Heuristics for Scheduling Hybrid Flow Shop with Time Windows

Chettha Chamnanlor and Kanchana Sethanan

***Abstract*—This paper focuses on the hybrid flow shop with time windows constraint. The objective of this research is to minimize the makespan. Time window constraints are often found in the hard drive industries requirement for controlling production time on the manufacturing shop floor. In this hybrid flow shop environment there are *S* production stages, each of which may have more than one unrelated machine and batch processing. In this paper, three heuristics which included a basic heuristic and two improved heuristics were developed to solve several problem sets. To access the performance of the heuristics developed, the solutions in terms of makespan, mean flow time, and number of lot- loss were compared with the best heuristic solution. The computation results proved that the heuristics are efficient.**

***Index Terms*—Hybrid flow shop, time windows, batch processing, heuristics**

I. Introduction

In today's industries, production is often found to rely on the hybrid flow shop, especially in the hard disk drive and electronic parts industries. The industrial process usually involves *S* stages (workstations), each of which consists of at least one machine. The machines in each stage may be identical, uniform, or unrelated. Each job is produced from Stage 1, Stage 2, and other stages; hence, it can be said that the hybrid flow shop is a generalization of the flow shop and the parallel processor environments.

Real production environments include many conditions that are major constraints in the production process. For example, the machines eligibility restriction constraint, where some machines may not be able to carry out all jobs. The constraint may involve batch processing, where a machine may be able to produce more than one job in one batch. Lastly, there is the time windows constraint. This means any job completed outside the time windows specified will lead to loss (i.e., reject lot). For example, production of certain materials that are sensitive to weather conditions, temperatures, or chemicals involved may lead to deterioration or wear during the process when the process requires or uses too much time [1]. This may result in rework or disposal as reject.

When the constraints and conditions listed above are considered in the hybrid flow shop, the problem becomes more complex. Solution by means of exact algorithms such as mathematical models may be intractable in terms of CPU time, or may not be viable at all in real world problems. This is where and why heuristic methods have been introduced to solve such NP problems, for they are able to provide good solutions within a suitable time frame and can be applied in real practical problems.

This research therefore emphasized the development of heuristics to solve scheduling problems in the hybrid flow shop. The unrelated machines, machines eligibility restriction, and time windows were the constraints being taken into account in order to minimize the makespan and loss. In the next section, the results of the review of related literature are presented. The characteristics of the problems and assumptions are described in Section III. This is followed by Section IV presenting the development of three heuristic algorithms for solving the problem. Section V outlines the experimental design and results. Finally, a summary of the main findings is given in Section VI.

II. Literature Review

The classical flow shop problem has long been studied by researchers. S.M. Johnson, a famous developer of algorithms in particular is well known for his two-machine flow shop ($F2//C_{max}$) [2]. Johnson's algorithms have been widely cited up to the present. However, most of the current production systems are in the form of hybrid flow shop characteristics. Therefore, the flow shop on a flexible flow line where at least one machine is considered at each stage is the problem of utmost interest. The problem has gone further into the hybrid flow shop consisting of 2 stages, and in each stage parallel machines are being included ($FFs\ (Pm_1,\ Pm_2)//C_{max}$). Moreover, in certain problems, limitations related to the production environment may have to be taken into account, for example sequence-dependent setup time ($FFs/S_{ipm}/C_{max}$) [3], or unrelated parallel machines.

The hybrid flow shop problems mentioned above have been solved by various methods. Mathematical models yield the optimal solutions if the problem is not too large [3]. Since the said problems belong to the NP problem, heuristic algorithms have come to receive a great deal of attention in this respect. For instance, Gupta et al. have developed an efficient heuristic method for scheduling a two-stage hybrid flow shop with parallel machines at the first stage [4]. Recently, hybrid flexible flow shops with constraints on sequence-dependent setup times and machines availability have been tackled by 5 approaches, including 3 heuristics and 2 meta-heuristics [5]. Additionally in the same year, Yaurima et al. proposed a heuristic and meta-heuristic method to solve hybrid flow shop (HFS) with unrelated machines,

Manuscript received June 8, 2012; revised September 13, 2012.
The authors are with the Industrial Engineering Department, Khon Kaen University, Khon Kaen, 40002 Thailand (e-mail: ch.cham@ hotmail.com; ksethanan@gmail.com).

sequence-dependent setup time (SDST), availability constraints and limited buffer [6].

The literature reviewed shows that production scheduling of hybrid flow shop by considering various constraints and using heuristic methods is of interest to a lot of researchers. However, no research has been performed on developing heuristics for solving the problem by considering time window constraints which is the crucial limitation factor in real problems. Hence, this research aimed to study the prioritization of production on a hybrid flow shop based on time window constraints.

III. Problem Statement

This research was related to prioritization of jobs and scheduling in real production situations of a hard drive industry. The manufacturing shop floor is a hybrid flow shop consisting of *S* production stages. Each stage is composed of $m(k)$ machines ($k = 1,2, \ldots, S$) for production of a total of *n* jobs (lots). The jobs were divided into 6 product groups in 2 family groups. Each job was provided with at least one machine that was able to produce that job and within any length of time each machine was able to produce the maximum of jobs not exceeding the batch processing of that machine.

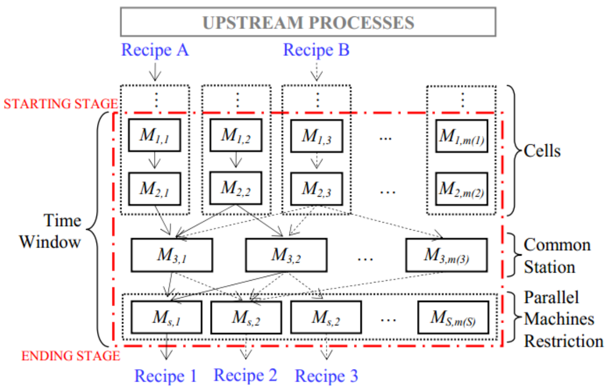

Fig. 1. Hybrid flow scheduling within time windows.

In addition, machines layout configuration of some stages is a permutation flow line, whereas in certain stages, it is a parallel common machine or parallel restricted machine. Moreover, the machine speed of each stage differs from the others. For instance, the speeds of some machines are uniform in some stages while in some other stages the machines' speeds are unrelated, which means the speed of each machine relies on the type of products to be produced.

Further still, decision of jobs or selection of machines to produce the jobs had to meet the time window constraint identified in some processes (shown in Fig. 1), with the aim of to minimizing the makespan.

A. Assumptions

The assumptions made in formulating this problem are as follows: (1) it is assumed that the decisions have been made from long and intermediate-range planning. (2) There are no due dates associated with products since the production was made-for-stock. (3) All jobs and machines are available at the beginning of the scheduling process. (4) Some stages of the hybrid flow shop production may have several unrelated machines. (5) Jobs can wait between two production stages and the intermediate storage is unlimited. (6) No preemption is allowed for any job. (7) The setup times are sequence independent. Finally, (8) the production time controlling of products is considered.

IV. Heuristic Algorithms

Three heuristic algorithms are proposed from this research work. The heuristics have been developed for solving the scheduling hybrid flow shop problem when time windows are considered. The objective is to minimize the makespan and to minimize the number of rejected jobs. The three developed heuristics consist of (1) Match HFS-recipes algorithm, (2) Pull late job first algorithm, and (3) Modified Johnson's algorithm. Parameters used in developing the 3 heuristics are shown below.

A. Notation of Parameters

i,i',i''	Product indices
j,j'	Machine indices
k	Stage indices
r,r'	Batch processing indices
N	The number of products
$m(k)$	The number of machines in stage *k*
S	The number of stages
Tw	The periods of production time controlling
ss	The starting stage for production time control
es	The ending stage for production time control
$B(r,j,k)$	The set of jobs in batch processing *r* on machine *j* of stage *k*
$F_{r,j,k}$	Finish time of jobs in batch processing *r* on machine *j* of stage *k*
$P_{i,j,k}$	Processing time of product *i* on machine *j* of stage *k*
$C_{i,j,k}$	Completion time of product *i* on machine *j* of stage *k*

B. Match HFS-Recipes Algorithm (MHFS)

This algorithm is developed in order to achieve smooth production flow by means of matching between product recipes at the first stage and product recipes at the last stage. The detailed procedure for this algorithm is given below:

Phase 1: Sequencing bottleneck (*BN*) stage first

Step 1: Determine the *BN* stage:

Set $BN = k$ when $\min_{k=1,..,S} \{capacities\ of\ stage\ k\}$

Step 2: Sort the machines for *BN* stage using the Least Flexible Machines rule

Step 3: Group the jobs according to the eligibility restriction of each machine

Step 4: Schedule each job in each group from Step 3 for each machine in *BN* stage using the Shortest Processing Time rule

Phase 2: Match recipes of *BN* stage to recipes of the 1st stage

Step 1: Transform the solution from *BN* stage above into string under recipes of the 1st stage

Step 2: Schedule each job for each machine in the 1st stage based on the Balancing Machines rule

Step 3: Calculate the completion time of each job in each batch processing of each machine in the 1st stage

Set $C_{ij1} \geq F_{r'j1} + P_{ij1}$; $r' = r - 1$ and $C_{ij1} \geq \max_{i' \in B(r,j,1),\, i \neq i'} \{C_{i'j1}\}$; $j \in m(1)$

If $r' \geq 1$, $F_{r'j1} = \max_{i'' \in B(r',j,1)} \{C_{i''j1}\}$; otherwise, $F_{r'j1} = 0$

Phase 3: Sequencing of other stages by Balancing Machines

Step 1: Sort jobs according to their completion time in the previous stage based on the Earliest Completion Time rule

Step 2: Schedule each job to each machine in other stages by allocating equal workload and considering the condition of machine configuration for the jobs as follows:

1) If the present stage is the permutation flow line, the order of the job in the present stage will be the same as the order in the previous stage.
2) If the present stage is under parallel common machines, the order of the jobs in the present stage will be scheduled by the LPT rule.
3) If the present stage is under single common machines, the order of the jobs in the present stage will be scheduled by the SPT rule.

Step 3: Calculate the completion time of each job in each batch process of each machine in the present stage.

Set $C_{ijk} \geq \max_{i \in N,\, r' = r-1} \{C_{ij(k-1)} + P_{ijk} , F_{r'jk} + P_{ijk}\}$ and $C_{ijk} \geq \max_{i' \in B(r,j,k),\, i \neq i'} \{C_{i'jk}\}$; $j \in m(k)$

If $r' \geq 1$, $F_{r'jk} = \max_{i'' \in B(r',j,k)} \{C_{i''jk}\}$; otherwise, $F_{r'jk} = 0$

Phase 4: Setting the makespan and computing the loss

Step 1: Set the makespan from the completion time of the last job completed in the last stage

Step 2: Calculate the loss from spending more time on completing any job than the time window.

If $Tw < \{C_{ij'k(es)} - C_{ijk(ss)}\}$, the product i is set to become the lost-lot.

C. Pull Late Job First Algorithm (PLJF)

The concept of the *PLJF* algorithm is developed for reduction of loss by taking the lost-lot obtained from the *MHFS* algorithm to be re-scheduled first. The procedure of *PLJF* algorithm is as follows:

Phase 1: Provide a set of the lost-lot numbers (*LLS*)

$LLS \Leftarrow$ Take the lost-lot numbers from *MHFS* algorithm and add them to the set by distributing into each machine

Phase 2: Sequence the 1st stage by *PLJF* rule

Step 1: Delete all lots in the 1st stage solution of *MHFS* algorithm which are the lost-lot numbers

Step 2: Insert all lots in the 1st stage solution of *MHFS* algorithm with lots from *LLS*

Step 3: Calculate the completion time of each job in each batch process of the machine in the 1st stage

D. Modified Johnson's Algorithm (MOJO)

The concept of Johnson was adapted appropriately for the developed algorithm. Johnson's rule is a classical theory which is well known for solving the *F2//Cmax* problem. The principle of scheduling jobs to machines yields well-balanced work even though the problem of this research was more complicated than the classical problem. The algorithm could be used effectively. The steps of the *MOJO* algorithm are as follows:

Phase 1: Construct the decision filter string (*DFS*)

DFS $\Leftarrow$ Order the sum of production time in all stages for each descending job

Phase 2: Sequence the 1st stage by *MOJO* rule

Step 1: Group recipes by sorting *DFS* according to recipe groups

Step 2: Group machines by sorting the jobs in each recipe groups according to the machine groups that can produce the recipes

Step 3: Schedule work for each machine by ordering jobs according to *DFS* from maximum to minimum when the order of the machine in the recipe is an odd number, and ordering jobs according to *DFS* from minimum to maximum when the order of the machine in the recipe is an even number.

Step 4: Calculate the completion time of each job in each batch process of each machine in the 1st stage

Carry out Phase 3 and Phase 4 in the same way as the steps in the *MHFS* algorithm.

V. Experimental Design and Computational Results

Two types of data characteristics were presented for each set, and 99 test problems were generated for all data types. The parameters for each data type were randomly selected from different uniform distributions as show in Table I.

The three heuristic algorithms were coded in MatLab and run on a 2.27 GHz PC, with 2 G-Byte of RAM, for testing and evaluation. The solutions to be tested and evaluated are (1) the makespan (C_{max}), (2) the mean flow time ($\bar{F}$), and (3) the number of lot-loss (*NLL*). The results of the average values obtained from the computations are presented in Table I. The quality of a solution generated by the heuristics is measured in terms of performance (*HP*), as computed from equation (1).

$$HP(\%) = 100 \times (LB / Heu_a) \quad (1)$$

When Heu_a = The solution obtained from Heuristic a (a = *MHFS, PLJF*, and *MOJO*),

LB = The Lower bound solution.

From the 99 sample problems, the best makespan was derived from 10, 58, 31 test problems of *MHFS*, *PLJF*, *MOJO*, respectively. The best mean flow time was derived from 6, 27, 66 test problems of *MHFS*, *PLJF*, *MOJO*, respectively. The best number of lot-loss was obtained from 11, 41, 47 sample problems of *MHFS*, *PLJF*, *MOJO*, respectively. Based on the results from equation (1), the performance of the *PLJF* and *MOJO* are better than *MHFS* since they yield the best solutions from the test problems.

Therefore, the research compared the Improving Heuristic (*IH*) by taking the best solutions of the *PLJF* and *MOJO* as the best solution computed from equation (2):

$$IH(\%) = 100 \times \{(Heu_{MHFS} - Heu_b) / Heu_{MHFS}\} \quad (2)$$

When Heu_b = value obtained from computing by Heuristic b (b = *PLJF, MOJO*).

TABLE I: VALUES OF PARAMETERS WITH THE DIFFERENT DATA TYPES AND SOLUTION WITH AVERAGE VALUES.

Data Set	Data Type		Solution with average values								
			MHFS			*PLJF*			*MOJO*		
	ProcTime	Lot Size(1)	C_{max}	$\bar{F}$	*NLL*	C_{max}	$\bar{F}$	*NLL*	C_{max}	$\bar{F}$	*NLL*
Set 1: EX	Unif(1,10)	20	27.7	15.2	7.7	22.9	14.1	4.3	24.4	13.9	4.7
		50	51.9	20.1	9.9	45.6	16.5	1	43.3	16.2	0.6
		120	105.4	30.4	17.6	92.9	22.2	0.3	87.9	21.2	0
	Unif(8,20)	20	85.8	49.5	11.7	73.6	44.4	7	72.3	42.5	5.3
		50	152.9	60.7	25.4	128.1	50	10.3	129.3	47.8	7.3
		120	298.2	80.7	45.9	263.2	60	4.2	267.2	59.9	10.9
	Unif(16,30)	20	118.1	71.6	7.3	112.4	68.3	3.1	117.2	68.6	3.2
		50	209.7	91.4	12.8	195.2	79.6	1.7	210	75.2	0.4
		120	480	120.3	39	385.6	100.5	16.7	428.9	92.2	6.6
Set 2: RW	Real Industrial Time	120	229.7	117.7	14.6	209.4	107.1	6.2	209.9	102.7	4.4
		500	765.4	338.3	254.6	693.1	266.6	163.1	692.9	245.1	128.6

Note: (1) Each lot size consists of two recipes (i.e; recipe#A and recipe#B). The quantities of each recipe are randomly generated as in the ratios 70:30, 50:50, and 30:70, and 3 test problems were generated for each ratio.

The result of variance testing (ANOVA) at the confident interval of 95% showed that the two heuristic algorithms (*PLJF*, *MOJO*) could significantly improve the solutions better than the *MHFS* method. From the multiple rank test based on the LSD approach, it was shown that the 13 machine problems (EX set), ratios (recipe#A: recipe#B) of various types significantly affected solutions by the two heuristics, except the number of lot-loss from *MOJO*. At the same time, different sets of production time (ProcTime) showed significant effects on solutions by the two heuristics, except the number of lot-loss from *PLJF*. In addition, different lot sizes had significant results on solutions by two heuristic methods, except the makespan from *PLJF* and *MOJO*.

The problems with 35 machines (RW set) and ratios (recipe#A: recipe#B) were found to be significant to solutions by two heuristics except the number of lot-loss from *MOJO* and makespan from *PLJF* and *MOJO*, while different lot sizes showed significant results on solutions by two heuristics except the makespan from the two heuristics.

VI. CONCLUSION

In this research work, three heuristics were developed for solving the scheduling of hybrid flow shop problems with the consideration of production time control. Various test problems were brought to experiment on the efficiency of the heuristics. The experimental results proved that *MOJO* yielded the best solutions among the three methods. *PLJF* and *MOJO* still could significantly improve solutions from *MHFS*. Moreover, the three methods showed that the computational time for the three heuristic algorithms are extremely small-less than 22 seconds. These times do not significantly increase with the size of the problem. This means that the three algorithms are very efficient, and more importantly are not sensitive to the problem size. Nevertheless, meta-heuristics will be used to find the best solutions for future research in order to increase the efficiency of the presented method.

ACKNOWLEDGMENT

This project was financially supported by the Industry/University Cooperative Research Center (I/UCRC) in HDD Component, the Faculty of Engineering, Khon Kaen University and National Electronics and Computer Technology Center, National Science and Technology Development Agency.

REFERENCES

[1] C. Chamnanlor and K. Sethanan, “Mixed integer programming models for scheduling hybrid flowshop with unrelated machines and time windows constraints,” in *Proc. 10th APIEMS Conf.*, 2009, pp. 2331-2335.

[2] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Englewood Cliff, New Jersey: Prentice-Hall, 1982.

[3] K. Sethanan, “Scheduling flexible flowshops with sequence dependent setup time,” Ph. D. dissertation, West Virginia Univ., West Virginia, 2001.

[4] J. N. D. Gupta, A. M. A. Hariri, and C. N. Potts, “Scheduling a two-stage hybrid flow shop with parallel machines at the first stage,” *Annals of Operation Research*, vol. 69, pp. 171-191, 1997.

[5] F. Jabbarizadeh, M. Zandieh, and D. Talebi, “Hybrid flexible flowshops with sequence-dependent setup times and machine availability constraints,” *Computers & Industrial Engineering*, in press, 2009.

[6] V. Yaurima, L. Burtseva, and A. Tchernykh, “Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers,” *Computer and Industrial Engineering*, vol. 56, pp. 1452-1463. 2009.

Chettha Chamnanlor was born in Bangkok, Thailand, in 1977. He is Ph.D. student in the Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand. He received a master’s degree from the Graduate School of Engineering at Khon Kaen University, Thailand in 2005. His research interests include logistic management and applied operations research. His email address is <ch.cham@hotmail.com>

Anchana Sethanan was born in Roi-Ed, Thailand, in 1969. She is a Lecturer in the Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand. She received a Ph.D. in Industrial Engineering from West Virginia University, USA in 2001. Her research interests include logistic and supply chain management, scheduling and applied operations research. Her email address is <ksethanan@gmail.com>