International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 01, 2020
ISSN: 1475-7192

MODEL HYBRID FLOW SHOP 3 STAGE TO MINIMIZE MAKESPAN

Case Study: PT XYZ Fishnet Manufacturing

[1]Didit Damur Rochman, Asep Anwar, Rendiyatna Ferdian, Riki Ridwan Margana

Abstract---*Hybrid flow shop production system is currently widely applied in manufacturing companies, one of the companies implementing it is PT XYZ which is engaged in the production of fishing net manufacturing. The HFS system is a production system consisting of several stages with identical or non-identical engine configurations. Based on observations that have been made problems arise when a new job comes while a stage is still producing a previous job. This resulted in the work having to wait until the previous work was completed. These conditions make the time of completion of a job (makespan) become hampered. This study aims to develop a hybrid flow shop production system model with the aim to reduce the makespan. Based on the results of data processing, it was found that by using the HFS model the time needed to produce 10 jobs was 295.25 with a computation time of 3.47 minutes using LINGO software.*

Keywords---*Hybrid Flow Shop, Three-stage, Makespan*

I. INTRODUCTION

PT XYZ is a company engaged in manufacturing fishing nets. The company's production process is Make to Order (MTO) where production is based on orders coming from consumers. As a manufacturing company, production planning is one of the important factors in the company's sustainability, because most of the costs incurred are production costs. The company is currently implementing a First Come First Serve (FCFS) system where production is scheduled based on orders that come first. This condition makes the order finished in the order in which the order arrived, but the system keeps the efficiency level of the machine low because the incoming order has to wait for the previous process to be finished producing. Low engine efficiency on the other hand also results in large production costs incurred.

Hybrid flow shop (HFS) production system is a production system where there are a number of n tasks that must be produced through m stages with specific objective functions (Ruiz et al, 2010). This system aims to minimize the makespan which on the other hand is also expected to increase engine efficiency. This research aims to plan a production system of PT XYZ using a hybrid flow shop production system with the objective function to makespan minimization.

[1]*Production System Laboratorium*
Industrial Engineering
Widyatama University Bandung
diditdr@widyatama.ac.id, asep.anwar@widyatama.ac.id, rendiyatna.ferdian@widyatama.ac.id, riki.ridwan@widyatama.ac.id

II. LITERATURE REVIEW

HFS production planning often focuses on three important factors including process complexity, model criteria, and solution-solving methods. Research on the current level of process complexity can be broadly categorized into three categories: two-stage HFS, three-stage HFS, and k-stage HFS (Linn and Zhang, 1999).

Two-stage HFS

Two-stage HFS consists of two stages of the production process where at each stage can consist of identical or non-identical machines. The first two-stage HFS problem solving is where the machine configuration consists of a single machine in the first stage, and an identical parallel machine in the second stage. The search for a solution with an objective function for makespan minimization in this configuration can be categorized as an NP-Hard problem (Hoogeveen et al, 1996). The next configuration consisted of a single machine in the first stage, and two non-identical machines in the second stage, Narasimhan (1984) developed a solution with a heuristic procedure that was named the Cumulative Minimum Deviation (CMD) rule. The procedure is proven to produce a better solution than using the Shortest Processing Time (SPT) and Longest processing Time (LPT) to reduce the idle time of a machine and the waiting time in stage two (Linn and Zhang, 1999). The third configuration where there are identical parallel machines in the first stage and a single machine in stage two can be solved using the Branch and Bound (B&B) algorithm properly by producing minimum makespan.

Three-stage HFS

The three-stage HFS configuration which consists of non-identical machines at stages 1, 2, and 3 can be completed with five approaches, namely identifying bottlenecks, calculating time lags for tasks at the bottleneck stage, calculating machine capacity at the bottleneck stage, scheduling tasks at the stage bottlenecks, and task scheduling at the non-bottleneck stage (Adler et al, 1993).

k-stage HFS

The configuration of the k-stage HFS consists of parallel machines at each stage can be completed using three approaches, namely determining which tasks will be allocated to a machine at each stage, sorting tasks on each machine, and finally determining the time settings for when a task starts on a stage.

Exact Algorithm

Based on several previous studies, it can be concluded that the Branch and Bound (B&B) algorithm is the most appropriate technique used to solve the HFS optimization problem. Although the realistic B&B algorithm is suitable for use in solving HFS problems, the method is not very good when used to solve real-world problems. High level of complexity in real cases makes problem solving using B&B algorithms less effective (Ruiz, 2010). Now, more research is needed regarding the method of solving HFS by using a simpler method that is useful but produces an efficient solution when applied to real cases.

III. PROBLEM STATEMENT

The object of research is the process of making spare parts with Hybrid Flow Shop model. The description of the problem in this study is that there are a number of jobs (j) that will be processed through several stages of the process. On stage 1 and stage 3 there are 2 identical machines installed in parallel while in stage 2 there is only 1 machine that can be used to process all outputs from stage 1.

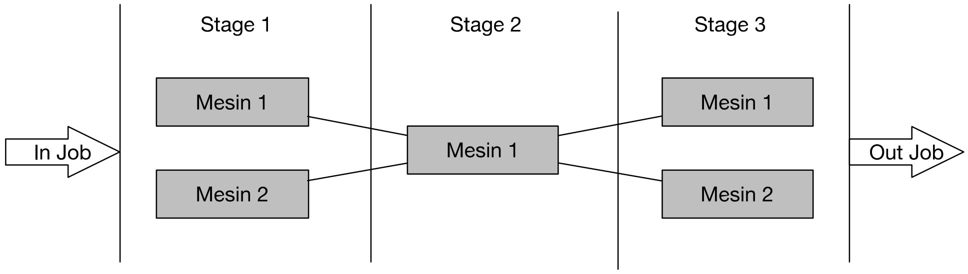

Figure 1. Machine Configuration

Notation:

j = job.

w = job sceduled.

a = Machine in stage 1

c = Machine in stage 3

$S1_j$ = Starting time job (j) for stage 1.

$S2_j$ = Starting time job (j) for stage 2.

$S3_j$ = Starting time job (j) for stage 3.

t_j = processing time job (j) in stage 1.

p_j = processing time job (j) in stage 2.

r_j = processing time job (j) in stage 3.

$C1_j$ = Completion time job (j) in stage 1.

$C2_j$ = Completion time job (j) in stage 2.

$C3_j$ = Completion time job (j) in stage 3.

$X_{j,a} = \begin{cases} 1, If\ job\ j\ assigned\ on\ machine\ a \\ 0, if\ job\ j\ unassigned\ on\ machine\ a \end{cases}$

$Y_{j,a} = \begin{cases} 1, If\ job\ j\ assigned\ on\ machine\ c \\ 0, if\ job\ j\ unassigned\ on\ machine\ c \end{cases}$

Objective Function:

$$MIN\ M = Max(C3_{j,c}) \qquad (1)$$

Subject to:

$$\sum_{a=1}^{A} X_{j,a} = 1 \qquad (2)$$

$$S1_{j,a} = C1_{w,a} \qquad (3)$$

$$C1_{j,a} = \left(S1_{j,a} + t_j\right) \times X_{j,a} \quad (4)$$

$$S2_j \geq C1_{j,a} \quad (5)$$

$$S2_j \geq C2_w \quad (6)$$

$$C2_j = S2_j + p_j \quad (7)$$

$$\sum_{c=1}^{C} Y_{j,c} = 1 \quad (8)$$

$$S3_{j,c} \geq C2_j \quad (9)$$

$$S3_{j,c} \geq C3_{w,c} \quad (10)$$

$$C3_{j,c} = \left(S3_{j,c} + r_j\right) \times Y_{j,c} \quad (11)$$

$$X_{j,a}, Y_{j,c} = \{0,1\} \quad (12)$$

$$S1_{j,a}, C1_{j,a}, S2_j, C2_j, S3_{j,c}, C3_{j,c} \geq 0 \quad (13)$$

Equation (1) is an objective function to minimize makespan. Equation (2) ensures that each job will only be processed by one machine in stage 1. Equation (3) guarantees that the time the job starts will be done if the previous job is finished on the same machine. Equation (4) calculates completion time for every job in stage 1. Equation (5) guarantees that a job will be processed in stage 2 after it has been processed in stage 1. Equation (6) ensures that the next processed job is done after the previous job has been processed stage 2. Equation (7) calculates completion time on stage 2. Equation (8) guarantees that jobs in stage 3 will only be done by one machine. Equation (9) ensures that a job will be processed on stage 3 after the job has finished processing on stage 2. Equation (10) ensures that a job on stage 3 will be processed after the previous job has finished processing. Equation (11) determine completion time job at stage 3. Equation (12) all decision variables are in binary form. Equation (13) all variables are non-negative.

IV. COMPUTATION AND RESULT

The amount of work that is the object of this study is 10 ordered in a certain period of time and has been sorted according to the FIFS method, so that the sorting process will not be carried out with other methods in this study. The job data can be seen in the table 1.

Table 1 Processing Time each Job

No Seq.	Stage 1 (Hour)	Stage 2 (Hour)	Stage 3 (Hour)
1	36,44	18,22	42,04
2	36,64	18,32	42,28
3	49,07	24,54	81,79
4	6,29	3,14	23,58
5	23,23	11,62	20,50
6	49,67	24,84	82,78
7	20,21	10,11	60,64

8	16,12	8,06	13,07
9	23,38	11,69	20,63
10	20,29	10,15	60,87

The computation process is carried out with the LINGO 11 software and using a 2.5GHz intel core i5 computer specification and 16GB RAM. The calculation process using the source code will be used as a calculation command in the LINGO software as shown.

```
Model:
sets:
job/1..10/:t,p,r,C2,S2;
jobw/1..10/;
mesin1/1..2/;
mesin3/1..2/;
Xj1(Job,mesin1):X,S1,C1;
Xj3(Job,mesin3):Y,S3,C3;

endsets
data:
t= 36.44 36.64 49.07 6.29 23.23 49.67 20.21 16.12 23.38 20.29;
p= 18.22 18.32 24.54 3.14 11.62 24.84 10.11 8.06 11.69 10.15;
r= 42.04 42.28 81.79 23.58 20.50 82.78 60.64 13.07 20.63 60.87;
enddata
!Objective Function;
MIN=@max(XJ3(j,c):C3(j,c));

!Stage 1;
@for(job(j):@sum(Mesin1(a):X(j,a))=1);
@for(job(j):@for(jobw(w)|w#LT#j:@for(mesin1(a):S1(j,a)>=C1(w,a))));
@for(job(j):@for(mesin1(a):C1(j,a)=(S1(j,a)+t(j))*X(j,a)));

!Stage 2;
@for(job(j):@for(mesin1(a):S2(j)>=C1(j,a)));
@for(job(j):@for(jobw(w)|w#LT#j:S2(j)>=C2(w)));
@for(job(j):C2(j)=S2(j)+p(j));

!Stage 3;
@for(job(j):@sum(Mesin3(c):Y(j,c))=1);
@for(job(j):@for(mesin3(c):S3(j,c)>=C2(j)));
@for(job(j):@for(jobw(w)|w#LT#j:@for(mesin3(c):S3(j,c)>=C3(w,c))));
```

```
@for(job(j):@for(mesin3(c):C3(j,c)=(S3(j,c)+r(j))*Y(j,c)));

@for(job(j):@for(mesin1(a):@BIN(X(j,a))));
@for(job(j):@for(mesin1(a):C1(j,a)>=0));
@for(job(j):C2(j)>=0);
@for(job(j):@for(mesin3(c):@BIN(Y(j,c))));
@for(job(j):@for(mesin3(c):C3(j,c)>=0));
end
```

Output after calculation using the application is obtained in optimal conditions as shown. The optimal value of the calculation results is 295.25 hours.

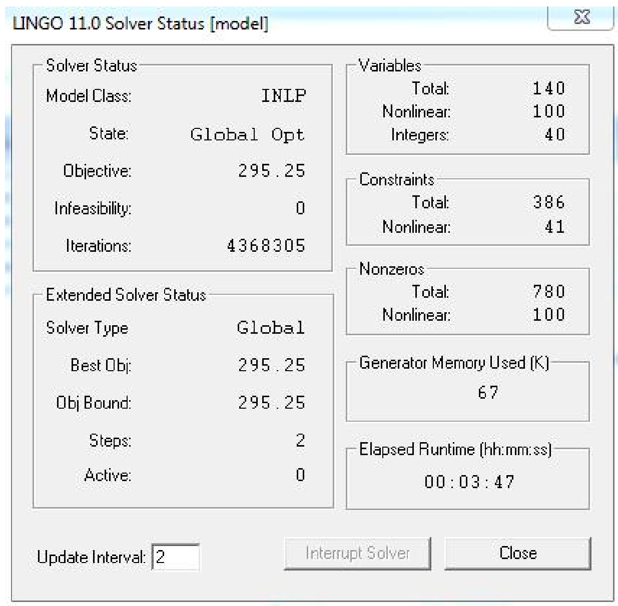

Figure 2 Solver Status LINGO

The results also get a sequence of job performance on each machine. Machine assignments can be seen from the table 2.

Table 2 Job Sequencing

Stage	Machine	Job Sequencing
Stage 1	1	1,3
	2	2,4,5,6,7,8,9,10
Stage 2	1	1,2,3,4,5,6,7,8,9,10
Stage 3	1	1,3,6,9
	2	2,4,5,7,8,10

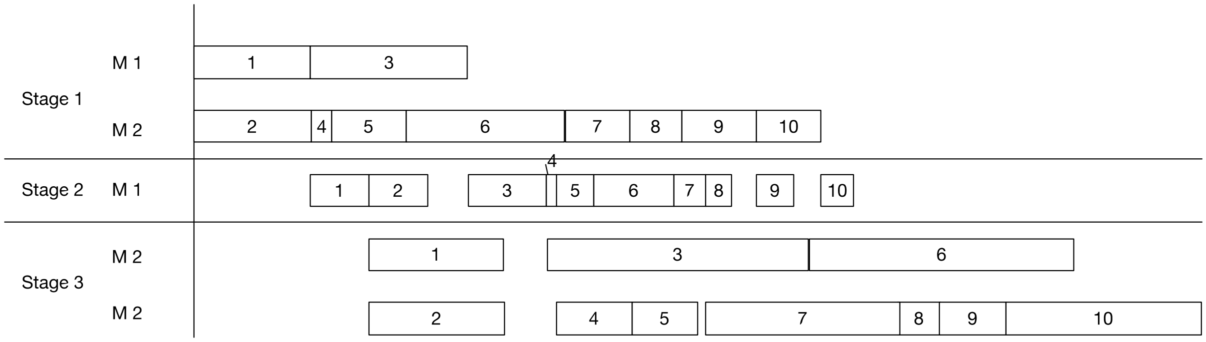

Figure 3 Ghant Chart for Result Scheduling

V. CONCLUSION

The results of this study are able to produce mathematical models that can be used to solve scheduling problems with a 3-stage hybrid flow shop model. After testing this model, it can be seen that this model can solve the hybrid flow shop scheduling problem with FCFS sequencing rules. By using this model, the optimal makespan value is 295.25 with the search time for a solution using LINGO software and the time required is 3.47 minutes.

This research is only used as a reference for completion with a small number of jobs, for the future it is necessary to implement a model for a large number of jobs so that this model will be a reference model for implementation in the field and not only for theoretical concepts. Implementation with more and more jobs will lead to the amount of time needed for computing.

REFERENCES

[1] Adler, LB. *et aL* (1993). BPSS: **A** schedulingsupportsystem**for the** packa~ng *industry.J.Oper.Res.4,1641-648*
[2] Hoogeveen, J. A., Lenstra, J. K. & Vel~nan, B. (1996). *Preemptive scheduling in a two-stage nultiprocessor flow shop is NP-hard*, Eur.J. Oper. Res.,89, 172-175.
[3] Linn, R., & Wei Zhang. (1999). *Hybrid Flow Shop Scheduling: A Survey*. Computers & Industrial Engineering 37 (1999) 57-61.
[4] Narasimhan, S.L & Panwalkar, S.S. (1984). *Scheduling in a two-stage manufacturing process. lnt. J. Prod. Res, 22,555-64.*
[5] Ruiz R., Jose Antonio V-R. (2010). *The Hybrid Flow Shop Scheduling Problem*. European Journal of Operation Research 205 (2010) 1-18.