PAPER • OPEN ACCESS

Batch Scheduling for Hybrid Assembly Differentiation Flow Shop to Minimize Total Actual Flow Time

To cite this article: R Maulidya *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **319** 012042

Batch Scheduling for Hybrid Assembly Differentiation Flow Shop to Minimize Total Actual Flow Time

R Maulidya, Suprayogi, R Wangsaputra and A H Halim

Industrial Engineering Faculty, Institut Teknologi Bandung
JL. Ganeca 10 Bandung, 40132, Indonesia

rahmimaulidya@gmail.com, yogi@mail.ti.itb.ac.id,
rachmawati_wangsaputra@yahoo.com, ahakimhalim@mail.ti.itb.ac.id

Abstract. A hybrid assembly differentiation flow shop is a three-stage flow shop consisting of Machining, Assembly and Differentiation Stages and producing different types of products. In the machining stage, parts are processed in batches on different (unrelated) machines. In the assembly stage, each part of the different parts is assembled into an assembly product. Finally, the assembled products will further be processed into different types of final products in the differentiation stage. In this paper, we develop a batch scheduling model for a hybrid assembly differentiation flow shop to minimize the total actual flow time defined as the total times part spent in the shop floor from the arrival times until its due date. We also proposed a heuristic algorithm for solving the problems. The proposed algorithm is tested using a set of hypothetic data. The solution shows that the algorithm can solve the problems effectively.

1. Introduction

In this paper, we study the three-stage production system consists of machining, assembly and differentiation stage called *Hybrid Assembly Differentiation Flow Shop*. The system is producing at least two types of products where every type of product consists of three parts. The first stage is the machining stage and has several parallel unrelated machines to produce each part of product types. The second stage is an assembly operation where all parts of a product is assembled into an assembly part. Each of the assembled parts is delivered to the final process at the differentiation stage. In the differentiation stage, an assembled part is further processed to become a certain type of final product. This system can be found in an electronic industry which has three stages production system and produces several types of products.

Recent research conducted by [1] and [2] has developed the system for job scheduling problem. [1] develop a forward scheduling model with the criteria of total flow time while [2] develop a backward scheduling model with the criteria of total actual flow time. In these research, the setup time is neglected. According to [3], the independent setup is an anticipatory setup for changing tool or cleaning the machine. These setup activity can be shared if jobs have the same characteristic. A reduction in total setup time can be achieved by grouping the jobs into batches [4].

The reason for batching jobs is to gain efficiency with the maximal set of jobs [3]. In job processor, jobs in a batch is processed sequentially so that the processing time of a batch is equal to the sum of the processing time of its jobs [5]. In some literatures, a batch consists of identical processing time of jobs [5-8] but others have develop the non-identical processing time of jobs in a batch [9-10]. The decision of batching and scheduling can be conducted sequentially [5-6],[8],[10] or simultaneously [7].

Some literatures develop the objective of scheduling problem such as minimizations of the makespan [9-10], the maximum lateness [11], the total completion time [13], the total flow time [1] and the multiple objective such as minimization of the makespan and maximum lateness and minimizations of the total flow time and sum of weighted tardiness [12]. These objectives of scheduling problem are completion time based and suitable for forward scheduling. The objective suits to backward scheduling should be due date based. The objective introduced by [5] define the total actual flow time as the interval time of parts in the shop floor started from the arrival time of parts until they finished at their due date. This objectives ensure that the completed product can be delivered to the costumer at their due date and we can control the arrival of parts at the shop at the time when the processes is started. In this paper, we adopt the total actual flow time as the objective of scheduling problem.

This paper deals with a problem of batch scheduling model in a hybrid assembly differentiation flow shop to minimize the total actual flow time. This paper is organized as follows. In Section 2, we formulate our problem into a non-integer linear programming model. In Section 3, we proposed an heuristic algorithm to solve the problem. In order to evaluate the performance of the algorithm, we conduct some computational experiments in Section 4 and we present our conclusions in Section 5.

2. Problem Formulation

There are J jobs (J=J_1,J_2,…,J_J) to be processed and are divided into two specific type of jobs JH_h (JH_1=J_1,J_2,J_3,J_4 and JH_2=J_5,J_6,J_7,J_8). All jobs must be processed in three stages i.e. the machining, assembly and differentiation stage. All jobs in a batch are processed one by one on the machine with its non-identical processing time of job. Each job J has k components (Part#1,…,Part#K) to be processed on k different parallel unrelated machines in the machining stage. When all jobs in a batch are completed, the batch will be transferred to the assembly stage. There is a setup time between the processed of two consecutive batches. In the assembly stage, the parts of a job is assembled. The group of jobs for the three stages production system is in the same order. In the differentiation stage, each of assembled part is finally processed to become a type of finished products. At the end of their common due date, all batches of finished product will be ready to be delivered to customer. Figure 1 shows the system of batch scheduling.

The problem is to determine the number of batches and the batch sizes and to define the production schedule for the three stages production system so that it can minimize the total actual flow time. The following assumptions are considered in the paper.

1. Productions are finished at their common due date.
2. Each batch only consists of a similar type of jobs.
3. The setup operation can be started before the batch arrives on the machine.
4. Transportation times are neglected.
5. No interruption is allowed until all the jobs in a batch are completed.
6. There are unlimited buffers between the stage one and two and the stage two and three

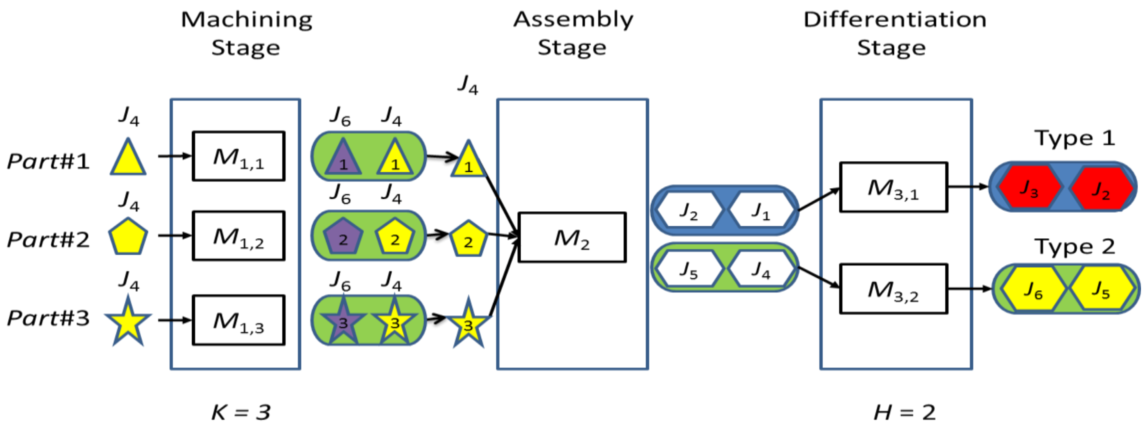

Figure 1. The three stages production system

Halim et.al. [5] define the actual flow time of a job as the time that the job spends in the shop from its starting time for processing until its due date. The actual flow time, $F_{[u]}^{a}$ can be written as follows.

$$F_{[u]}^{a} = d - B_{[u]}, \qquad u=1,2,\ldots,U \tag{1}$$

For batch scheduling problems, the actual flow time of a batch is similar to equation (1). The differentiation is that a batch consists of parts that wait in the batch until all parts in the batch are completed. The actual flow time can be written as follows.

$$F_{[u]}^{a} = \left(d - B_{[u]}\right) Q_{[u]} \tag{2}$$

The machining stage in this paper consists of three parallel machines. So the actual flow time of a batch defines as due date minus the starting time of a batch *u* in the machining stage ($\mathrm{B}_{[u]}^{(1)}$), times the batch sizes of batch *u*. The formulation is as follows.

$$F_{[u]}^{a} = \left(d - \min\left(B_{[u,k]}^{(1)}\right)\right) Q_{[u]} \tag{3}$$

The formulation of the total actual flow time (*TAFT*) is:

$$TAFT = \sum_{\mathrm{u}=1}^{\mathrm{U}} \left(d - \min_{\mathrm{k}\le\mathrm{K}}\left(\mathrm{B}_{[\mathrm{u}],\mathrm{k}}^{(1)}\right)\right) Q_{[u]} \tag{4}$$

Notation for this model is introduced below:

Indexes and parameters

j	index of job, j=1, 2,…,J
u_m	index of a batch at stage m, $u_m = 1, \ldots, U_m$, m=1,..,3.
k	index of a machine in the machining stage, k=1, 2,...,K
h	index of a machine in the differentiation stage, and also as the index of product type, h=1, 2,…,H
JH_h	number of processed job at machine h
d	common due date
ϕ	a large number
$M_{1,k}$	machine for processing part k of all the job in the machining stage, k = 1, 2,…,K
M_2	assembly machine in the assembly stage
$M_{3,h}$	dedicated machine for processing product type h in the differentiation stage, h = 1, 2,…,H
$p_{j,k}^{(1)}$	processing time of job J_j on machine $M_{1,k}$ at the machining stage
$p_{j}^{(2)}$	assembly time of job J_j on assembly machine M_2
$p_{j,h}^{(3)}$	processing time of job J_j on machine $M_{3,h}$ at the differentiation stage, $p_{j,h}^{(3)} > 0$ if $j \in n_h$; 0, otherwise
$A_{j,h}$	binary variable for processing *job j* on machine h, $A_{j,h} = 1$ if $p_{j,h}^{(3)} > 0$; $A_{j,h} = 0$ if $p_{j,h}^{(3)} = 0$.

Decision variables

$W_{j[u_m]}^{(m)}$	binary variable, $W_{j[u_m]}^{(m)} = 1$ if job j is assigned to batch u ; 0, otherwise..
$Q_{[u_m]}^{(m)}$	batch size at stage m
$QQ_{[u_m],h}^{(m)}$	binary variable for processing batch u_m on machine h, $QQ_{[u],h} = 1$ if batch u_m at stage m is assigned to machine h; 0, otherwise.
$\mathrm{B}_{[u_1],\mathrm{k}}^{(1)}$	starting time of processing batch u_1 on machine k in the machining stage (stage 1)
$\mathrm{B}_{[u_2]}^{(2)}$	starting time of assembly operation for batch u_2 in the assembly stage (stage 2)
$\mathrm{B}_{[u_3],\mathrm{h}}^{(3)}$	starting time of processing batch u_3 on machine h in the differentiation stage (stage 3)

The batch scheduling for hybrid assembly differentiation flow shop model can be formulated as follows.

Minimize $TAFT = \sum_{u_1=1}^{U_1}\left(d - \min_{1\le k\le 3}\left(B_{[u_1],k}^{(1)}\right)\right)Q_{[u_1]}^{(1)}$ (5)

Subject to:

$$B_{[u_3],h}^{(3)} = QQ_{[u_3],h}^{(3)}\left(d - \sum_{z_3=1}^{u_3}\left(QQ_{[z_3],h}^{(3)}\sum_{j=1}^{J}\left(W_{j[z_3]}^{(3)}.p_{j,h}^{(3)}\right)+S_h\right)-S_h\right), z_3 \le u_3, \forall u_3, \forall h \quad (6)$$

$$B_{[u_2]}^{(2)} + \left(\sum_{j=1}^{J}\left(W_{j[u_2]}^{(2)}.p_j^{(2)}\right)\right) \le \sum_{h=1}^{H} B_{[u_3],h}^{(3)}\frac{X_{[u_2],[u_3]}}{\sum_{j=1}^{J}\left(W_{j[u_2]}^{(2)}.W_{j[u_3]}^{(3)}\right)} + \left(1 - \frac{X_{[u_2],[u_3]}}{\sum_{j=1}^{J}\left(W_{j[u_2]}^{(2)}.W_{j[u_3]}^{(3)}\right)}\right)\phi, \quad \forall u_2, \forall u_3, \forall h \quad (7)$$

$$B_{[u_2+1]}^{(2)} + \left(\sum_{j=1}^{J}\left(W_{j[u_2+1]}^{(2)}.p_j^{(2)}\right)\right) + \sum_{h=1}^{H}\left(QQ_{[u_2],h}^{(2)}.S_h\right) \le B_{[u_2]}^{(2)}, \quad u_2 < U_2, \forall u_2 \quad (8)$$

$$B_{[u_1],k}^{(1)} + \left(\sum_{j=1}^{J}\left(W_{j[u_1]}^{(1)}.p_{j,k}^{(1)}\right)\right) \le B_{[u_2]}^{(2)}\frac{X_{[u_1],[u_2]}}{\sum_{j=1}^{J}\left(W_{j[u_1]}^{(1)}.W_{j[u_2]}^{(2)}\right)} + \left(1 - \frac{X_{[u_1],[u_2]}}{\sum_{j=1}^{J}\left(W_{j[u_1]}^{(1)}.W_{j[u_2]}^{(2)}\right)}\right)\phi, \quad \forall u_1, \forall u_2, \forall k \quad (9)$$

$$B_{[u_1+1]}^{(1)} + \left(\sum_{j=1}^{J}\left(W_{j[u_1+1]}^{(1)}.p_{j,k}^{(1)}\right)\right) + \sum_{h=1}^{H}\left(QQ_{[u_1],h}^{(1)}.S_h\right) \le B_{[u_1],k}^{(1)}, \quad u_1 < U_1, \forall u_1, \forall k \quad (10)$$

$$A_{j,h} = \begin{cases} 1, & \text{jika } p_{j,h}^{(3)} > 0 \\ 0, & \text{jika } p_{j,h}^{(3)} = 0 \end{cases} \quad j = 1,2..J, \forall h \quad (11)$$

$$\sum_{h=1}^{H}\sum_{u_m=1}^{U_m}\left(Q_{[u_m]}^{(m)}.QQ_{[u_m],h}^{(m)}\right) = J, \quad (12)$$

$$\sum_{u_m=1}^{U_m}\left(W_{j[u_m]}^{(m)} * QQ_{[u_m],h}^{(m)}\right) = A_{j,h}, \quad \forall j, \forall h \quad (13)$$

$$\sum_{h=1}^{H} QQ_{[u_m],h}^{(m)} = 1, \quad \forall u_m \quad (14)$$

$$\sum_{u_m=1}^{U_m} W_{j[u_m]}^{(m)} = 1, \quad \forall j \quad (15)$$

$$\sum_{j=1}^{J} W_{j[u_m]}^{(m)} = Q_{[u_m]}^{(1)}, \quad \forall u_m, \forall j \quad (16)$$

$$X_{[u_m],[u_{m+1}]} = \sum_{j=1}^{J}\left(W_{j[u_m]}^{(m)}.W_{j[u_{m+1}]}^{(m+1)}\right), \quad \forall u_m, \forall j \quad (17)$$

$$QQ_{[u_m],h}^{(m)} \in \{0,1\}, \quad \forall u_m, \forall h \quad (18)$$

$$W_{j[u_m]}^{(m)} \in \{0,1\}, \quad \forall u_m, \forall j \quad (19)$$

$$Q_{[u_m]}^{(m)} > 0, \quad \forall u_m \quad (20)$$

$$B_{[u_1],k}^{(1)}, B_{[u_2]}^{(2)}, B_{[u_3],h}^{(3)} \ge 0, \quad \forall u_1, \forall u_2, \forall u_3, \forall h \quad (21)$$

Equation (5) shows the total actual flow time where the actual flow time of each batch is the longest interval between the parts being processed in the machining stage until its common due date. Constraint (6) show that the completion time of every batch processed at the differentiation stage should exactly coincide with the due date. Constraint (7) show that the completion time of batch processed at the assembly stage is less than the starting time of batch processed at the differentiation stage. Constraint (9) show that the completion time of every batch at the machining stage is less than the starting time of the batch at the assembly stage. Constraint (8) and (10) ensures that each batch can only be processed at the same stage after the next position is finished. Constraint (11) define the job that is processed in machine *h* at the differentiation stage has the processing time greater than zero. Constraint (12), (13) and (14) ensure batch u contained only the same type of jobs. Constraint (15) ensure that a job is only assigned to one batch. Constraint (16) accomplishes a material balance in the shop. Constraint (17) defines the batching relation for job *j* that belong to a certain batch. Constraint (18), (19) and (21) define the domain of the decision variables. Constraint (20) ensure that the batch size is greater than zero.

IOP Conf. Series: Materials Science and Engineering **319** (2018) 012042 doi:10.1088/1757-899X/319/1/012042

3. Algorithm

The problem is divided into three subproblems, i.e. batching, integrating job and batch, and scheduling. In solving the first problem we define the number of batches and the batch sizes based on [5]. Each stage has a different number of batches and also the batch sizes. So, equation (22) until (25) determine the batching for each stage, except for machining stage there are alternatives to be selected as the batching solution. The second subproblem is solved by sequencing the jobs with SPT based heuristic. The job sequence is grouped into batches and a batch is set only for the same type of product. The job sequence is optimized by variable neighborhood search (Insert move and swap move). The third subproblem is solved by examining the resulting batches to minimize the total actual flow time. The stopping rule is the condition where the iterations give the minimum total actual flow time.

In order to determine the number of batches and the batch sizes, we use equation (22) to (25) based on Halim et.al. (1994) as follows.

$$Q_{[u]} = \frac{J}{N} + \frac{\left(\frac{s}{t}\right).(N+1)}{2} - \left(\frac{s}{t}\right)u \qquad u=1,\ldots,N \qquad (22)$$

$$N = \frac{-Z}{s} + \left(\frac{1}{s}\right)\sqrt{Z^2 + 2Jst} \qquad (23)$$

Where:

$$Z = \frac{1}{2}Jt + \frac{Js}{\frac{d}{t}+\frac{s}{t}-J} - \frac{1}{2}d - \frac{1}{2}s \qquad (24)$$

$$N_{max} = \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{2Jt}{s}} \qquad (25)$$

Variable Neighbourhood Search in this paper consists of two neighborhood operators i.e. Insert and Swap move [1]. The Insert move is the process of moving the job in position u to position v, whereas all jobs in position k, with k=u+1,...,v, are shifted one position forward. The swap move is the process of swapping the job in position u and the job in position v. Figure 2 represent the insert and swap move.

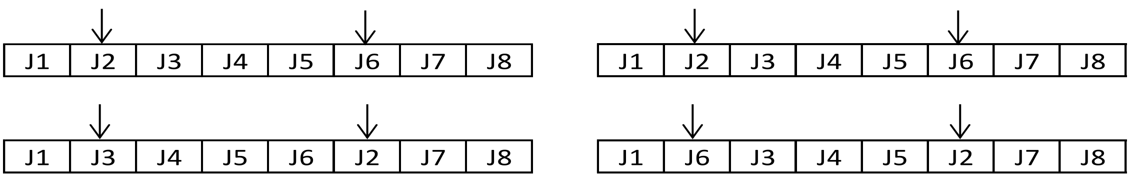

Figure 2. The insert move (left) and swap move (right) for u=2 and v=6

The complete algorithm is as follows.

Step 0: Initialize the problem

Step 1: Define the number of batches and the batch sizes for each machine using equation (22) through (25) following the steps from [5].

Step 2: Define six of job sequences based on SPT based heuristic [1]. Grouped a job sequence into batches of the same type of product and define the total actual flow time. Set the minimum TAFT as the initial solution.

Step 3: Set the maximum iteration. Perform procedure VNS on the initial solution as follows.

- Randomly choose two positions u and v, where $u<v$.
- Move the job in position u to position v, whereas all jobs in position k, with $k = u+1,\ldots,v$, are shifted one position forward along solution x. Then a new solution will be x'.
- If TAFT (x') < TAFT (x), then $x \leftarrow x'$. else
- swap the job in position u and the job in position v of solution x. Then a new solution will be x'.
- If TAFT (x') < TAFT (x), then $x \leftarrow x'$. end.
- Repeat this procedure until the maximum iteration.

Step 4: Is the solution better than the best initial solution?
If Yes, output the best solution
If No, go to step 3.

4. Numerical Experience

We use hypothetic data for analyzing the model. The data are: The number of jobs processed (J) is 8 jobs, the common due date (d) is 1000, The setup time for type h (S_h) are 3 and 5, respectively. Table 1 shows the processing time of 2 product types (Type 1 and Type 2). Type 1 consists of jobs J_1, J_2, J_3 and J_4, written as $JH_1=\{J_1,J_2,J_3,J_4\}$ and Type 2 consists of jobs J_5, J_6 ,J_7 and J_8 written as $JH_2=\{J_5,J_6,J_7,J_8\}$. This illustration is for the number of jobs (J)=8, the number of machines in the first stage (K)=3, and the number of machines in the third stage (H)=2. We use processor of Intel Core i7-6500U CPU, 2,50 GHz and 12 GB RAM to run all data. Table 2 represents the resulting number of batches and the batch sizes of the algorithm from step 1.

Table 1. Processing time

Job	Type	$p_{j,1}^{(1)}$	$p_{j,2}^{(1)}$	$p_{j,3}^{(1)}$	$p_j^{(2)}$	$p_{j,1}^{(3)}$	$p_{j,2}^{(3)}$
J_1	1	5	4	3	3	6	0
J_2	1	6	3	4	9	10	0
J_3	1	6	4	10	3	9	0
J_4	1	3	5	5	4	5	0
J_5	2	3	4	5	6	0	8
J_6	2	3	4	6	4	0	7
J_7	2	10	4	6	5	0	6
J_8	2	5	4	3	10	0	4

The number of batches and the batch sizes for every stage can be seen in table 2. $M_{1,1}$, $M_{1,2}$ and $M_{1,3}$ is the unrelated parallel machine and the batching must be selected from N=5 and N=6 so there are no parts left behind in the assembly stage. The resulting batch proceeds to the assembly stage (M_2).

Table 2. The number of batches and the batch sizes

	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	M_2	$M_{3,1}$	$M_{3,2}$
N_{max}	6	5	6	6	6	4
$Q_{[1]}$	3	4	3	3	1	2
$Q_{[2]}$	2	3	2	2	1	1
$Q_{[3]}$	2	2	2	2	1	1
$Q_{[4]}$	1	1	1	1	1	0
$Q_{[5]}$	1	0	1	1	0	0
$Q_{[6]}$	0	0	0	0	0	0

The initial solution with the minimum $TAFT$=555 is S2 with the resulting batch as follows.

$M_{1,k}$	M_2	$M_{3,1}$	$M_{3,2}$
$Q_{[1]}=J_1,J_3,J_4$	$Q_{[1]}=J_1,J_3,J_4$	$Q_{[1]}=J_1$	$Q_{[1]}=J_6,J_7$
$Q_{[2]}=J_6,J_7$	$Q_{[2]}=J_6,J_7$	$Q_{[2]}=J_3$	$Q_{[2]}=J_5$
$Q_{[3]}=J_5,J_8$	$Q_{[3]}=J_5,J_8$	$Q_{[3]}=J_4$	$Q_{[3]}=J_8$
$Q_{[4]}=J_2$	$Q_{[4]}=J_2$	$Q_{[4]}=J_2$	

Table 3. Variable Neighbourhood Search

u	v	Job sequence	$M_{1,k}$	M_2	$M_{3,1}$	$M_{3,2}$	TAFT
-	-	$J_1,J_3,J_4,J_6,J_7,J_5,J_2,J_8$	$Q_{[1]}=J_1,J_3,J_4$	$Q_{[1]}=J_1,J_3,J_4$	$Q_{[1]}=J_1$	$Q_{[1]}=J_6,J_7$	555
			$Q_{[2]}=J_6,J_7$	$Q_{[2]}=J_6,J_7$	$Q_{[2]}=J_3$	$Q_{[2]}=J_5$	
			$Q_{[3]}=J_5,J_8$	$Q_{[3]}=J_5,J_8$	$Q_{[3]}=J_4$	$Q_{[3]}=J_8$	
			$Q_{[4]}=J_2$	$Q_{[4]}=J_2$	$Q_{[4]}=J_2$		
Insert Move							

2	5	$J_1, J_4, J_6, J_7, J_3, J_5, J_2, J_8$	$Q_{[1]} = J_1, J_4, J_3$	$Q_{[1]} = J_1, J_4, J_3$	$Q_{[1]} = J_1$	$Q_{[1]} = J_6, J_7$	555
			$Q_{[2]} = J_6, J_7$	$Q_{[2]} = J_6, J_7$	$Q_{[2]} = J_3$	$Q_{[2]} = J_5$	
			$Q_{[3]} = J_5, J_8$	$Q_{[3]} = J_5, J_8$	$Q_{[3]} = J_4$	$Q_{[3]} = J_8$	
			$Q_{[4]} = J_2$	$Q_{[4]} = J_2$	$Q_{[4]} = J_2$		
Swap Move							
2	5	$J_1, J_7, J_4, J_6, J_3, J_5, J_2, J_8$	$Q_{[1]} = J_1, J_4, J_3$	$Q_{[1]} = J_1, J_4, J_3$	$Q_{[1]} = J_1$	$Q_{[1]} = J_7, J_6$	555
			$Q_{[2]} = J_7, J_6$	$Q_{[2]} = J_7, J_6$	$Q_{[2]} = J_3$	$Q_{[2]} = J_5$	
			$Q_{[3]} = J_5, J_8$	$Q_{[3]} = J_5, J_8$	$Q_{[3]} = J_4$	$Q_{[3]} = J_8$	
			$Q_{[4]} = J_2$	$Q_{[4]} = J_2$	$Q_{[4]} = J_2$		

The algorithm is set for 50 iterations and there is no new solution. The resulting batch for stage 1 and stage 2 is the same. It means that the part processed in the machining stage will be grouped with the same number of batch and the batch sizes. Figure 3 below shows the Gantt chart for the resulting batch.

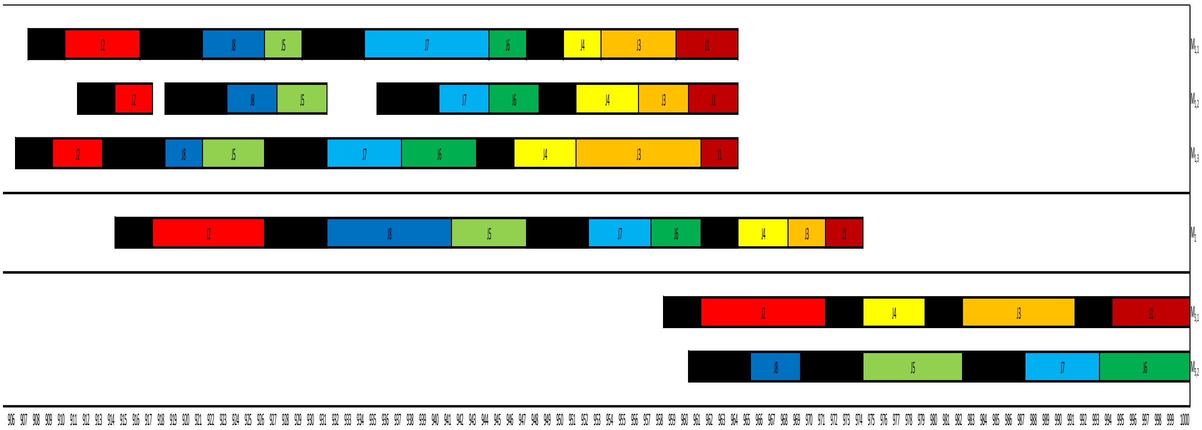

Figure 3. Gantt Chart for *TAFT*=555

5. Conclusions

This paper deals with the batch scheduling problems for a Hybrid Assembly Differentiation Flow Shop, processing *H* different types of products to minimize the total actual flow time. This research has shown that the non-identical processing time of the same type of jobs can be batched with unnecessarily equal batch sizes. The numerical example of this approach is limited to 8 jobs with the scenario that each stage produces a different number of batches. The proposed algorithm consists of batching, integrating jobs into batches, and scheduling. The final schedule shows that the batches are sequenced from the shortest processing time. But it still needs to be observed how batches are built. The future research is to find the proposed algorithm that can give an optimal solution in short time.

References

[1] Xiong F, Xing K and Wang F 2015 Scheduling a Hybrid Assembly-Differentiation Flow Shop to Minimize Total Flow Time. *European Journal of Operation Research* **240** 338-54

[2] Maulidya R, Suprayogi, Wangsaputra R and Halim A H 2016 Job Scheduling for Hybrid Assembly Differentiation Flowshop to Minimize Total Actual Flow Time. *Proc. of 17th Asia Pacific Industrial Engineering and Management System Conference* (*APIEMS*) *Conf.* December 7-10, 2016 (Taipei) 1198-1204

[3] Potts C N and Kovalyov M Y 2000 Scheduling with Batching: A Review. *European Journal of Operational Research* **120** 228–49

[4] Kovalyov M Y, Potts C N and Strusevich V A 2004 Batching decisions for assembly production systems, *European Journal of Operational Research* **157** 620–642

[5] Halim A H, Miyazaki S and Ohta H 1994a Batch-Scheduling Problems to Minimize Actual Flow Times of Parts Through The Shop under JIT Environment. *European Journal of Operation Research* **72** 529-44

[6] Halim A H, Miyazaki S and Ohta H 1994b Lot Scheduling Problems of Multiple Items in the shop with Both Receiving and Delivery Just in Times. *Production Planning and Control* **5(2)** 175-184

[7] Halim A H, Suryadhini P P and Toha I S 2006 Batch Scheduling to Minimize Total Actual Flow Time in a Two-stage Flow Shop with Dedicated Machines in the Second Stage, *Proc. of 7th Asia Pasific Industrial Engineering & Management System (APIEMS) Conf.* December 17-20, 2006 (Bangkok) 1343-49

[8] Halim A H and Yusriski R 2009 Batch Scheduling for the Two-stage Assembly Model to Minimize Total Actual Flow Time. *Proc. of 7th Asia Pasific Industrial Engineering & Management System (APIEMS) Conf. 2009* (Kitakyushu)

[9] Huang T and Lin B M T 2013 Batch Scheduling in Differentiation Flow Shops for Makespan Minimization, *International Journal of Production Research* **51(17)** 5073-82

[10] Shen L and Gupta J N D 2017 Family scheduling with batch availability in flow shops to minimize makespan, *Journal Scheduling*, 1-15 (New York: Springer)

[11] Hariri A M A and Potts C N 1997 Single Scheduling with Batch Set-up Times to Minimize Maximum Lateness, *Annals of Operation Research* **70** 75-92

[12] Nikzad F, Rezaeian J, Mahdavi I and Rastgar I 2015 Scheduling of Multi-component Products in a Two-stage Flexible Flow Shop. *Applied Soft Computing* **32** 132-43

[13] Cetinkaya F C and Kayaligil M S 1992 Unit Sized Transfer Batch Scheduling with Setup Times. *Computers Industrial Engineering* **22(2)** 177-83