Job Scheduling for Hybrid Assembly Differentiation Flowshop to Minimize Total Actual Flow Time

Rahmi Maulidya†, **Rachmawati Wangsaputra, Suprayogi, Abdul Hakim Halim**
Faculty of Industrial Technology
Institut Teknologi Bandung (ITB), Indonesia
Tel: (+062) 22-2504189, Email: rahmimaulidya@gmail.com,
rachmawati_wangsaputra@yahoo.com, yogi@mail.ti.itb.ac.id, ahakimhalim@gmail.com

Abstract. This paper deals with scheduling problems of a hybrid assembly differentiation flow shop consisting of parts manufacturing, assembly operation and differentiation stages. In the first stage, a number of different parts of a job are processed on (unrelated) dedicated machines, and then, in the second stage, the completed parts are assembled on a common machine to produce common jobs. Finally, in the third stage consisting dedicated machines, each particullar of the jobs is processed on a dedicated machine. The problem is to find a job schedule so as to minimize total actual flow time to accommodate that parts should arrive in the shop at the right times and in the right quantities, and that the finished product should be delivered at their due dates. A Mixed Integer Linear Programming (MILP) model is proposed to represent the problem and it can be solved efficiently for small size problems. However, for larger-size problems, an approach of heuristic algorithms (SPT-based and NEH-based) for generating an initial solution, and meta-heuristic algorithm (Hybrid Genetic Algorithm-Variable Neighborhood Search) for obtaining the final solution is proposed. The approach is tested using a set of hypothetic data, and it can be shown that the algorithm can solve the problems effectively.

Keywords: job scheduling, hybrid assembly differentiation flow shop, total actual flow time

1. INTRODUCTION

It can be observed that there are three stages in a doorline of car manufacturing, i.e., machining, assembly and painting in this order. In the machining stage, door parts are manufactured. Those parts are assembled into a body in the door line for the next step to produce variety of finished products. The stage resulting variety of finished products is called as a differentiation stage (Lin and Hwang, 2011, Xiong et al., 2015). Cases for differentiation as the finishing stage can be found in painting process (Xiong et al., 2015), and cutting process of sticker label (Lin and Liao, 2003). A differentiation stage will result different types of finished products coming from the output of assembly operation (Xiong et al., 2015). Xiong et al. (2015) have discussed scheduling problems with three stages manufacturing to minimize total flow time and called a hybrid assembly differentiation flow shops. The research includes a fabrication stage with three parallel machines, assembly stage with a single assembly machine, and differentiation stage with two dedicated machines each of which produces a type of products.

This research is about an arrangement of machines called hybrid flowshop which flows sequencially with a different number of parallel machines in every stage (Ruiz and Vazquez-Rodriguez, 2010). Researchers develop the hybrid flowshop scheduling problem based on variation of shop configuration, constraint, assumption and objective function (Ruiz and Vazquez-Rodriguez, 2010).

The hybrid flow shop could be related to the differentiation stage when producing different type of products and the shop is arranged due to the product type. According to Wang and Liu (2013), hybrid flow shop with differentiation stage has several names, such as two-stage hybrid flow shop with dedicated machines (Lin and Liao, 2003, Wang and Liu, 2013), two-stage differentiation flowshop (Lin and Hwang, 2011) and two-stage hybrid flow shop with machine eligibility.

The hybrid flow shop could also be related to assembly operation, known as hybrid assembly flow shops, where parts from the fabrication stage are assembled into an assembly part at the assembly stage (Fattahi et al., 2014). The assembly operation can be arranged in assembly shop or assembly line with conveyor (Morton and Pentico, 1993).

Operators of the assembly operation do their job manually or using tools or operating semi automated machine (Groover, 2008). According to a product structure, research has developed on one assembly level (Fattahi et al., 2014, Halim and Yusriski, 2009) and two or more assembly levels (Thiagarajan and Rajendran, 2005).

A hybrid flowshop that is related to differentiation stage and assembly stage is known as a hybrid assembly differentiation flow shop. It is introduced by Xiong et.al (2015). The research addresses job scheduling for three production stages of the hybrid assembly differentiation flow shop model where the first stage is component manufacturing, the second stage is an assembly operation and the third stage is a differentiation. The approach was the forward scheduling with the objective function of minimizing total flow time.

Literature shows that there is another approach called the backward scheduling in which jobs are scheduled from the due date to the time zero direction. Much research on backward scheduling (Halim, 1993, Halim et.al., 2006) has adopted the criteria of total actual flow time (Halim and Yusriski, 2009, Halim and Ohta, 1994) is the criteria of the interval time of parts in the shop floor. The total actual flow time is defined as the total minimizing interval time between respective product's arrival times and their common due date. The actual flow time assumes that the product arrives at the shop floor when needed and the completed product is delivered to the costumer at their due date. A part does not have to be available at time zero but may arrive at the shop at the time of manufacturing. Therefore, the resulting schedule decisions give impact to the inventory level and part receiving time at the production line (Halim, 1993).

In this paper, we develop a mathematical model and an algorithm for hybrid assembly differentiation flow shop to minimize total actual flow time. Section 2 describes and formulates the model consideration. Section 3 shows the algorithm as a solution for the problem. Section 4 shows the numerical examples and the result for the problem. Section 5 concludes the paper and gives suggestions for future research.

2. PROBLEM FORMULATION

The problem in this paper is developed for three stages production system, i.e. machining, assembly and differentiation stage, each of which has different number of machines. The system produces G types of products that is similar to the number of machines at the third stage. Figure 1 shows the system we represent. In the first stage, a number of different parts of a job are processed on (unrelated) dedicated machines. Parts are processed on each machine in the same order. In the second stage, the completed parts are assembled on a common machine to produce common jobs. In the third stage consisting dedicated machines, each particullar of the jobs is processed on a dedicated machine to become a certain type of product. All jobs are requested at a common due date d.

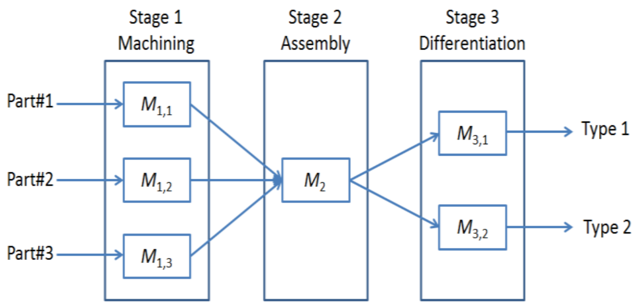


Figure 1. Three stages production system

The problem is to find a job schedule so as to minimize total actual flow time to accommodate that parts should arrive in the shop at the right times in the right quantities, and the finished product should be delivered at their due dates.

Several assumptions are made as follows:

- Production are finished at their common due date
- Each machine can process at most one job at a time
- Each job can be processed on at most one machine at a time
- Setup times and transportation times are neglected
- Job processing cannot be preempted before it is finished
- There are unlimited buffers between the machines of the stage one and two and the stage two and three

Halim et al. (1994) define 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, $F^a_{[i]}$, can be formulated as follows:

$$F^a_{[i]} = d - B_{[i]}, \qquad i=1,2,\ldots,n \qquad (1)$$

Where d is a common due date for n jobs and $B_{[i]}$ is the starting time for processing $J_{[i]}$. The jobs are scheduled backwardly so the position i is counted from the due date.

Let us modify the formulation of the actual flow time of a job for the problem concerned in this paper.

$$F^a_{[i]} = d - \min_{k\le K}\left(B^{(1)}_{[i],k}\right) \qquad (2)$$

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

$$TAFT = \sum_{i=1}^{n}\left(d - \min_{k\le K}\left(B^{(1)}_{[i],k}\right)\right) \qquad (3)$$

This paper adopts the following notations:

Index

j = index of job, j=1, 2,…,n

i = index of job positions, i=1, 2,…,n

k = index of machines at the first stage, k=1, 2,...,K

h = index of machines at the third stage, also state the product type, h=1, 2,...,G

Parameter

K = number of machines at the firt stage

G = number of machines at the third stage

n = number of processed job

n_h = number of job processed in machine h

N = set of processed job

N_h = set of job that is processed in machine h

d = common due date

M_{1k} = machine for processing part k of all the jobs at the first stage, k = 1, 2,…,K

M_2 = assembly machine in the second stage

M_{3h} = differentiation machine for processing types of products in the third stage, h = 1, 2,…,G

$p_{j,k}^{(1)}$ = the processing time of part k job j in machine M_{1k} at the first stage

$p_j^{(2)}$ = the processing time of assembly operation of job j on machine M_2 at the second stage

$p_{j,h}^{(3)}$ = the processing time of job j type h in machine M_{3h} at the third stage. $p_{j,h}^{(3)} > 0$ if $j \in N_h$; 0, otherwise.

$A_{j,h}$ = the binary variable equals 1 if *job j* is assigned in machine h with the processing time $p_{j,h}^{(3)} > 0$; otherwise 0

Decision variables

$X_{j[i]}$ = the binary variable equals 1 if job j is assigned to position i; otherwise 0

$Y_{[i],h}$ = the binary variable for the third stage equals 1 if job in position i is processed in machine h; otherwise 0;

$B_{[i],k}^{(1)}$ = the starting time of job in position i in machine M_{1k} at the first stage

$B_{[i]}^{(2)}$ = the starting time of assembly job in position i in machine M_2 at the second stage

$B_{[i],h}^{(3)}$ = the starting time of job in position i in machine M_{3h} at the third stage

Objective function

$TAFT$ = total actual flow time

The problem can be formulated as follows.

$$\text{Min } TAFT = \sum_{i=1}^{n}\left(d - \min_{k \le K}\left(B_{[i],k}^{(1)}\right)\right) \tag{4}$$

Subject to:

$$B_{[i],h}^{(3)} = Y_{[i],h} * \left(d - \sum_{j=1}^{n}\sum_{i^*=1}^{i}\left(X_{j[i^*]}.p_{j,h}^{(3)}\right)\right), \quad i^* \le i,\ \forall i, \forall h \tag{5}$$

$$B_{[i]}^{(2)} + \sum_{j=1}^{n}\left(X_{j[i]}.p_j^{(2)}\right) \le d - \left(Y_{[i],h} * \sum_{j=1}^{n}\sum_{i^*=1}^{i}\left(X_{j[i^*]}.p_{j,h}^{(3)}\right)\right), i^* \le i, \forall i, \forall h \tag{6}$$

$$B_{[i+1]}^{(2)} \le B_{[i]}^{(2)} - \sum_{j=1}^{n}\left(X_{j[i+1]}.p_j^{(2)}\right), \quad i < n, \forall i \tag{7}$$

$$B_{[i],k}^{(1)} + \sum_{j=1}^{n}\left(X_{j[i]}.p_{j,k}^{(1)}\right) \le B_{[i]}^{(2)} \quad \forall i, \forall k \tag{8}$$

$$B_{[i+1],k}^{(1)} \le B_{[i],k}^{(1)} - \sum_{j=1}^{n}\left(X_{j[i+1]}.p_{j,k}^{(1)}\right), i < n, \forall i, \forall k \tag{9}$$

$$A_{j,h} = \begin{cases} 1, & \text{if} \quad p_{j,h}^{(3)} > 0 \\ 0, & \text{if} \quad p_{j,h}^{(3)} = 0 \end{cases} \quad j = 1,2..n, \forall h \tag{10}$$

$$\sum_{j=1}^{n}\left(X_{j[i]}.A_{j,h}\right) = Y_{[i],h}, \quad \forall i,\ \forall h \tag{11}$$

$$\sum_{j=1}^{n} X_{j[i]} = 1, \quad i = 1,2..n \tag{12}$$

$$\sum_{i=1}^{n} X_{j[i]} = 1, \quad j = 1,2..n \tag{13}$$

$$X_{j[i]} \in \{0,1\} \tag{14}$$

$$B_{[i],k}^{(1)}, B_{[i]}^{(2)}, B_{[i],h}^{(3)} \ge 0, \text{ and integer} \tag{15}$$

The objective function (4) shows the total actual flow time that is calculated from the actual flow time of all position where the actual flow time of each position is the longest interval between the part being processed in the first stage until its common due date. Constraint (5) define the starting time of job in position i in machine h at the third stage from its common due date considering the decision that position i is assigned in machine h. Constraint (6) define the starting time of job in position i at the second stage from backward based on job in position i at the third stage, with condition where $i^* \le i$. Constraint (7) and (9) ensure that each position can only be processed at the same stage after the next position is finished. Constraint (8) define the starting time of job in position i on machine k at the first stage based on the starting time of the same job in position i at the second stage. Constraint (10) define the job with the processing time at the third stage greater than zero, is processed in machine h at the third stage. Constraint (11) decide the job in position i that is assigned on machine h at the third stage only if the value is 1. Constraint (12) and (13) ensure that each position in the sequence of job is assigned to only one job, and each job is assigned only to one position in the sequence of job. Constraint (14) and (15) define the domain of the decision variables.

3. SOLUTION

The decision variable of this model are the starting time of jobs on each stage and the job sequence. Following is the algorithm developed in Xiong et.al (2015) that consists of two heuristics (SPT-based algorithm and NEH-based algorithm) as an an initial solution and the meta-heuristic algorithm (Hybrid Genetic Algorithm-Variable Neighborhood Search) for finding the best solution. In this paper, we modify the algorithm by scheduling the jobs backwardly from the due date to minimize the total actual flow time. The algorithms to determine initial solution for the Hybrid Assembly Differentiation Flow Shop-Total Actual Flow Time are as follows.

SPT-based heuristic

Step 1: Generate six job sequences S_1, S_2, S_3, S_4, S_5, and S_6 in increasing orders of $\max_{1\le k\le 3}\left(p_{j,k}^{(1)}\right), p_j^{(2)}, p_{j,h}^{(3)}$, $\max_{1\le k\le 3}\left(p_{j,k}^{(1)}\right) + p_j^{(2)}, p_j^{(2)} + p_{j,h}^{(3)}$ and $\max_{1\le k\le 3}\left(p_{j,k}^{(1)}\right) + p_j^{(2)} + p_{j,h}^{(3)}$, respectively. Schedule the jobs from due date.

Step 2: select the best one of the six job sequences as the solution. Then output it and its *TAFT* value.

NEH-based heuristic

Step 1: obtain a job sequence S_0 in ascending order of

$$\max_{k=1,2,\dots,K}\left\{p_{j,k}^{(1)}\right\} + p_j^{(2)} + \max_{h=1,2,..G}\left\{p_{j,h}^{(3)}\right\}$$

Step 2: set r=2. Select the first two jobs from S_0 and schedule jobs from due date to minimize the total actual flow time as if there are only two jobs. Set the best one as a current solution S_1.

Step 3: while $r<n+1$ do

Set $r \leftarrow r+1$. Generate r candidate sequence by inserting the r^{th} job in the job sequence S_0 in each slot of the current solution. Select the best one with the least partial total actual flow time. Update the best one as a current solution S_1.

End while

Step 4: output the current solution S_1 and its *TAFT* value.

The best initial solution is selected from the six job sequences of SPT-based and one job sequence of NEH-based heuristic. The best initial solution will be the input of the Hybrid Genetic Algorithm-Variable Neighborhood Search shown as follows.

The Hybrid Genetic Algorithm-Variable Neighborhood Search (HGA-VNS)

Step 1: Input the best initial solution for an instance and set algorithm parameters.

Step 2: Generate Population Size individuals as the initial population. Set maximum runtime t_{max}.

Step 3: Is $t^* > t_{max}$?

If Yes, Output the best solution

If No, go to step 4.

Step 4: perform procedure VNSL-I on each individual as follows.

- Input initial solution x and set maximum iteration
- 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 $k = u+1,\dots,v$, 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 5: Find the best individual of the population and improved it by procedure VNSL-II as follows.

- Input initial solution x and set maximum iteration
- Randomly choose two positions u and v, where $u<v$.
- Move the job randomly in position u to position v, whereas all jobs in position k, with $k = u+1,\dots,v$, 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 $x \leftarrow x'$. else
- Inverse the jobs between positions u and v of solution x. Then a new solution will be x'.
- If *TAFT* $(x') <$ *TAFT* (x), then $x \leftarrow x'$. else
- Insert the job in position u and the job in position $u+1$ between position v and $v+1$, whereas all jobs in position k, with $k = u+2,\dots,v$, are shifted two positions forward along 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 6: Perform Genetic algorithm

Step 6.1. Select the offspring to the next generation

Select pairs of strings from the current population.

Step 6.2. Perform crossover operations

Apply two point crossover to selected pairs to generate solution.

Step 6.3. Perform mutation operations

Mutation is performed on single individuals like insert move, swap move and inverse move operation.

Step 6.4. Add the best string to the current population and remove the other string.

Step 6.5. Go to step 6.1. for another pairs of strings until the maximum iteration is achieved.

Step 7: Collect runtime and add the runtime into t^*. Go the step 3.

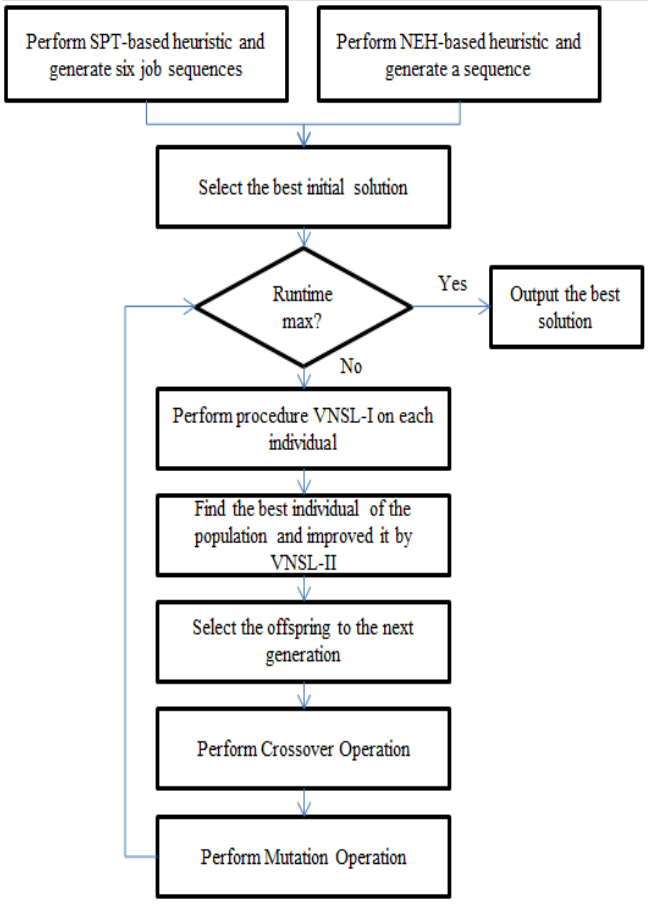


Figure 2. Flowchart of the algorithm

4. NUMERICAL EXPERIENCE

Suppose there is an instance of 8 jobs (n=8) to be processed in the system. 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 $N_1=\{J_1,J_2,J_3,J_4\}$ and type 2 consists of jobs J_5, J_6 ,J_7 and J_8 written as $N_2=\{J_5,J_6,J_7,J_8\}$. This illustration is for the number of jobs (n)=8, the number of machine in the first stage (K)=3, and the number of machine in the third stage (G)=2. The completed products should be delivered at common due date d=1000.

Table 1. Processing time of n=8, K=3, G=2

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

The best initial solution is obtained with the sequence of $J_4,J_8,J_1,J_5,J_3,J_7,J_2,J_6$ and $TAFT$=266 from the SPT-based heuristic algorithm. Then, it is improved with the Hybrid Genetic Algorithm-Variable Neighborhood Search. Table 3. shows the result for the procedure of VNSL-I on step 4.

Table 2. Result of the procedure of VNSL-I on step 4

Insert move				Swap move			
u	v	Sequence	$TAFT$	u	v	Sequence	$TAFT$
-	-	$J_4,J_8,J_1,J_5,J_3,J_7,J_2,J_6$	266				
2	6	$J_4,J_1,J_5,J_3,J_7,J_8,J_2,J_6$	289	3	5	$J_4,J_8,J_3,J_5,J_1,J_7,J_2,J_6$	277
				2	4	$J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$	264
-	-	$J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$	264				
3	5	$J_4,J_5,J_8,J_3,J_1,J_7,J_2,J_6$	270	4	6	$J_4,J_5,J_1,J_7,J_3,J_8,J_2,J_6$	272
				3	5	$J_4,J_5,J_3,J_8,J_1,J_7,J_2,J_6$	274
				2	6	$J_4,J_7,J_1,J_8,J_3,J_5,J_2,J_6$	287
				1	5	$J_3,J_5,J_1,J_8,J_4,J_7,J_2,J_6$	302
				2	7	$J_4,J_2,J_1,J_8,J_3,J_7,J_5,J_6$	341
				2	3	$J_4,J_1,J_5,J_8,J_3,J_7,J_2,J_6$	270
				5	7	$J_4,J_5,J_1,J_8,J_2,J_7,J_3,J_6$	273
				5	8	$J_4,J_5,J_1,J_8,J_6,J_7,J_2,J_3$	267

In Table 3, the best initial solution of $J_4,J_8,J_1,J_5,J_3,J_7,J_2,J_6$ is become the initial solution x and the first insert move operation is moving the job in position u=2 (J_8) to position v=6. The result of sequence is $J_4,J_1,J_5,J_3,J_7,J_8,J_2,J_6$ with $TAFT$=289. Because the solution x is not improved, then the next procedure is swaping the job in position u=3 (J_1) and position v=5 (J_3) and the result of sequence is $J_4,J_8,J_3,J_5,J_1,J_7,J_2,J_6$ with $TAFT$=277. The best result is in the sequence of $J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$ with $TAFT$=264 from step 4.

Next step is procedure of VNSL-II on step 5. The procedure of insert move and swap move are the same as in step 5. For the procedure of inverse move, the jobs are inversed between position u=3 and position v=7. The result of sequence is $J_4,J_5,J_2,J_7,J_3,J_8,J_1,J_6$ with $TAFT$=317. For the procedure of Or-Opt move, the jobs are shifted two positions forward along solution x. The job in position u=3 and u+1=4 are inserted between position v=6 and v+1=7. The result of sequence is $J_4,J_5,J_3,J_7,J_2,J_1,J_8,J_6$ with $TAFT$=285. Table 4 shows the result for step 5 and the best individual population is the sequence of

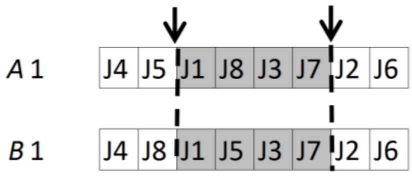
$J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$ with $TAFT$=264.

In step 6.1, offspring are generated from two parents. The first parent is the sequence from step 5 with $TAFT$=264 and the second parent is the sequence from the best initial solution with $TAFT$=266. In step 6.2, Crossover is applied to generate offspring by exchanging some genes of the two parents. The first parent is $J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$ and the second parent is $J_4,J_8,J_1,J_5,J_3,J_7,J_2,J_6$. Those are the parent chromosomes for two point crossover.

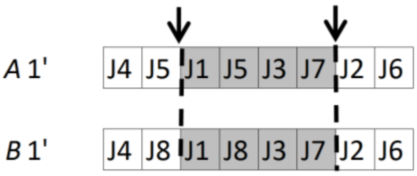
Table 3. Result for step 5 (procedure of VNSL-II)

Insert move				Swap move				Inverse move				Or-Opt move			
u	v	Sequence	$TAFT$	u	v	Sequence	$TAFT$	u	v	Sequence	$TAFT$	u	v	Sequence	$TAFT$
-	-	$J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$	264												
3	7	$J_4,J_5,J_8,J_3,J_7,J_2,J_1,J_6$	277	3	7	$J_4,J_5,J_2,J_8,J_3,J_7,J_1,J_6$	307	3	7	$J_4,J_5,J_2,J_7,J_3,J_8,J_1,J_6$	317	3	7	$J_4,J_5,J_3,J_7,J_2,J_1,J_8,J_6$	285
2	5	$J_4,J_1,J_8,J_3,J_5,J_7,J_2,J_6$	275	3	4	$J_4,J_5,J_8,J_1,J_3,J_7,J_2,J_6$	265	2	5	$J_4,J_7,J_3,J_8,J_1,J_5,J_2,J_6$	301	2	5	$J_4,J_8,J_3,J_7,J_5,J_1,J_2,J_6$	291
3	5	$J_4,J_5,J_8,J_3,J_1,J_7,J_2,J_6$	270	2	5	$J_4,J_3,J_1,J_8,J_5,J_7,J_2,J_6$	301	2	7	$J_4,J_2,J_7,J_3,J_8,J_1,J_5,J_6$	361	4	8	$J_4,J_5,J_1,J_2,J_6,J_8,J_3,J_7$	309
2	6	$J_4,J_1,J_8,J_3,J_7,J_5,J_2,J_6$	282	2	7	$J_4,J_2,J_1,J_8,J_3,J_7,J_5,J_6$	341	1	3	$J_1,J_5,J_4,J_8,J_3,J_7,J_2,J_6$	268	2	7	$J_4,J_8,J_3,J_7,J_2,J_5,J_1,J_6$	294
4	7	$J_4,J_5,J_1,J_3,J_7,J_2,J_8,J_6$	287	1	4	$J_8,J_5,J_1,J_4,J_3,J_7,J_2,J_6$	270	5	8	$J_4,J_5,J_1,J_8,J_6,J_2,J_7,J_3$	278	1	5	$J_1,J_8,J_3,J_4,J_5,J_7,J_2,J_6$	279
1	5	$J_5,J_1,J_8,J_3,J_4,J_7,J_2,J_6$	272	4	8	$J_4,J_5,J_1,J_6,J_3,J_7,J_2,J_8$	284	4	6	$J_4,J_5,J_1,J_7,J_3,J_8,J_2,J_6$	272	5	8	$J_4,J_5,J_1,J_8,J_2,J_6,J_3,J_7$	278
2	4	$J_4,J_1,J_8,J_5,J_3,J_7,J_2,J_6$	269	2	3	$J_4,J_1,J_5,J_8,J_3,J_7,J_2,J_6$	270	1	5	$J_3,J_8,J_1,J_5,J_4,J_7,J_2,J_6$	306	1	8	$J_1,J_8,J_3,J_7,J_2,J_6,J_4,J_5$	298
1	3	$J_5,J_1,J_4,J_8,J_3,J_7,J_2,J_6$	268	2	4	$J_4,J_8,J_1,J_5,J_3,J_7,J_2,J_6$	266	2	4	$J_4,J_3,J_8,J_1,J_5,J_7,J_2,J_6$	302	2	4	$J_4,J_8,J_3,J_5,J_1,J_7,J_2,J_6$	277
1	5	$J_5,J_1,J_8,J_3,J_4,J_7,J_2,J_6$	272	3	8	$J_4,J_5,J_6,J_8,J_3,J_7,J_2,J_1$	314	1	7	$J_2,J_7,J_3,J_8,J_1,J_5,J_4,J_6$	376	3	8	$J_4,J_5,J_3,J_7,J_2,J_6,J_1,J_8$	294
4	8	$J_4,J_5,J_1,J_3,J_7,J_2,J_6,J_8$	288	5	7	$J_4,J_5,J_1,J_8,J_2,J_7,J_3,J_6$	273	3	5	$J_4,J_5,J_3,J_8,J_1,J_7,J_2,J_6$	274	1	4	$J_1,J_8,J_4,J_5,J_3,J_7,J_2,J_6$	268

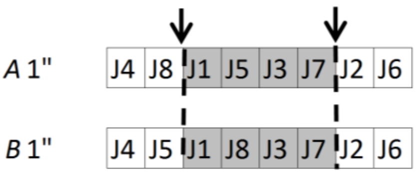
The two point crossover are developed as follows.
The parent chromosome $A1$ and $B1$ are:



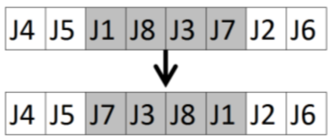
The offspring chromosome $A1$' and $B1$' are:



The repaired offspring chromosome A1” and B1” are:



The best solution for the crossover procedure is $J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$ with $TAFT$=264.
On step 6.3, mutation is performed on single individuals like insert move, swap move and inverse move operation. Example for mutation with inverse move can be seen as follows.



The mutation operation is performed in inversing the jobs between u=3 and v=6, and repeated until the maximum iteration is achieved. The best solution for the mutation procedure is $J_4,J_5,J_1,J_8,J_3,J_7,J_2,J_6$ with $TAFT$=264. Gantt chart for the resulting schedule can be seen in Figure 4.

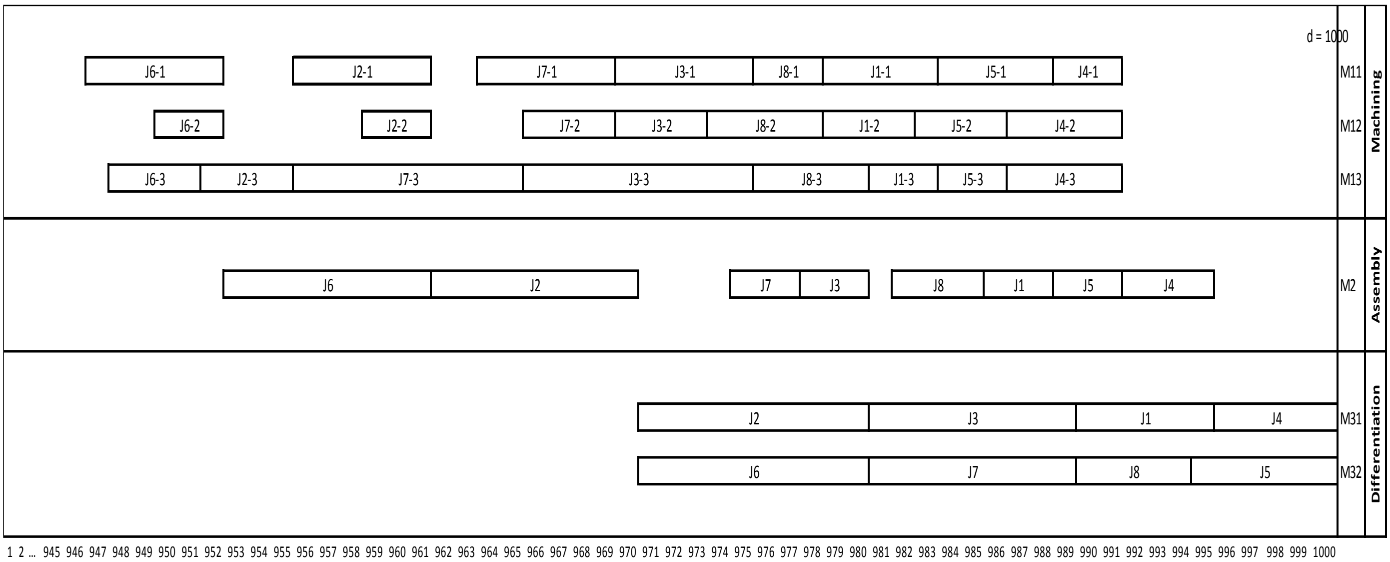


Figure 4. Gantt chart for the resulting schedule of n=8, K=3, G=2

In order to shows the behaviour of the solutions, the numerical experience is done under the combination of number of jobs (n = 4, 6 and 8), number of machines on machining stage (K = 3 and 4), and number of machines on differentiation stage (G = 2, 3, and 4). Total problem sizes is obtained from nxKxG = 3x2x3 = 18 problem sizes. The instance for each number of job is obtained as a part of Table 1. For n=4, the processing time can be seen from the intance of J_1, J_2, J_3 and J_4, and for n=6, the processing time can be seen from the intance of J_1, J_2, J_3, J_4, J_5, and J_6. All combination were run on a PC with Intel Core i7-3770 CPU, 3,10 GHz and 8 GB RAM.

The result of algorithm is compared to optimized model. Table 5 shows the comparison of optimized model and algorithm.

In Table 5, the symbol '*' shows that the result of optimized model is still on local optimal after the runtime is interupted at 100 hours. It is probably because the instance is consisted of duplication data and can not get rid of looping condition. For problem size n=4 and n=6, the algorithm solution can achieve the optimal solution. The result can be compared from the average gap. The SPT-based algorithm is the best initial solution with the average gap 1,9 compared to NEH-based algorithm. The HGA-VNS has the lowest average gap of 0,5. It means that the HGA VNS can effectively give the solution for the medium and large size problem.

*local optimal, due to the completion time limitation

Table 4. Result of optimized model and algorithm

			Model	SPT		NEH		HGA-VNS	
n	*K*	*G*	*TAFT*	*TAFT*	Gap	*TAFT*	Gap	*TAFT*	Gap
4	3	2	91	91	0	91	0	91	0
4	3	3	90	90	0	90	0	90	0
4	3	4	90	90	0	90	0	90	0
4	4	2	91	91	0	91	0	91	0
4	4	3	90	90	0	90	0	90	0
4	4	4	90	90	0	90	0	90	0
6	3	2	165	173	8	173	8	165	0
6	3	3	160	160	0	160	0	160	0
6	3	4	160	160	0	160	0	160	0
6	4	2	165	173	8	173	8	165	0
6	4	3	160	160	0	160	0	160	0
6	4	4	160	160	0	160	0	160	0
8	3	2	257	266	9	270	13	264	7
8	3	3	255*	265	-	263	-	258	-
8	3	4	258*	265	-	263	-	259	-
8	4	2	257*	266	-	270	-	259	-
8	4	3	258*	265	-	263	-	263	-
8	4	4	258*	265	-	263	-	258	-
Average					1.9		2.2		0.5

5. CONCLUDING REMARKS

This paper shows the problem of job scheduling for three stages production system called the hybrid assembly differentiation flow shop, processing *G* different types of products to minimize total actual flow time. The approach for this problem shows that final schedule can be obtained from heuristics algorithms and then finalized by meta-heuristic algorithm. The numerical examples for this approach are limited to 4, 6 and 8 jobs. The number of iterations and the maximum runtime must be set higher to get better result. The illustrations have shown the improvement of the best solution.

The future research is to find the proposed algorithm that can give optimal solution in short time compare to the current approach.

REFERENCES

Fattahi, P., Hosseini, S.M.H., Jolai, F. and Tavakkoli-Moghaddam, R. (2014) A Branch and Bound Algorithm for Hybrid Flow Shop Scheduling Problem with Setup Time and Assembly Operation. *Applied Mathematical Modelling,* **38,** 119-134.

Groover, M.P. (2008) *Automation, Production Systems, and Computer Integrated Manufacturing,* 3rd edition, Pearson International Edition, NJ.

Halim, A.H. (1993) Batch Scheduling for Production System under Just in Time Environment. *Dissertation.* Osaka Perfecture University Jepang.

Halim, A.H. and Ohta, H. (1994) Batch Scheduling Problem to Minimize Inventory Cost in the Shop with Both Receiving and Delivery Just In Times. *International Journal of Productions Economics,* **33,** 185-194.

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.* 17-20 December 2006 (Bangkok), 1343-1349.

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).

Lin, B.M.T. and Hwang, F.J. (2011) Total Completion Time Minimization in a 2-Stage Differentiation Flowshop with Fixed Sequences per Job Type. *Information Processing Letters,* **111,** 208–212.

Lin, H.T. and Liao, C.J. (2003) A Case Study in a Two-stage Hybrid Flowshop with Setup Time and Dedicated Machines. *International Journal of Production Economics,* **86,** 133-143.

Morton, T.E. and Pentico, D.W. (1993) *Heuristic Scheduling System with Application in Production System and Project Management*. John Wiley & sons, New York.

Ruiz, R. and Vazquez-Rodriguez, J.A. (2010) The Hybrid Flow Shop Scheduling Problem. *European Journal of Operation Research,* **205,** 1-18.

Thiagarajan, S. and Rajendran, C. (2005) Scheduling in Dynamic Assembly Job-Shops to Minimize the Sum of Weighted Earliness, Weighted Tardiness, and Weighted Flow Time of Jobs. *Computer & Industrial Engineering,* **49,** 463-503.

Wang, S. and Liu, M. (2013) A heuristic method for two-stage hybrid flow shop with dedicated machines. *Computers & Operations Research,* **40,** 438-450.

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,** 338-354.

BIOGRAPHY

Rahmi Maulidya is a lecturer at Industrial Engineering Department, Trisakti University in Jakarta, Indonesia. She received her undergraduate on Industrial Engineering, Trisakti University in 1998, and her Master degree from Industrial Engineering and Management Institut Teknologi Bandung (ITB) in 2005. She is doctoral student in Industrial Engineering and Management at Institut Teknologi Bandung (ITB). Her research interests are Production Planning and Scheduling. Her email address is rachmi@trisakti.ac.id and rahmimaulidya@gmail.com.

Rachmawati Wangsaputra is a lecturer at the Department of Industrial Engineering, Institut Teknologi Bandung (ITB). She received her doctoral degree from the Department of Industrial Engineering, New Mexico State University, USA in 2002. Her teaching and research interest include dynamic optimization, manufacturing systems design, vision and robotic system, manufacturing automation and artificial intelligent with special emphasis on intelligent vision system. Her email address is rwangsap@mail.ti.itb.ac.id and rachmawati_wangsaputra@yahoo.com.

Suprayogi is an Associate Professor at Faculty of Industrial Technology, Institut Teknologi Bandung (ITB). He received his undergraduate and master degree in Industrial Engineering from ITB and his Doctorate degree from University of Tokyo, Japan. His research interests are Transportation and Distribution, Maritime Logistic, Operation Research, and Soft Computing and Algorithm Design. His email address is yogi@mail.ti.itb.ac.id

Abdul Hakim Halim is Professor of Industrial Engineering at Industrial Engineering Department, Institut Teknologi Bandung (ITB), Bandung, Indonesia. He received his undergraduate and master degree in Industrial Engineering from ITB, and his Doctorate degree from Industrial Engineering Department, Osaka Prefecture University, Osaka, Japan. His research interests are Production Scheduling, Inventory control, FMS, and JIT systems. He published his papers in International Journal of Production Research, European Journal of Operational Research, International Journal of Production Economics, and Production Planning and Control, as well as in Journal of Japan and several national journals on production system written in Indonesian. His email address is ahakimhalim@mail.ti.itb.ac.id and ahakimhalim@gmail.com.