*	17	1	2	3
		2	3	4
		3	1	5
5	10	1	3	3
		2	3	4
6	25	1	3	3
		2	1	3
		3	1	4
7*	20	1	1	3
		2	3	4
		3	2	5

\* rush order

TABLE II  
LIST OF THE MACHINES

Machine Type	Machine
1	1
	2
2	3
3	4
	5
	6

When new job,  $J_{n+1}$  (namely  $J_5$  and  $J_6$ ) arrives at time  $t=2$ , a new schedule needs to be produced. To adopt the Genetic Match-up (GMU) rescheduling methodology, the following steps were performed:

#### STEP 1

- (a) Identify the initial schedule that is currently implemented in the shop floor.

(b) Determine the rescheduling horizon for the initial schedule,  $[startHorizon, endHorizon]$ . Let  $t = 2$ , the arrival time of new jobs,  $J_5$  and  $J_6$  and  $t + 5$  will be the  $startHorizon$ . Then, an improved precedence-based idle time collection strategy for multiple jobs insertion to parallel machines was used to define the  $endHorizon$ . The original strategy had been proposed by Moratori *et al.* [11]. Table III presents the example of determining the  $endHorizon$  based on the initial schedule. The table is divided into two parts which are the provided data on the left side and the calculated data on the right side. Meanwhile, the variety pair of shapes reflects the derivation of the values.

#### STEP 2

- (a) Identify the idle timeslots within the rescheduling horizon. The same strategy [11] is used to collect the available idle timeslot (not necessarily as a single timeslot) on machines required by the new job. The total idle time collected on each machine reflects the required processing time of new operation on that particular machine. The precedence constraints are considered during the collection process where the collection of the idle times for operation  $O_{n+1,j}$  on machine  $k$  starts after the completion time of the preceding operation  $O_{n+1,j-1}$ .

- (b) Find the collected idle timeslots from each machine.

#### STEP 3

- (a) Construct and initialise the chromosome. The algorithm uses the existing schedule as the basis of a chromosome. The idle timeslots collected from the existing schedule (within rescheduling horizon) will be allocated to each operation.

Let us consider that new jobs,  $J_6$  and  $J_7$  consist of six operations that require a machine from  $Y_3, Y_2, Y_1$  and  $Y_1, Y_3, Y_2$  for operations  $O_{61}, O_{62}, O_{63}$  and  $O_{71}, O_{72}, O_{73}$ , respectively.

The collected idle timeslots from the current schedule represent the values (allele) of the chromosome. Thus, each gene of the chromosome represents the proposed slot for an operation that needs to be inserted in the current schedule. aint propagating.

#### STEP 4 Evolve iteratively for $nGen$ times.

- (a) Perform mutation (select, mutate and evaluate). Mutation operator selects one individual randomly. In order to maintain the feasible solution, another idle timeslots from the candidate slots for each machine type are randomly chosen and this slot replaces the old one when performing a mutation.
- (b) Perform crossover (select, crossover and evaluate). Crossover operator selects two individuals from the population using tournament selection method (a number of individuals were evaluated and the best one is chosen). In order to maintain the feasibility, two-point crossover was used where two points between the jobs are randomly chosen and the two-point crossover is performed. Each time after performing crossover, the produced offspring will be verified and repaired to avoid any potential conflicts.

The algorithms perform mutation and crossover for the specified number of generations. Each generation will produce three off springs (two by the crossover and one by the mutation). During the fitness evaluation, the proposed solution is combined with the remaining operations out of the rescheduling horizons from the initial schedule and then evaluated as a whole schedule. If there is an operation which crosses the  $endHorizon$  or finish after the  $endHorizon$ , the consecutive operation after the  $endHorizon$  point will be right-shifted. Then the start time for each operation of the offspring is calculated and the fitness of the offspring is evaluated. If an offspring is fitter than the weakest individual of the current population, then it replaces the weakest.

TABLE III  
AN EXAMPLE OF *ENDHORIZON* DETERMINATION

$O_{n+1,j}$	$P_{n+1,j}$	$Y_b$	$M_k$	dST	dFT	eFT ( $M_k$ )	eFT ( $O_{n+1,j-l}$ )	$B_{n+1,j}^k$	$C_{n+1,j}^k$	Min $C_{n+1,j}^k$	end Horizon
6(1)	3	3	4	15	18	19	0	19	22		
			5	8	11	12	0	12	15	8	
			6	5 <sup>+</sup>	8	5	0	5	8		
6(2)	3	2	3	12	15	18	8	18	21	21	
6(3)	4	1	1	10	14	13	21	21	25	25	
			2	17 <sup>+</sup>	21	17	21	25	25		
7(1)	3	1	1	10	13	14	0	14	17	8	
			2	5 <sup>+</sup>	8	5	0	5	8		
7(2)	4	3	4	15	19	18	8	18	22		
			5	8	12	11	8	11	15	15	
			6	12 <sup>+</sup>	16	12	8	12	16		
7(3)	5	2	3	12	18	15	15	15	20	20	

provided data     
 calculated data

\* The variety paired of shapes reflect the derivation of the values

### III. RESULTS AND DISCUSSION

The rescheduling methodology is used to solve the rescheduling problem. Three datasets of producing two, four and six models of products are used for the initial schedules with three different saturation levels, respectively. The scheduling planning horizon is four months. Saturation (*sat*) level introduced by Moratori *et al.* [11] is used to measure the idle time in the shop floor. The *sat* level is defined as the ratio between the mean of collected idle times on machines and the makespan of the initial schedule. The insertion time, *insTime* = {*begin*, *middle*, *end*} refers to a *startHorizon* (arrival time of the new order) that is equal to 20%, 50% and 80% of the makespan of the initial schedule, respectively.

The efficiency and stability measures for each tested parameter on each datasets are reported in Table IV. Elaboration of the results in Figure 4, 5, 6, 7 and 8 were explained in the next paragraph.

As can be clearly seen from Figure 4 the satisfaction grade of all objectives, *S* for normal orders is better than rush orders. This is expected since the rush orders push the new jobs to be performed first and cause the lower priority jobs to be delayed. The greater value of *S* means that the customer satisfaction grade of new schedules is high and vice versa. As shown in Figure 4, generally in most cases, the higher the *sat* level of the initial schedule, the higher the satisfaction grade of *S* of the new schedule obtained. In the meantime, Figure 4 also reveals that the satisfaction of *S* becomes higher from the earlier *insTime* to the later *insTime*, except for the two cases of the medium and the high *sat* level of schedule for rush orders. It is believed that the differences in these two cases exist due to the decreasing of the mean

flow time,  $\Omega$  at the end of *insTime* for all *sat* levels of schedule for rush orders as appeared in Figure 8.

Surprisingly, as presented in Figure 5, there is no significant effect to the stability, *Q* of normal and rush orders. The high stability, *Q* means that the new schedule has only a small changes as compared to the old one and vice versa. After a great deal of observation and evaluation, non-reschedule-based strategy definitely contributes to the stability of the schedule. The stability, *Q* of the new schedule is 100% achieved when the rescheduling takes place at the end and sometimes in the middle of *insTime* for all *sat* levels of the schedule.

Figure 6 demonstrates the satisfaction level of tardiness, *F* are bad whenever rescheduling occurs in the beginning and at the middle of a highly saturated schedule (low *sat* level) for both types of orders. As expected, the admittance of new orders particularly the rush one to the highly saturated schedule may cause the dragging of lower priority jobs and lead to the tardiness of some jobs. However, the satisfactions are increase to the medium and the low saturated schedule (high *sat* level). The high satisfaction of *F* means that there are a small number of tardy jobs and zero value of *F* means that there are more than 50% of jobs that are tardy.

The aim of  $\Xi$  in pull scheduling is to hold excessive inventory by avoiding early finished jobs. Figure 7 exhibits that the satisfaction of earliness,  $\Xi$  of normal orders are slightly higher than rush orders. The high value of  $\Xi$  means that almost all jobs are finished just on the due dates that relatively decrease the size of the company's inventory. In pull manufacturing, the schedule is pulled forward from downstream to set the due date that the job is expected to finish, to upstream to determine the release date. The downstream is the work area in which one passes the product

to when he has finished a part, while the upstream is the work area from which one receives a product to work on.

There is a significant difference in  $\Omega$  for normal and rush orders. As expected, the trade-off between the lower and the higher priority jobs contributes to the variability of the mean flow time satisfaction level,  $\Omega$  for normal orders and extremely for rush orders. The high  $\Omega$  means that there is a small amount of WIP in the shop floor as the aim of  $\Omega$  is to reduce the waiting time between the operations. Figure 8 demonstrates that the fairly high value of  $\Omega$  for normal orders because the flow time of the lower priority jobs is expanded in order to admit the higher priority jobs.

From the overall observations, it can be concluded that from the most eminent measures of all satisfaction grades,  $S$  comes from the highest sat level of initial schedule for normal orders with  $S = 0.839$  to  $0.862$  and the worst of  $S$  is from the lowest sat level of the initial schedule for rush orders with  $S = 0.342$  to  $0.500$ . These findings make sense because definitely the highest *sat* level of initial schedule (low saturated schedule) that contains large amount of idle times may easily allow the new jobs to be accommodated compared to the one with a lower *sat* level. However, on the other hand the highest *sat* level of the initial schedule is not an efficient schedule with respect to machine utilisation as it contains large amount of machine idle times. Therefore, it can be summarised that a low saturated initial schedule may result in high satisfaction grade of new schedule after rescheduling, while a highly saturated initial schedule may produce a reasonable quality of new schedule whenever new orders arrive using the non-reshuffle-based genetic match-up algorithms.

#### IV. CONCLUSIONS

The development of science and technology can be obtained among others by a wide variety of media available [32]. This industry is categorized by an increasing demand for diversity and increasing demands on product development time because of an increasing customer needs [33]. This work investigates a pull paradigm FMS rescheduling problem. The scheduling problem is considered dynamic since new orders may arrive day-to-day and need to be incorporated into the current schedule. The non-reshuffle-based genetic match-up algorithms that modify only a part of the schedule are implemented in order to allow in new coming jobs. The efficiency and stability of the schedules that are produced by the rescheduling algorithms with different insertion time on the different levels of schedule saturation were investigated and compared. The experiments have shown that the satisfaction grades of the efficiency and stability of normal orders are higher than rush orders. Based on the experimental results obtained, it can be concluded that the non-reshuffle rescheduling strategy can be expected to maintain the efficiency and stability of the new schedule when new orders arrive.

#### ACKNOWLEDGMENT

This work is funded by the National Science Fund under Grant 79105 and the Bumiputra Academic Training Scheme (SLAB) from Public Service Department of Malaysia (JPA). The authors would like to thank the aforesaid parties for the grants. The authors' utmost appreciations are also extended to ASSB and Zilun Systems Sdn. Bhd. for the provided data.

TABLE IV  
EXPERIMENTAL RESULTS OF THE TESTED PROBLEM

<i>orderType</i>	<i>sat</i>	<i>insTime</i>	All Objectives ( <i>S</i> )	Stability ( <i>Q</i> )	Tardiness ( <i>F</i> )	Earliness ( <i>E</i> )	Mean Flow Time ( $\Omega$ )
Normal	Low	Begin	0.542	0.935	0.249	0.969	0.751
		Middle	0.603	0.957	0.362	0.988	0.838
		End	0.658	1.000	0.424	0.995	1.000
	Medium	Begin	0.707	0.957	0.598	0.980	0.966
		Middle	0.751	1.000	0.671	0.994	1.000
		End	0.789	1.000	0.774	0.991	1.000
	High	Begin	0.839	0.978	0.952	0.959	0.941
		Middle	0.857	1.000	0.966	0.961	1.000
		End	0.862	1.000	0.975	0.967	1.000
Rush	Low	Begin	0.342	0.905	0.000	0.927	0.000
		Middle	0.420	0.928	0.000	0.985	0.516
		End	0.500	1.000	0.224	0.993	0.333
	Medium	Begin	0.594	0.923	0.409	0.970	0.707
		Middle	0.657	0.978	0.584	0.993	0.551
		End	0.638	1.000	0.704	0.979	0.016
	High	Begin	0.792	0.946	0.939	0.960	0.669
		Middle	0.806	1.000	0.954	0.954	0.631
		End	0.785	1.000	0.967	0.958	0.422

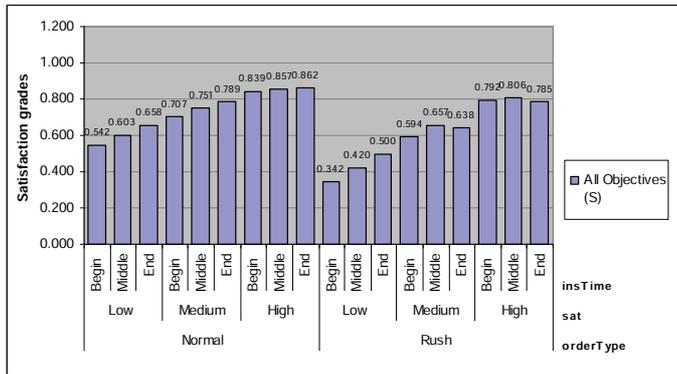


Fig. 4 The satisfaction grades for all objectives ( $S$ ) of both normal and rush orders

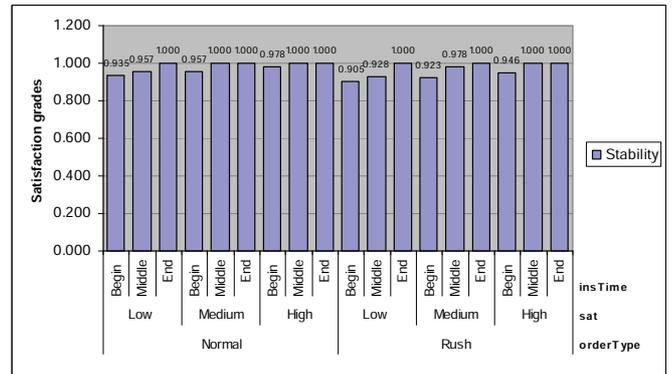


Fig. 5 The satisfaction grades for the tardiness ( $F$ ) of both normal and rush orders

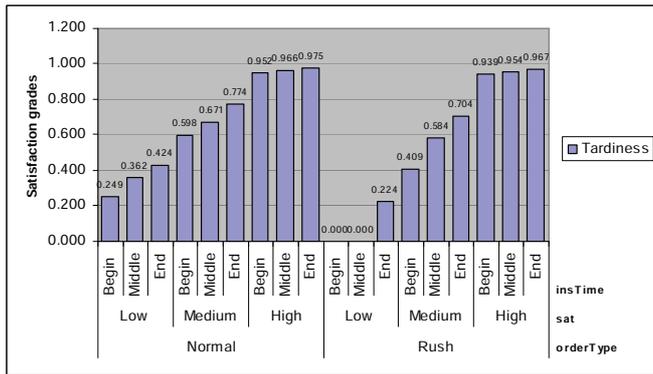


Fig. 6 The satisfaction grades for the stability ( $Q$ ) of both normal and rush orders

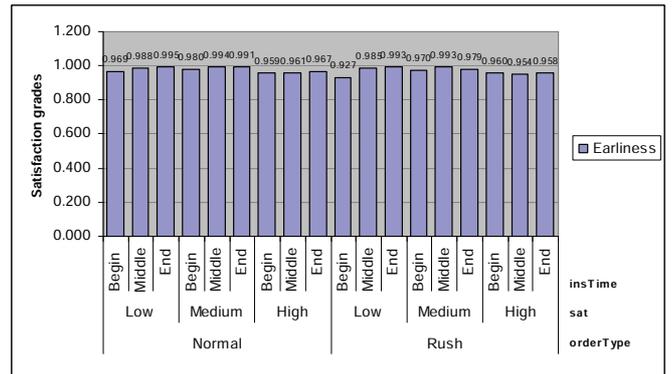


Fig. 7 The satisfaction grades for the earliness ( $E$ ) of both normal and rush orders

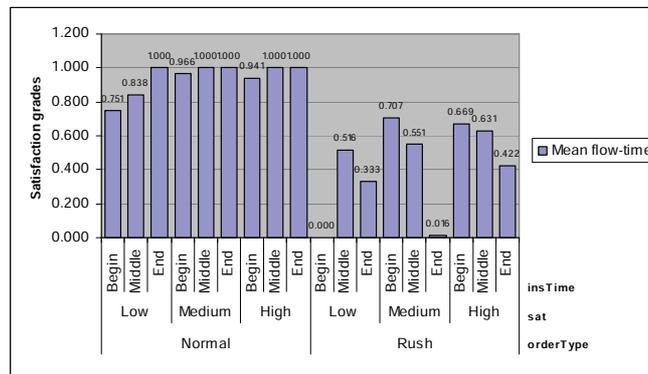


Fig. 8 The satisfaction grades for the mean flow time ( $Q$ ) of both normal and rush orders

## REFERENCES

- [1] P.H. Terry, L.L. Hau, J.N. John, The Practice of Supply Chain Management: where theory and application converge. Boston, MA : Kluwer Academic Publishers, 2003.
- [2] H. Aytug, M.A. Lawley, K. McKay, S. Mohan, R. Uzsoy, Executing production schedules in the face of uncertainties: a review and some future directions. European Journal of Operational Research, 161 (2005) 86-110.
- [3] I. Sabuncuoglu, M. Bayiz, Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research, 126 (2000) 567-586.
- [4] G.E. Vieira, J.W. Herrmann, E. Lin, Rescheduling manufacturing systems: a framework of strategies, policies and methods. Journal of Scheduling, 6 (2003) 35-58.
- [5] A.J. Davenport, C. Gefflot, J.C. Beck, Slack-based techniques for robust schedules. Sixth European Conference on Planning (ECP-2001). Toledo, Spain, 2001.
- [6] M.T. Jensen, Generating robust and flexible job shop schedules using genetic algorithms. IEEE Transactions on Evolutionary Computation, 7 (2003) 275-288.
- [7] H. Yang, Z. Wu, The application of adaptive genetic algorithms in FMS dynamic rescheduling. International Journal of Computer Integrated Manufacturing, 16 (2003) 382-397.
- [8] W. Shi-jin, X. Li-feng, Z. Bing-hai, Filtered-beam-search-based algorithm for dynamic rescheduling in FMS. Robotics and Computer-Integrated Manufacturing, 23 (2007) 457-468.
- [9] J. Bean, J. Birge, J. Mittenthal, C. Noon, Match-up scheduling with multiple resources, release dates and disruptions. Operations Research, 39 (1991) 470-483.
- [10] M.S. Akturk, E. Gorgulu, Match-up scheduling under a machine breakdown. European Journal of Operational Research, 112 (1999) 81-97.

- [11] Moratori, P.B. Petrovic, S. Vázquez, A. Match-up strategies for job shop rescheduling. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, 2008, pp. 119-128.
- [12] E. Szelke, R. Kerr, Knowledge-based reactive scheduling. *Production Planning & Control*, 5 (1994) 124-145.
- [13] G.P. Henning, J. Cerda, An expert system for predictive and reactive scheduling of multiproduct batch plants. *Latin American Applied Research*, 25 (1995) 187-198.
- [14] L. Davis, Job shop scheduling with genetic algorithms, in: J. J. Grefenstette, ed. *Proceedings of the first international conference on genetic algorithms and their applications*. San Mateo: Morgan Kaufmann, 1985, pp. 136-140.
- [15] S. Deris, S. Omatu, H. Ohta, P. Saad, Incorporating constraint propagation in genetic algorithm for university timetable planning. *Journal of the Engineering Application of Artificial Intelligence*, 12 (1999) 241-253.
- [16] A. Rossi, G. Dini, Dynamic scheduling of FMS using a real-time genetic algorithm. *International Journal of Production Research*, 38 (2000) 1-20.
- [17] C. Zhao, Z. Wu, A genetic algorithm approach to the scheduling of FMS with multiple routes. *International Journal of Flexible Manufacturing Systems*, 13 (2001) 71-88.
- [18] S. Petrovic, C. Fayad, A genetic algorithm for job shop scheduling with load balancing. In S. Zhang, R. Jarvis (Eds.). *Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence 3809*. Springer, 2005, pp. 339-348.
- [19] Z.N. Azimi, Comparison of metaheuristic algorithms for examination timetabling problem. *Applied Mathematics and Computation*, 16 (2004) 337-354.
- [20] S. Deris, S. Omatu, H. Ohta, S. Kutar, P.A. Samat, Ship maintenance scheduling by genetic algorithm and constraint-based reasoning. *European Journal of Operational Research*. 112 (1999) 489-502.
- [21] E.K. Burke, A.J. Smith, Hybrid evolutionary techniques for the maintenance scheduling problem. *IEEE Transaction Power System*, 15 (2000) 122-128.
- [22] U. Aickelin, K. Dowsland, An indirect genetic algorithm for a nurse scheduling problem. *Computers and Operations Research*, 31 (2004) 761-778.
- [23] C.F. Tsai, K.M. Chao, An effective chromosome representation for optimising product quality, *IEEE International Conference on Computer Supported Cooperative Work in Design*, 2007, pp. 1032-1037.
- [24] J. Birge, M. Dempster, Optimal match-up strategies in stochastic scheduling. *Discrete Applied Mathematics*, 57 (1995) 105-120.
- [25] M.A. Gamila, S. Motavalli, A modeling technique for loading and scheduling problems in FMS. *Robotics and Computer-Integrated Manufacturing*, 19 (2003) 45-54.
- [26] A. Duenas, D.Petrovic, An approach to predictive-reactive scheduling of parallel machines subject to disruptions. *Annals of Operations Research*, 159 (2008) 65-82.
- [27] M.L. Pinedo, *Planning and scheduling in manufacturing and services*, 1st ed. New York: Springer, 2005.
- [28] L. Li, D. J. Fonseca, D. S. Chen, Earliness-tardiness production planning for just-in-time manufacturing: A unifying approach by goal programming. *European Journal of Operational Research*, 175 (2006) 508-515.
- [29] D.J. Hoiomt, P.B. Luh, K.R. Pattipati, A practical approach to job-shop scheduling problems. *IEEE Transactions on Robotics and Automation*, 9 (1993) 1-13.
- [30] H. Ishibuchi, T. Murata, Flowshop scheduling with fuzzy due date and fuzzy processing time, in: R. Slowinski, M. Hapke, eds. *Scheduling under fuzziness*, Heidelberg, Germany: Physica-Verlag, 6(1), 2000, pp. 113-143.
- [31] S. Petrovic, C. Fayad, D. Petrovic, E.K. Burke, G. Kendall, i, *Annals of Operations Research*, 159 (2008) 275-292.
- [32] Veronice V, Yelfiarita Y, Darnetti D. Analysis of Characteristics Extension Workers to Utilization of Information and Communication Technology. *International Journal on Advanced Science, Engineering and Information Technology*. 2015 Aug 30;5(4):303-5.
- [33] Ibrahim Z, Mohamed SB, Minhat M, Mohamed AS, Musanih MR, Rahman ZA, Rizman ZI. Reason Maintenance in Product Modelling via Open Source CAD System. *International Journal on Advanced Science, Engineering and Information Technology*. 2016 Dec 27;6(6):990-6.