Journal of Telematics and Informatics (JTI)
Vol.6, No.3, September 2018, pp. 187~192
ISSN: 2303-3703

Integer Programming for Scheduling Computation Alternative Machines Parallel Multi Operations

Irwan Sukendar[1], Suprayogi[2], Imam Much Ibnu Subroto[3]
[1]Industrial Engineering Department, Universitas Islam Sultan Agung Semarang
[2]Industrial Engineering Department, Institut Teknologi Bandung
[3]Informatics Engineering Department, Universitas Islam Sultan Agung Semarang
e-mail: irwan@unissula.ac.id

Abstract

This paper deals with integer programming for computing alternative scheduling problems of multi-parallel machine operation. The purpose function used is weighted total tardiness. The model formulation comprises the formulation of the objective function and the formulation of the 16 limiting functions. The model was tested using a numerical example consisting of four multi-operation jobs and several machine alternatives in mathematical software Lingo 9. The programming results show that a feasible solution with a weighted total tardiness measure size of eight.

Keywords*: Integer Programming, Scheduling*

1. Introduction

Manufacturing systems are required to be able to produce flexibly. One type of flexibility in manufacturing is the flexibility of the process. Process flexibility is related to the ability of the manufacturing system to produce the same type of product by using several different machines. However, although the machines are not the same, however, are able to produce the same type of product. In terms of production planning (scheduling), the problem is known as alternative engine scheduling.

The problem of alternative machine scheduling is a problem that often occurs in manufacturing system industry. With industry performance targets to minimize tardiness, not many manufacturing industries are able to solve it well.

Suppose given a set of jobs I. . Each jobs $i \in I$ has due date D_i and weight of interest w_i. Each jobs $i \in I$ has a number of operations J_i. Suppose H stated the alternative set of machines. Any kind of machine h on slot k has availibility H_{hk}. $H_{ij} \subseteq H$, $H_{ij} \neq \varnothing$ states the alternative set of machines that can perform operations j on job i. Each operation j on job i can be done on one machine $h \in H_{ij}$. P_{ij} states the set of operations that becomes the direct precedence of the operation j pada job i. Operation time j job i on machine h stated by matrics t_{ijh}. Time Horizon is discretized on K unit time. Work on each operation j need operator. Operator availability on slot k stated by P_k.

The expected goal of solving the problem of alternative machine scheduling is to minimize the total weighted tardiness of the manufacturing system.

Research on alternative computing scheduling machines has been done on [1] . In the research, a computation model of single machine operation scheduling is performed using integer programming. Although the computational model is solely for solving single operation problems, it has contributed to integer programming formulations for scheduling computing for alternative machine problems.

Subsequent research on programming for computing solving scheduling is [2]. In this research, programming model for computing scheduling identic parallel machine with objective function of flow time minimization.

Research on the programming of other scheduling computing is [3]. In this research a metaheuristic programming model is produced to solve single machine scheduling case.

Research on integer programming for computation scheduling is also done on [4]. The result of his research is scheduling computing programming at high schools in 11 countries around the world.

The computational research of alternative machine scheduling is also done using heuristic method on [5]. In this research, heuristic computation model is produced for single machine operation alternative problem.

Research on integer scheduling programming is further done by [6]. The result of his research is an integer programming model for multi-project scheduling computing.

Research on the programming model for troubleshooting other machine scheduling problems is [7]. In this study, combined between mathematical programming model and linear programming to solve the problem of scheduling. The model focused on solving a single machine problem with the resulting model is a quadratic programming model.

2. Related Study

The integer programming model of alternative scheduling of operating machines is developed from the integer programming model of alternative scheduling of a single operating machine generated from [1] . From the model formulation, there are three indexes, namely index i, which represents the job to i, index h, which represents the machine to h, and index k, which represents the time slot to k.

Since the model is built for computing single scheduling operating problems, no index represents the operation to.

Therefore, for the case of scheduling alternative machines of compound operating machines in this study, the model was developed by adding index j, representing the operation to j.

Overall the notation consists of: set, index, parameter, decision variable, and measure of performance.

Alternative set of machines that can perform operations j job i is represented by notation of Hij and the set of operations that precedence directly from operation j on job i is represented with the notation of Pij.

Indexes for jobs, operations, machines, and time slots, are represented by notation: j, i, h, and k.

The weighted parameters of job i, Duedate job i, and time horizon are represented by notation: Wi, Di, and K. As for the working time of operation j job i on the machine h, Where h $\in$Hij represented by notation of tijh, Availability parallel operator on time slot k is represented by notation of Pk, and constants indicating the availability of machine h in slot k represented by notation of Hhk.

Decision variables of completion time of job i, completion time of operation j job i, completion time of operation j job i on machine h, Tardiness job i, Earliness job i, operation time j job i, and start operation time j jobs i, respectively represented by notation: Ci, Cij, Cijh, Ti, Ei= tij, and Bij.

As for binary variables 0-1 which states that the operation j job i done on the machine h represented by notation of Xijh, binary variables 0-1, worth it 1 if job i operation j worked on machine h completed in time slot k, worth it 0 if so, represented by notation of Cijhk, and binary variables 0-1, worth it 1 if job i operation j on time slot k use machine h, worth it 0 if so, represented by notation of Xijhk.

Meanwhile, the performance measure on this model is the total weighted tardiness, which is represented by notation Z.

In this integer programming model, there are six underlying assumptions. First, the dependence relationship of a job's operation constitutes directed acyclic graph and each job ends with a single operation; Second, All jobs are considered available on time k=; third, the operation time already includes set up time; fourth, the work of a job nonpreemptive; fifth, Time Horizon K is considered long enough to complete all jobs, Ci $\leq$K for all i; and sixth, each operation is done simultaneously between one machine and one operator.

On the basis of these notations and assumptions, a model formulation was constructed which consisted of the objective formulation and the formulation of the limiting function. The purpose function as the purpose of the study is to minimize total weighted tardiness. The model formulation of the objective function is the same as the model on [1].

Barriers that limit the function of the destination include: operating time constraints, limits on start of operation, delimiters during operation completion, delimiter, machine availability, and carrier restrictions. The restriction formulation is as follows :

$$C_i = C_{ij_i} \ , \forall i \tag{1}$$

$$C_{ig} + t_{ij} \le C_{ij} \ , \forall i, \forall j, \forall g; g \in P_{ij} \tag{2}$$

$$C_{ij} \ge t_{ij} \ , \forall i, \forall j \tag{3}$$

$$t_{ij} = \sum_h t_{ijh} X_{ih} \ , \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{4}$$

$$\sum_h X_{ijh} = 1, \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{5}$$

$$X_{ijh} \in \{0,1\}, \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{6}$$

$$C_{ijh} \ge t_{ijh} \ , \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{7}$$

$$\sum_h C_{ijh} = C_{ij}, \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{8}$$

$$\sum_k C_{ijhk} k = C_{ijh}, \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{9}$$

$$\sum_k C_{ijhk} = 1, \forall i, \forall j, \forall h; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{10}$$

$$C_{ijhk} \in \{0,1\}, \forall i, \forall j, \forall h, \forall k; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{11}$$

$$\sum_{k \le l \le k + t_{jo} - 1} C_{ijhl} = X_{ijhk} \ , \ \forall i, \forall j, \forall h, \forall k; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{12}$$

$$\sum_i \sum_j X_{ijhk} \le H_{hk} \ , \forall h, \forall k; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{13}$$

$$X_{ijhk} \in \{0,1\}, \ \forall i, \forall j, \forall h, \forall k; h \in H_{ij}; H_{ij} \subseteq H; H_{ij} \ne \phi \tag{14}$$

$$\sum_i \sum_j \sum_h X_{ijhk} \le P_k \ , \forall h, \forall k; h \in H_{ij}; H_{ij} \subset H; H_{ij} \ne \phi \tag{15}$$

$$B_{ij} = C_{ij} - t_{ij} + 1, \ \ \forall i, j \tag{16}$$

3. Research Method

The integer programming model that has been produced, tested on a numerical example representing the problem of alternative scheduling problems of compound peripheral operation machine.

The numerical example that will be generated in it contains some jobs that need to be scheduled, where each job consists of several operations; contained several alternative machines; and known duedate, weight and time horizon.

Model testing is done on mathematical programming software Lingo 9. Therefore, the model needs to be changed in the form of mathematical programming Lingo 9. Program verification needs to be done to verify the logic and program script.

Program computing results need to be validated to ensure that the program can run on real systems. The success of the integer programming model is seen from the resulting performance measure, that is the total weighted tardiness.

4. Result and Analysis

In the analysis phase, the resulting model is tested on a numerical example that can represent the problem of alternative scheduling problem of parallel machine compound operation. The numerical examples are as follows:

Tabel 1. Numerical Example

Jobs	Operations	Operation time dan operator simultaneously (days)						Due date (days)	Weight	Time Horison (days)
		1	2	3	4	5	6			
1	1	2	2	-	-	-	-	6	1	16
	2	-	-	3	2	-	-			
	3	-	-	-	-	3	2			
2	1	3	2	-	-	-	-	6	2	
	2	2	3	-	-	-	-			
	3	-	-	3	2	-	-			
	4	-	-	-	-	3	2			
3	1	2	3	-	-	-	-	6	2	
	2	3	2	-	-	-	-			
	3	-	-	2	3	-	-			
	4	-	-	-	-	2	3			
4	1	2	3	-	-	-	-	5	4	
	2	3	3	-	-	-	-			
	3	3	2	-	-	-	-			
	4	-	-	-	-	2	3			
Parallel machines		2	2	1	2	2	2			
Number of operator				8						

Model testing of numerical examples is done using mathematical programming software, Lingo 9. For that, the integer programming model needs to be changed in the formulation of mathematical programming Lingo 9. Here is a mathematical programming formulation Lingo 9 :

```
Model:

Min= @SUM(Job(i):Wi(i)*Ti(i));
@FOR(Job(i):@FOR(JobOperasi(i,j)|j#EQ#Ji(i):Ci(i)=Cij(i,j)));
@FOR(Pred(f,g,i,j):Cij(f,g)+tij(i,j)<=Cij(i,j));
@FOR(JobOperasi(i,j):Cij(i,j)>=tij(i,j));
@FOR(JobOperasi(i,j):@SUM(JobOperasiMesin(i,j,h):tijh(i,j,h))=tij(i,j)
);
@FOR(JobOperasiMesin(i,j,h)|fijh(i,j,h)#GT#0:fijh(i,j,h)*Xijh(i,j,h)=t
ijh(i,j,h));
@FOR(JobOperasi(i,j):@SUM(JobOperasiMesin(i,j,h)|fijh(i,j,h)#GT#0:Xijh
(i,j,h))=1);
@FOR(JobOperasiMesin(i,j,h):@BIN(Xijh(i,j,h)));
@FOR(JobOperasiMesin(i,j,h):Cijh(i,j,h)>=tijh(i,j,h));
@FOR(JobOperasi(i,j):@SUM(JobOperasiMesin(i,j,h):Cijh(i,j,h))=Cij(i,j)
);
@FOR(JobOperasiMesin(i,j,h):
@SUM(JobOperasiMesinSlot(i,j,h,k):Cijhk(i,j,h,k)*k)=Cijh(i,j,h));
@FOR(JobOperasi(i,j):@SUM(JobOperasiMesinSlot(i,j,h,k):Cijhk(i,j,h,k))
=1);
@FOR(JobOperasiMesinSlot(i,j,h,k):@BIN(Cijhk(i,j,h,k)));
@FOR(JobOperasiMesinSlot(i,j,h,k):@SUM(JobOperasiMesinSlot(i,j,h,l)|(k
#LE#l)#And#(l#LE#(k+fijh(i,j,h)-1)):Cijhk(i,j,h,l))=Xijhk(i,j,h,k));
@FOR(MesinSlot(h,k):@SUM(JobOperasiMesinSlot(i,j,h,k):Xijhk(i,j,h,k))<
=Hhk(h,k));
@FOR(Slot(k):@SUM(JobOperasiMesinSlot(i,j,h,k):Xijhk(i,j,h,k))<=8);
@FOR(JobOperasiMesinSlot(i,j,h,k):@BIN(Xijhk(i,j,h,k)));
```

The programming, has been through the verification stage and then run. Computational results are made in table form as follows:

Tabel 2. Scheduling computation solution

Machines	Parallel Machines	Time Horizon (days) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2,2		4,1													
	2	3,1			4,3												
2	1	1,1		3,2													
	2	2,1			4,2												
3	1	3,3															
4	1			1,2													
	2			2,3													
5	1					3,4											
	2						4,4										
6	1					1,3											
	2					2,4											
need of operator		5	5	6	6	5	4	1	0	0	0	0	0	0	0	0	0
operator availibility		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8

Based on the scheduling solution in Table 2, Job 1 operation 1 is scheduled on machine 2 on days 1 and 2; Job 1 operation 2 is scheduled on machine 4 on day 3 and 4; and job 1 operation 3 is scheduled on machine 6 on day 5 and day 6.

Job 2 operation 1 is scheduled on machine 2 on day 1 and 2; job 2 operation 2 is scheduled on machine 1 on day 1 and 2; job 2 operation 3 is scheduled on machine 4 on day 3 and 4; and job 2 operation 4 is scheduled on machine 6 on day 5 and 6.

Job 3 operation 1 is scheduled on machine 1 on day 1 and 2; job 3 operation 2 is scheduled on machine 2 on day 3 and 4; job 3 operation 3 is scheduled on machine 3 on day 1 and 2; and job 3 operation 4 is scheduled on machine 5 on day 5 and to 6.

Job 4 operation 1 is scheduled on machine 1 on day 3 and 4; job 4 operation 2 is scheduled on machine 2 on day 3, 4th and 5th; job 4 operation 3 is scheduled on machine 1 on day 3, to 4, and 5; and job 4 operation 4 is scheduled on machine 5 on day 6 and 7.

The resulting performance measure is the total weighted tardiness of eight days.

5. Discussion

So far, scheduling solutions have been found. To ensure the validity of the solution, the validation process needs to be done. validation is done in three ways. First, the resulting schedule should not be overlapping; the second, the machine must be available according to the scheduling solution; and third, the number of operators must meet for the scheduling.

Based on the first validation, no overlapping jobs and resources; based on the second validation, the availability of the machine is met, based on the third validation, the number of operators used in the time slot $k \leq$ number of machines and operators available. This can be seen in the table, where the number of operators is available there are 8 operators, while the number of operators used on the day to 1 to 7 days each of a number of 5 operators, 5 operators, 6 operators, 6 operators, 5 operators, 4 operators, and 1 operator.

6. Conclusion

The integer programming model for computing alternative scheduling parallel machine compound operations is built with the purpose of weighted total tardiness and 16 constraints. This model is tested with a numerical example of four multi-job scheduling schedules, which are assigned duedate, weight, and time horizon. Model testing is done in mathematical programming software Lingo 9.

Solution of scheduling solution in table 2, Job 1 operation 1 is scheduled on machine 2 on day 1 and 2; Job 1 operation 2 is scheduled on machine 4 on day 3 and 4; and job 1 operation 3 is scheduled on machine 6 on day 5 and to 6. Next Job 2 operation 1 is scheduled on machine 2 on day 1 and 2; job 2 operation 2 is scheduled on machine 1 on day 1 and 2; job 2 operation 3 is scheduled on machine 4 on day 3 and 4; and job 2 operation 4 is scheduled on machine 6 on day 5 and to 6. The Job 3 operation 1 is scheduled on machine 1 on day 1 and 2; job 3 operation 2 is scheduled on machine 2 on day 3 and 4; job 3 operation 3 is scheduled on machine 3 on day 1 and 2; and job 3 operation 4 is scheduled on machine 5 on day 5 and to 6. Where the resulting performance measure is the total weighted tardiness of eight days.

References

[1] Sukendar, I.: 'Integer Linear Programming Model'

[2] Bellenguez-Morineau, O., Chrobak, M., Dürr, C., and Prot, D.: 'A note on $ ${\mathbb {NP}} $ $ NP-hardness of preemptive mean flow-time scheduling for parallel machines', Journal of Scheduling, 2015, 18, (3), pp. 299-304

[3] Thevenin, S., Zufferey, N., and Widmer, M.: 'Metaheuristics for a scheduling problem with rejection and tardiness penalties', Journal of Scheduling, 2015, 18, (1), pp. 89-105

[4] Kristiansen, S., Sørensen, M., and Stidsen, T.R.: 'Integer programming for the generalized high school timetabling problem', Journal of Scheduling, 2015, 18, (4), pp. 377-392

[5] Sukendar, I.: 'HEURISTIC MODEL WITH DISCRITIZED TIME HORIZON FOR SOLVING ALTERNATIVE MACHINE SCHEDULING PROBLEM ON SINGLE OPERATION', 2015

[6] Toffolo, T.A., Santos, H.G., Carvalho, M.A., and Soares, J.A.: 'An integer programming approach to the multimode resource-constrained multiproject scheduling problem', Journal of Scheduling, 2016, 19, (3), pp. 295-307

[7] Della Croce, F.: 'MP or not MP: that is the question', Journal of Scheduling, 2016, 19, (1), pp. 33-42