IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Job Scheduling for Hybrid Assembly Differentiation Flow Shop to Minimize Total Actual Flow Time considering Multi-Due-Dates

To cite this article: R Maulidya *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **528** 012063

Job Scheduling for Hybrid Assembly Differentiation Flow Shop to Minimize Total Actual Flow Time considering Multi-Due-Dates

R Maulidya, Suprayogi, R Wangsaputra and AH Halim

Industrial Engineering Faculty, Institut Teknologi Bandung, Bandung 40132, Indonesia
rahmimaulidya@gmail.com, yogi@mail.ti.itb.ac.id, rwangsap@mail.ti.itb.ac.id, ahakimhalim@ti.itb.ac.id

Abstract. The hybrid assembly differentiation flow shop is a three-stage flow shop system with machining, assembly and differentiation stages. The machining stage produces *k* parts in independently unrelated dedicated machines. After *k* parts of a job completed, they are assembled at the assembly stage and assigned to dedicated machines for a final process. At the end, several types of finished products are resulted. The finished products should be delivered at their respective different due dates. The problem is to find a schedule considering multi due dates, and the criterion of minimizing total actual flow time. This paper proposes a model for job scheduling in a hybrid assembly differentiation flow shops and its algorithm for solving the problems. The initial solution is defined using SPT-based heuristic and it is optimized using variable neighbourhood descend method. The proposed model is tested using a set of hypothetic data. The solution shows that the algorithm can solve the problems effectively.
Keywords: job scheduling, hybrid assembly differentiation flow shop, total actual flow time, multi due dates

1. Introduction

This paper deals with a three-stage hybrid flow shop that consists of machining parts, an assembly operation and processing different end products called Hybrid Assembly Differentiation Flow Shop. The first stage is the machining stage where several dedicated machines produce different parts in parallel manner. The second stage is the stage where all parts of a product will be assembled in a single machine after they completed. The final stage is the differentiation stage consists of dedicated machines where an assembled part is processed into a certain type of products. The application of this system is illustrated from the electronic manufacturing where the system consists of a three-stage production system producing different types of products.

The recent pieces of research [1-4] developed the three-stage production system for job scheduling problem. According to [1], the three stage of production system consists of machining, transportation and assembly stages. Transportation is seen as an important aspect in job scheduling problem. The difference of [1], [2] define a three-stage production system as the machining processes in three stages. Other pieces of research [3-4] developed a three-stage production system that consists of machining, assembly and differentiation stages. In this paper, we extend the research proposed in [4] to cover a condition of multi-due-date.

The literature of [1-3] and [5] for job scheduling usually solve the problem using forward approach with the assumption of parts that should be available at time zero. The Model proposed in [3] uses the total flow time as the criteria to minimize the time spending by parts in the shop from time zero until the completion time. Other criteria developed by [5] using the performance measure of total tardiness to minimize the penalty between the job's due date and its completion time. In this paper, we use backward approach and the criterion of minimizing total actual flow time to minimize the interval time of parts spend in the shop from the arrival time until its due date. Total actual flow time is defined for some cases where parts are not available simultaneously in time zero, but will arrive when its needed, and other cases that consider the delivery time from due date of parts [6].

This paper deals with a problem of job scheduling model for a hybrid assembly differentiation flow shop to minimize the total actual flow time. We extend the problem in [4] to cover a condition where the finished products are demanded at several due dates meaning that there are several delivery times within a production planning. This paper is organized as section 2 for formulating the model, section 3 for the proposed algorithm, section 4 for the illustration and section 5 for the conclusion and the future research.

2. Method

2.1. Problem Formulation

Consider the problem of scheduling J jobs at a hybrid assembly differentiation flow shop to make H different types of finished products. We can denote the jobs belong to a certain type as JH_h. An illustration in Figure 1 show that there are four jobs in the system that consist of product type 1, JH_1=J_1,J_2, and product type 2, JH_2=J_3,J_4. Each job consists of part#1, part#2…part#k produced by unrelated parallel machines in the machining stage. Parts are processed on each machine in the same order. After all parts of a job are completed, they are assembled in a single assembly machine. The differentiation stage processes the assembled parts into a certain types of products in particular dedicated machines. There is a condition where the products must be delivered in different due dates. It means that each job must fulfil their own due date.

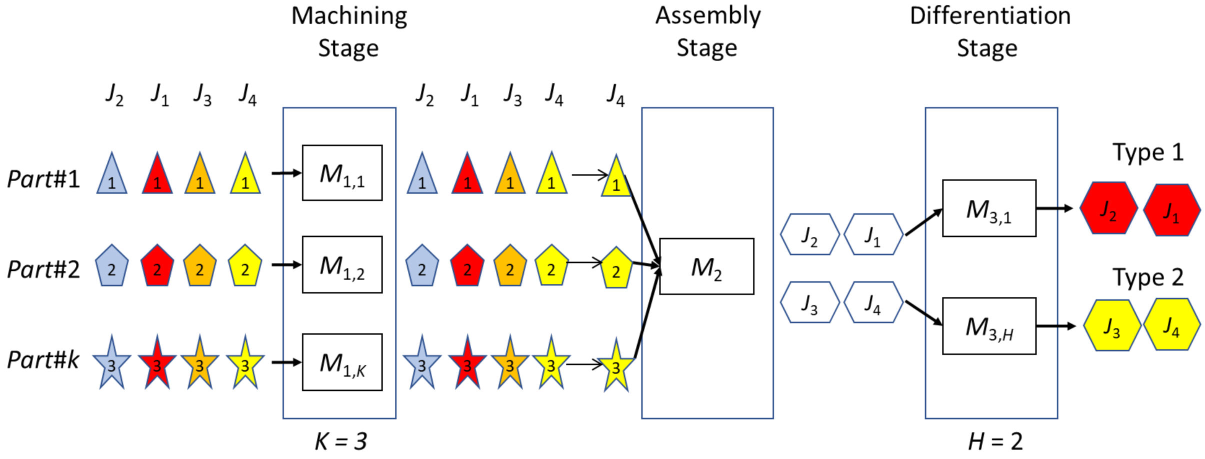

Figure 1. The three stages production system

The assumptions in this paper are:

- Production are finished at their corresponding due dates
- Each machine can process at most one job at a time
- Each job can be processed on at most one machine at a time
- Machines are always available

- Travel time between stages are neglected
- Job processing cannot be interrupted, until the completion of job.
- Jobs are available for processing at a stage immediately after processing completion at the previous stage.

2.2. Model

The actual flow time of a job ($F^a_{[i]}$), defined by [7], is the time spend by a job from its arrival time in the shop until its due date. The formulation for $F^a_{[i]}$ is as follows.

$$F^a_{[i]} = d - B_{[i]}, \qquad i=1,2\ldots J \tag{1}$$

Where d is a due date and $B_{[i]}$ is the starting time for processing J_j. The schedule is started backwardly and the position of $[i]$ is started from its due date.

The formulation of the actual flow time of a job with a due date is as follows.

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

Figure 2 shows the actual flow time of several due dates d_1 and d_2, respectively.

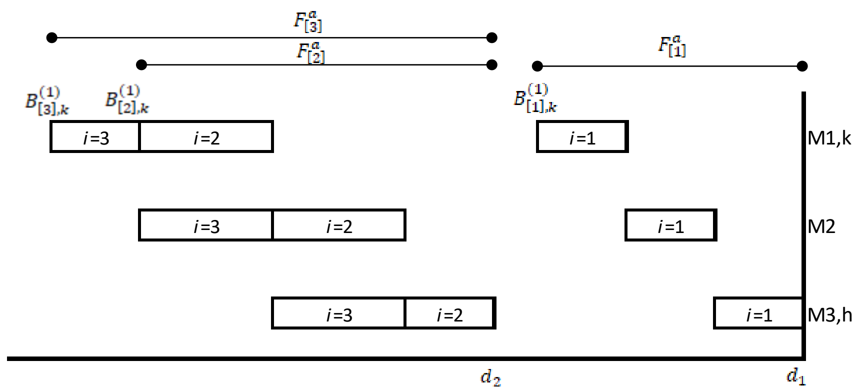

Figure 2. Actual Flow Time for multi due dates

From Figure 2, we have the actual flow time for position i=1,2,3.

$$F^a_{[1]} = d_1 - \min_{k\le K}\left(B^{(1)}_{[1],k}\right)$$

$$F^a_{[2]} = d_2 - \min_{k\le K}\left(B^{(1)}_{[2],k}\right)$$

$$F^a_{[3]} = d_2 - \min_{k\le K}\left(B^{(1)}_{[3],k}\right)$$

$$TAFT = F^a_{[1]} + F^a_{[2]} + F^a_{[3]}$$

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

Notation

Index

j = index of jobs, j=1, 2…J

i = index of positions, i=1, 2…J

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

h = index of machines and the product types at the differentiation stage, h=1, 2…H

Parameter

K = number of machines at the machining stage

H = number of machines at the differentiation stage

J = number of processed job

JH_h = number of job processed in machine h

d_j = due date for job j

G_g = interval of two consecutive due dates

M_{1k} = machine for processing part k of all the jobs at the machining stage, k = 1, 2…K
M_2 = assembly machine in the assembly stage
M_{3h} = differentiation machine for processing types of products in the differentiation stage, h = 1, 2…H
$p_{j,k}^{(1)}$ = the processing time of part k job j in machine M_{1k} at the machining stage
$p_j^{(2)}$ = the processing time of assembly operation of job j on machine M_2 at the assembly stage
$p_{j,h}^{(3)}$ = the processing time of job j type h in machine M_{3h} at the differentiation 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 differentiation 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 machining stage
$B_{[i]}^{(2)}$ = the starting time of assembly job in position i in machine M_2 at the assembly stage
$B_{[i],h}^{(3)}$ = the starting time of job in position i in machine M_{3h} at the differentiation stage
Objective function
$TAFT$= total actual flow time

The problem can be formulated as follows.

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

Subject to:

$$B_{[i],h}^{(3)} = Y_{[i],h}\left(X_{j[i]}\left(d_j - p_{j,h}^{(3)}\right) - G_{[z],h}\left(\left(d_j * X_{j[z]}\right) - B_{[z],h}^{(3)}\right)\right), \qquad z \le i,\ \forall i, \forall h \tag{5}$$

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

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

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

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

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

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

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

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

$$G_{[z],h} = \begin{cases} 1, & \text{if } B_{[z],h}^{(3)} < (d_j * Y_{[z],h} * X_{j[z]}) \\ 0, & \text{if } B_{[z],h}^{(3)} \ge (d_j * Y_{[z],h} * X_{j[z]}) \end{cases} \quad z < i, j = 1,2..n, \forall h \tag{14}$$

$$X_{j[i]} \in \{0,1\} \tag{15}$$

$$G_{[z],h} \in \{0,1\} \tag{16}$$

$$B_{[i],k}^{(1)}, B_{[i]}^{(2)}, B_{[i],h}^{(3)} \ge 0, \text{ and integer} \tag{17}$$

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) defines the starting time of job in position i in machine h at the third stage from the due date of the job considering the decision that position i is assigned in machine h and there is no starting time of other job before the due date d_j. Constraint (6) defines 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 $i^* \leq i$. Constraint (7) and (9) ensure that each position can only be processed at the same stage after the next position is finished. Constraint (8) defines 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) defines the job with the processing time at the third stage greater than zero, is processed in machine h at the third stage. Constraint (11) decides 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) defines that there is no other job of position z processed until the due date. Constraint (15) and (16) define the domain of the decision variables. Constraint (17) ensures that the starting time is feasible from time zero.

2.3. Algorithm

We develop a solution using SPT based heuristic and optimized by Variable Neighbourhood Descent (VND) methods. The multi-due-date has condition where the jobs must be scheduled to meet their due dates. Figure 3 show that there are several due dates and between two consecutive due dates the jobs must be processed. The proposed algorithm divided the jobs according to the interval they belong.

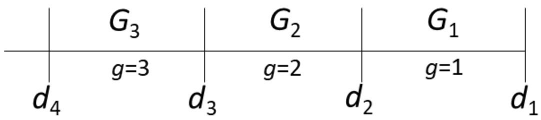

Figure 3. Interval of two consecutive due dates

SPT-based heuristic generates the job sequences based on the processing time of the three stages in increasing orders. The jobs are scheduled from their corresponding due dates and the value of TAFT is defined. The best initial solution from SPT-based heuristic is optimized using VND method. VND with the insert move determined the job sequence that minimizes the total actual flow time. In VND-insert move, a job is moved to a position so that the current positions are shifted one position forward along the solution [8]. The stopping rule is the condition where the iterations give no new minimum total actual flow time after a set of iterations.

The proposed algorithm

Step 0: Initialize the problem

Step 1: Sequence the due dates in increasing order, the greatest due dates will be g=1.

Step 2: Determine G_g as the interval between two consecutive due dates using:

$$G_g = \begin{cases} d_g - d_{g+1}; & g = 1, \dots, r-1 \\ d_r; & g = r \end{cases}$$

Step 3: Set g=1

Step 4: Generate the job sequences in increasing orders for all jobs in interval G_g based on the SPT-based heuristic for $\max_{1\leq k\leq 3}\left(p_{j,k}^{(1)}\right), p_j^{(2)}, p_{j,h}^{(3)}$, $\max_{1\leq k\leq 3}\left(p_{j,k}^{(1)}\right) + p_j^{(2)}, p_j^{(2)} + p_{j,h}^{(3)}$ and $\max_{1\leq k\leq 3}\left(p_{j,k}^{(1)}\right) + p_j^{(2)} + p_{j,h}^{(3)}$, respectively. Schedule the jobs from theirs corresponding due dates.

Step 5: Set the best total actual flow time as the best initial solution.

Step 6: Perform procedure Variable Neighbourhood Descent (insert move) as follows.

- For all the jobs, i=1,2…J, in vn_{seq}, repeat
- Set i^{th} job in vn_{seq} sequence as J^{i}_{vn}
- Insert J^{i}_{vn} in all possible positions in vn_{seq} to get $N^1(x)$, a subset of $N(x)$.
- Find *TAFT* of all sequences in $N^1(x)$
- Set sequence with minimum *TAFT* in $vn_{seq.}$
- Output the final sequence vn_{best}.

Step 6: Is the solution better than the best initial solution?
If Yes, output the best solution
If No, then stop.

3. Result and Discussion

Consider an illustration of eight jobs (J=8) to analyse the model. There are two product types (Type 1 and Type 2). Table 1 shows the data that 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 $JH_2=\{J_5,J_6,J_7,J_8\}$. The number of machine in the machining stage (K)=3, and the number of machine in the differentiation stage (H)=2. This illustration is run using the processor of Intel Core i7-6500U CPU, 2,50 GHz and 12 GB RAM.

Table 1. Processing time and due date of jobs

Job	Type	$p_{j,1}^{(1)}$	$p_{j,2}^{(1)}$	$p_{j,3}^{(1)}$	$p_{j}^{(2)}$	$p_{j,1}^{(3)}$	$p_{j,2}^{(3)}$	d_j
J_1	1	5	4	3	3	6	0	990
J_2	1	6	3	4	9	10	0	990
J_3	1	6	4	10	3	9	0	980
J_4	1	3	5	5	4	5	0	1000
J_5	2	3	4	5	6	0	8	980
J_6	2	3	4	6	4	0	7	990
J_7	2	10	4	6	5	0	6	980
J_8	2	5	4	3	10	0	4	1000

We can divide the due dates into three sections (g=3) and the jobs are put from their corresponding due dates backwardly. Table 2 shows the initial solution using SPT-based heuristic. There are two sequences with the best TAFT, J_4-J_8-J_1-J_6-J_2-J_5-J_3-J_7 and J_4-J_8-J_1-J_6-J_2-J_5-J_7-J_3.

Table 2. SPT-based heuristic for multi-due-date

$\max_{k\le K}(p_{j,k}^{(1)})$	J_8	J_4	J_1	J_2	J_6	J_5	J_3	J_7	TAFT
	5	5	5	6	6	5	10	10	241
$p_{j,2}^{(1)}$	J_4	J_8	J_1	J_6	J_2	J_3	J_7	J_5	
	4	10	3	4	9	3	5	6	216
$p_{j,h}^{(3)}$	J_8	J_4	J_1	J_1	J_2	J_7	J_5	J_3	
	4	5	6	7	10	6	8	9	218
$\max_{k\le K}(p_{j,k}^{(1)})+p_{j,2}^{(1)}$	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	
	9	**15**	**8**	**10**	**15**	**11**	**13**	**15**	**214**
$p_{j,2}^{(1)}+p_{j,h}^{(3)}$	J_4	J_8	J_1	J_6	J_2	J_7	J_3	$\boldsymbol{J_5}$	
	9	14	9	11	19	11	12	14	219
$\max_{k\le K}(p_{j,k}^{(1)})+p_{j,2}^{(1)}+p_{j,h}^{(3)}$	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_7}$	$\boldsymbol{J_3}$	
	14	**19**	**14**	**17**	**25**	**19**	**21**	**22**	**214**

The proposed algorithm optimized the initial solution using variable neighbourhood descent with the insert moves that can be seen in Table 3. From the two best initial solution, we select the first sequence J_4-J_8-J_1-J_6-J_2-J_5-J_3-J_7. The iteration of Variable Neighbourhood Descent started with the first job of J_4 to be inserted into vn_{seq} (sequence 1 until sequence 7) with different positions. The *TAFT* of each sequences are defined. In iteration 1, there are no improvement for the vn_{best}. The iteration continued with the next job from the sequence selected. In iteration 5 and 8, we get the new vn_{best} for the same TAFT value of 211. Those sequences are J_4-J_8-J_1-J_6-J_5-J_3-J_7-J_2 and J_4-J_8-J_1-J_6-J_7-J_2-J_5-J_3. We can see that J_2 with the longest processing time in both assembly and differentiation stages is scheduled at the third section (the respective due date). It is because the interval between the second and the third due date are smaller than the total three job's processing time (J_1, J_2, J_6). Figure 4 is the Gantt chart for the best sequence after the optimization. Figure 4 also shows the cluster of jobs according to the interval between two due dates and the position of J_2.

Table 3. Variable Neighbourhood Descent-insert moves

Iteration 1 (J_4)										Iteration 2 (J_8)									
vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	**J3**	$\boldsymbol{J_7}$	**214**	vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	**214**
1	J_8	J_4	J_1	J_6	J_2	J_5	J3	J_7	216	1	J_8	J_4	J_1	J_6	J_2	J_5	J_3	J_7	216
2	J_8	J_1	J_4	J_6	J_2	J_5	J3	J_7	244	2	J_4	J_1	J_8	J_6	J_2	J_5	J_3	J_7	254
3	J_8	J_1	J_6	J_4	J_2	J_5	J3	J_7	241	3	J_4	J_1	J_6	J_8	J_2	J_5	J_3	J_7	248
4	J_8	J_1	J_6	J_2	J_4	J_5	J3	J_7	243	4	J_4	J_1	J_6	J_2	J_8	J_5	J_3	J_7	262
5	J_8	J_1	J_6	J_2	J_5	J_4	J3	J_7	248	5	J_4	J_1	J_6	J_2	J_5	J_8	J_3	J_7	256
6	J_8	J_1	J_6	J_2	J_5	J3	J_4	J_7	253	6	J_4	J_1	J_6	J_2	J_5	J_3	J_8	J_7	248
7	J_8	J_1	J_6	J_2	J_5	J3	J_7	J_4	254	7	J_4	J_1	J_6	J_2	J_5	J_3	J_7	J_8	247
Iteration 3 (J_1)										Iteration 4 (J_6)									
vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	**214**	vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	**214**
1	J_1	J_4	J_8	J_6	J_2	J_5	J_3	J_7	306	1	J_6	J_4	J_8	J_1	J_2	J_5	J_3	J_7	307
2	J_4	J_1	J_8	J_6	J_2	J_5	J_3	J_7	254	2	J_4	J_6	J_8	J_1	J_2	J_5	J_3	J_7	265
3	J_4	J_8	J_6	J_1	J_2	J_5	J_3	J_7	214	3	J_4	J_8	J_1	J_6	J_2	J_5	J_3	J_7	214
4	J_4	J_8	J_6	J_2	J_1	J_5	J_3	J_7	237	4	J_4	J_8	J_1	J_2	J_6	J_5	J_3	J_7	239
5	J_4	J_8	J_6	J_2	J_5	J_1	J_3	J_7	222	5	J_4	J_8	J_1	J_2	J_5	J_6	J_3	J_7	246
6	J_4	J_8	J_6	J_2	J_5	J_3	J_1	J_7	229	6	J_4	J_8	J_1	J_2	J_5	J_3	J_6	J_7	250
7	J_4	J_8	J_6	J_2	J_5	J_3	J_7	J_1	232	7	J_4	J_8	J_1	J_2	J_5	J_3	J_7	J_6	250
Iteration 5 (J_2)										Iteration 6 (J_5)									
vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	**214**	vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	$\boldsymbol{J_2}$	**211**
1	J_2	J_4	J_8	J_1	J_6	J_5	J_3	J_7	349	1	J_5	J_4	J_8	J_1	J_6	J_2	J_3	J_7	399
2	J_4	J_2	J_8	J_1	J_6	J_5	J_3	J_7	301	2	J_4	J_5	J_8	J_1	J_6	J_2	J_3	J_7	229
3	J_4	J_8	J_2	J_1	J_6	J_5	J_3	J_7	233	3	J_4	J_8	J_5	J_1	J_6	J_2	J_3	J_7	269
4	J_4	J_8	J_1	J_2	J_6	J_5	J_3	J_7	239	4	J_4	J_8	J_1	J_5	J_6	J_2	J_3	J_7	247
5	J_4	J_8	J_1	J_6	J_5	J_2	J_3	J_7	219	5	J_4	J_8	J_1	J_6	J_2	J_5	J_3	J_7	214
6	J_4	J_8	J_1	J_6	J_5	J_3	J_2	J_7	215	6	J_4	J_8	J_1	J_6	J_2	J_3	J_5	J_7	213
7	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	$\boldsymbol{J_2}$	**211**	7	J_4	J_8	J_1	J_6	J_2	J_3	J_7	J_5	230
Iteration 7 (J_3)										Iteration 8 (J_7)									
vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	$\boldsymbol{J_2}$	**211**	vn_{best}	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	$\boldsymbol{J_7}$	$\boldsymbol{J_2}$	**211**
1	J_3	J_4	J_8	J_1	J_6	J_2	J_5	J_7	401	1	J_7	J_4	J_8	J_1	J_6	J_2	J_5	J_3	379
2	J_4	J_3	J_8	J_1	J_6	J_2	J_5	J_7	273	2	J_4	J_7	J_8	J_1	J_6	J_2	J_5	J_3	325
3	J_4	J_8	J_3	J_1	J_6	J_2	J_5	J_7	231	3	J_4	J_8	J_7	J_1	J_6	J_2	J_5	J_3	265
4	J_4	J_8	J_1	J_3	J_6	J_2	J_5	J_7	249	4	J_4	J_8	J_1	J_7	J_6	J_2	J_5	J_3	238
5	J_4	J_8	J_1	J_6	J_3	J_2	J_5	J_7	233	5	$\boldsymbol{J_4}$	$\boldsymbol{J_8}$	$\boldsymbol{J_1}$	$\boldsymbol{J_6}$	$\boldsymbol{J_7}$	$\boldsymbol{J_2}$	$\boldsymbol{J_5}$	$\boldsymbol{J_3}$	**211**
6	J_4	J_8	J_1	J_6	J_2	J_3	J_5	J_7	213	6	J_4	J_8	J_1	J_6	J_2	J_7	J_5	J_3	216
7	J_4	J_8	J_1	J_6	J_2	J_5	J_7	J_3	230	7	J_4	J_8	J_1	J_6	J_2	J_5	J_7	J_3	230

The optimal model has run for five hours using a programming software, but the result is still local optimal with the best TAFT=346. The performance from this illustrative example is 14.31%. For this illustration, the result is better than the optimal model. Though it still need more illustrative data to make the best conclusion.

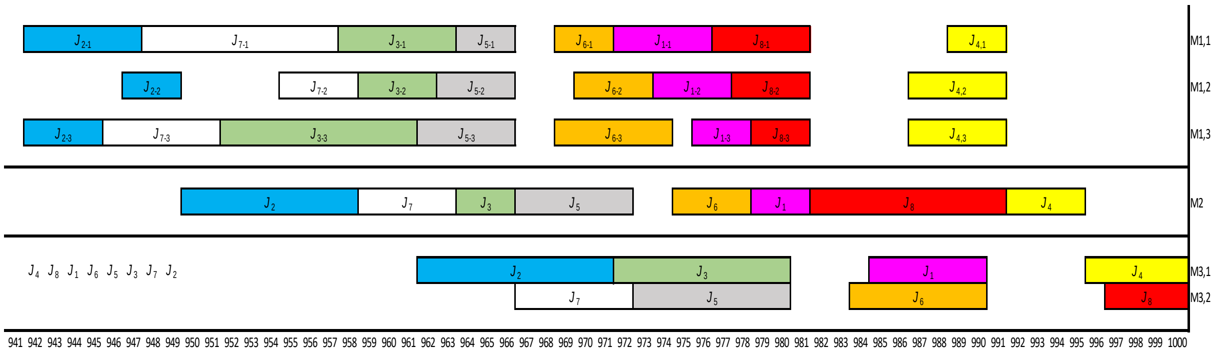

Figure 4. Gantt Chart for *TAFT*=211

4. Conclusions

This paper deals with job scheduling problems for hybrid assembly differentiation flow shop to minimize total actual flow time with the condition of multi-due-date. The results of this paper are job scheduling model for multi-due-date condition and the proposed algorithm consists of SPT based heuristic for initial solution and variable neighbourhood descend method for optimization. The illustration is only for eight jobs and there must be more experiments to have a better conclusion. The proposed algorithm can give the better solution and shows the improvement from the initial solution. The future research is to find the proposed algorithm that can give optimal solution in short time compare to the current approach.

Acknowledgments

This research was fully funded by Doctoral Dissertation Research Funding from Directorate General Higher Education Indonesia (2018), Ministry of Research, Technology and Higher Education under contract No.011/KM/PNT/2018 with Universitas Trisakti.

5. References

[1] Koulamas, C and Kyparisis, G J 2001 The Three-stage Assembly Flowshop Scheduling Problem. *Computers & Operations Research* **28** pp 689-704

[2] Babayan, A and He, D 2004 Solving The n-job 3-stage Flexible Flowshop Scheduling Problem using An Agent-based Approach. *International Journal of Production Research* **42**(4) pp 777-799.

[3] 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** pp 338-54

[4] 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) pp 1198-1204

[5] Allahverdi A and Aydilek H 2015 The two-stage assembly flowshop scheduling problem to minimize total tardiness. *J Intell Manuf* **26** pp 225–237

[6] 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** pp 529-44

[7] Halim AH, 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) pp 175-184

[8] Vanchipura R, Sridharan R and Babu A S 2014 Improvement of constructive heuristics using variable neighbourhood descent for scheduling a flow shop with sequence dependent setup time. *Journal of Manufacturing Systems* **33** pp 65–75