J. Basic. Appl. Sci. Res., 2(5)5169-5178, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Designing a Mathematical Model for Just-In-Time Preemptive Identical Parallel Machine

Elnaz Nikoofarid[a], Mohammad Kazemi[b], Amin Aalaei[c*], Vahid Kayvanfar[a]

[a] Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran
[b] Department of Industrial Engineering, Birjand University of Technology, Birjand, Iran
[c] Department of Industrial Engineering & Management Systems, Amirkabir University of Technology, Hafez Ave, Tehran, Iran

ABSTRACT

This article presents a new mathematical model for an identical parallel-machines scheduling problem with preemptive jobs in a just-in-time (JIT) environment that minimizes Earliness/Tardiness (E/T) penalties. In ordinary in non-preemptive problems, E/T penalties are a function of the completion time of all the jobs. However, we initiate a preemptive scheduling model where the earliness penalty depends on the start time of a job. The model is linearized by an elaborately–designed procedure to reach the optimum solution. To validate and verify the performance of proposed model, computational results are presented by solving a number of numerical examples.
Keywords: Identical parallel-machines scheduling; Earliness/Tardiness; Just-in-time; Preemption.

1. INTRODUCTION

In the single machine scheduling problem with E/T, a set of jobs, each with an associated due date, has to be scheduled on a single machine. Each job has a penalty per unit time associated with completing before its due date, and a penalty per unit time associated with completing after its due date. Many researches has been extensively done in the recent years to minimize the weighted number of early and tardy jobs in single machine scheduling problem (Davis and Kanet [5]; Hoogeveen and Van De Velde [12]; Wan and Yen [23]; Sourd and Kedad-Sidhoum [20]; Luo et al. [16]; Hendel and Sourd [9]; Esteve et al. [7]; Liao and Cheng [15]). Just-in-time scheduling problems form a well-studied class of multi-criteria scheduling problems. Indeed, these problems with earliness penalties are very useful to represent practical problems, in which perishable goods should be delivered or storage costs cannot be disregarded. Most of these problems are based on tardiness, T_i, and earliness, E_i, computed by: $T_i = \max(0, C_i - d_i)$, $E_i = \max(0, d_i - C_i)$, where, C_i and d_i denote the completion time and due date of job i, respectively. Clearly, a job cannot have a positive tardiness and a positive earliness simultaneously (i.e., a job is either tardy or early). Moreover, earliness and tardiness are often comparable in terms of costs (or penalties) induced by a delivery that is not on time. Therefore, these two criteria are often aggregated into a single criterion, which is called the weighted deviation of the job with respect to its due date, computed by: $f = \sum(\alpha_i \times E_i + \beta_i \times T_i)$, where, α_i and β_i are the unit cost of tardiness and earliness of job i, respectively. Interestingly, the above function can be generalized to express more complex combinations of contradicting criteria and soft or hard constraints on the date a customer should be delivered.

In this paper, we consider the scheduling problem preemptively by n jobs on m parallel identical machines in a JIT environment. In non-preemptive problems, earliness-tardiness costs are related to the job completion time. It means that a tardiness (or earliness) penalty is due if the job completes after (or before) its due date. Allowing preemption, we can interrupt the job and process another job, and then continue the previous job. With preemptive jobs, there is a chance to remove the machine idle time by occupying with other interrupted jobs. On the other hand, taking some jobs' processing time too long and consequently assigning the earliness-tardiness penalty is the difficulty of preemptive jobs. Many researches in minimizing the weighted number of early and tardy jobs are carried out in the literature. Runge and Sourd [19] addressed a new model for the single machine E/T scheduling problem where preemption is allowed. In this model presented interruption costs are based on the work-in-process (WIP) of the job. The WIP costs are based on the differences among the start and completion times of the jobs. This model developed two main advantages over an existing model HS presented by Hendel and Sourd [10]. Primary, HS does not penalize interruption in all cases. And the next advantage is that liberty among the E/T and the WIP costs allows them to design a new timing algorithm with a better time complexity. Also they discussed for several dominance rules and the particular case of the scheduling problem around a common due date. Furthermore, investigated the lower bound for the timing algorithm and shown that a local search algorithm based on their new timing algorithm is sooner than an alike local search algorithm that uses the timing algorithm proposed by Hendel and Sourd [10] has been considered. Hendel et al. [8] considered a single-machine E/T scheduling problem with

***Corresponding Author:** Amin Aalaei, Department of Industrial Engineering & Management Systems, Amirkabir University of Technology, Hafez Ave, Tehran, Iran. Tel.: +98-912-8464608, E-mail: amin_aalaei@aut.ac.ir

preemption, in which each job has two due dates instead of one; one deals with the start time and the other related to completion time of jobs. They investigated a new single machine scheduling problem with earliness and tardiness to capture the just-in-time philosophy, where the earliness costs depend on the start times of the jobs and tardiness costs depend on completion times. They also applied an efficient representation of dominant schedules and introduce a polynomial algorithm to compute the best schedule for a given representation. By using local search algorithm and a branch-and-bound (B&B) procedure, the authors showed there is a very small gap between their results and optimum solutions. Wan and Yen [22] considered a single machine scheduling problem to minimize the total weighted earliness with the minimum number of tardy jobs. They proposed a heuristic and a B&B algorithm and compared these two algorithms for different size problems. Nowicki and Zdrzalka [18] studied on a bi-criteria approach for an *m* parallel identical machines scheduling problem with *n* preemptive jobs. The completion time and processing cost were two objectives of this problem. In the case of identical machines, a greedy algorithm was proposed to solve a problem of least processing cost under the limited completion time. Djellab [6] considered *m* parallel uniform machine scheduling problems to complete *n* preemptive jobs at a later time in order to minimize the makespan. They developed two new heuristics, superior interval order (SUPIO) and inferior interval order (INFIO), to find upper and lower bounds to the makespan, respectively. Chen and Powell [4] studied on Dantzig-Wolfe exact decomposition algorithm for the problem of scheduling *n* jobs on *m* identical parallel machines that minimizes the total weighted E/T with an unrestrictively large common due date. A branch-and-bound algorithm is established for the problem as a set partitioning one, in which each node is the linear relaxation problem of a set partitioning problem with side constraints in the branch-and-bound tree. Their investigated decomposition algorithm is suitable just for problems with up to 60 jobs in a reasonable CPU time. Xing and Zhang [24] considered a parallel machine scheduling problems, in which jobs may be split arbitrarily to continuous sublots. They solved a number of simple cases with independent-job setup times in a polynomial time, and furthermore presented a heuristic method to minimize the makespan. Hiraishi et al. [11] surveyed a machines scheduling problem with *n* non-preemptively jobs on *m* uniform parallel machines to maximize the weighted number of jobs completed exactly at their due dates. They solved this problem in a polynomial time with positive setup times. Azizoglu [1] considered the problem of scheduling *n* preemptive jobs with deadlines on *m* identical parallel machines, in which the objective function is to minimize total completion time. They investigated a polynomial time algorithm for their problem with agreeable deadline for each job and when preemption is allowed. A scheduling problem on uniform parallel machines minimizing the number of JIT jobs was analyzed by Cepek and Sung [3], in which a processing time and a due date are given for each job. But they called a job is JIT if it is exactly completed on its due date. Lushchakova [17] surveyed on a scheduling problem with two uniform parallel machines and *n* preemptive jobs to minimize the mean flowtime where a job becomes available for processing with a given release date, and all the jobs have equivalent processing times. Biskup et al. [2] investigated the problem of scheduling jobs on a specified number of uniform parallel machines to minimize the total tardiness. They presented three algorithms that are general enough for solving numerous problems for finding optimal or near-optimal solutions to minimize the total tardiness. Kravchenko and Werner [13] developed a linear programming model for parallel machines scheduling problems that is solved by a polynomial algorithm. This algorithm transforms a solution of the given problem to an optimal solution. Kravchenko and Werner [14] worked on a preemptive parallel identical machines scheduling problem that minimizes the sum of completion times and presented a polynomial algorithm. Sun and Li [21] presented a problem of processing a set of *n* jobs on two identical parallel machines where preemptive jobs are not allowed and machine maintenance must be done periodically. They proposed two approaches minimizing the makespan and the completion time, respectively.

The rest of this paper is as follows. Section 3 discusses the problem definition, presentation of mathematical nonlinear programming, and its linearization process. The linearization procedure and the linearized model are presented in Section 4. Section 5 shows the numerical examples to validate and verify the performance of proposed model. Finally, conclusion is given in Section 6.

1. Problem formulation

This section presents a new mathematical model for a parallel machines scheduling problem with preemptive jobs in a JIT environment that minimizes the total tardiness-earliness penalties. Then, we linearize this nonlinear programming model.

Consider a non-preemptive parallel machine scheduling problem with *N* jobs on *M* parallel identical machines. Associated with each job i , $i = 1, \dots, N$, are several parameters: P_i , the processing time for job i ; D_i^c , the due date for job i ; β_i , the tardiness cost per unit time if job i completes processing after D_i^c ; and earliness costs as $E_i = max(0, D_i^s - S_i)$; S_i where is the start time of job i ; and $D_i^s = D_i^c - P_i + 1$ is the ideal start time for job i (that is the target start time); and α_i , the earliness cost per unit time if job i starts processing before D_i^s . We assume that the processing times, start times and due dates are integers.

The following assumptions are considered in the presented model.

3.1. Assumptions

- Processing times for all jobs are known and deterministic.
- No job can be processed on more than one machine simultaneously and any machine can process any job.
- Each machine can process only one job at a time and Preemption of jobs is allowable.
- All machines are identical and a job can be processed by any free machine.
- Completing jobs earlier than due dates are impossible.
- Transportation time between machines is negligible.
- Work-in-process inventory is allowed and its associated costs are negligible.
- Machine setup time is negligible and Machines are available throughout the scheduling period (i.e., no breakdown).

3.2. Notations

3.2.1. Subscripts

N	Number of jobs
J	Number of positions
M	Number of machines
i	Index for job (i=1,2,…,N)
j	Index for position (j=1, 2,…,J)
m	Index for machine (m=1, 2,…,M)

3.2.2. Input parameters

P_i	Processing time of job i
D_i^c	Ideal completion time (or due-date) of job i
α_i	Unit earliness penalty of job i
β_i	Unit tardiness penalty of job i
A	An arbitrary big positive number

3.2.3. Decision variables

C_i	Completion time of job i
D_i^s	Ideal start time for job i
X_{imj}	1 if job i on machine m in position j; otherwise, it is zero.
E_i	Earliness of job i
T_i	Tardiness of job i
S_i	Starting time of job i

3.3. Mathematical model

$$\min \sum_{i=1}^{N} (\alpha_i E_i + \beta_i T_i) \tag{1}$$

Subject to:

$$\sum_{m=1}^{M} \sum_{j=1}^{J} X_{imj} = P_i \qquad \forall i; \tag{2}$$

$$\sum_{i=1}^{N} X_{imj} \le 1 \qquad \forall m, j; \tag{3}$$

$$\sum_{m=1}^{M} X_{imj} \le 1 \qquad \forall i, j; \tag{4}$$

$$T_i \ge C_i - D_i^c \qquad \forall i; \tag{5}$$

$$E_i \ge D_i^s - S_i \qquad \forall i; \tag{6}$$

$$C_i = \max_m [\max_j (j \times X_{imj})] \qquad \forall i; \tag{7}$$

$$S_i = \min_m [\min_j [j + A(1 - X_{imj})]] \qquad \forall i; \tag{8}$$

$$X_{imj} \in \{0,1\} \qquad \forall i, m, j; \tag{9}$$

$T_i, E_i \in$ ¥$^{+}$ $\forall i;$ (10)

The objective is to minimize the total weighted earliness and tardiness cost for all jabs. The constraints ensure that jobs start at or after their respective ready times and that jobs do not overlap.

Equality (2) guarantees that the number of positions of all machines in which job *i* is processed is equal to the processing time of job *i*. Inequality (3) necessitates that in position *j* on machine *m* only one job can be processed and Inequality (4) ensure that job *i* in position *j* only one machine can be processed. The tardiness and earliness of each job are calculated by Constraints (5) and (6). Equation (7) and (8) presents the completion time and start time of each job. Constraints (9) and (10) provide the logical binary and non-negativity integer necessities for the decision variables.

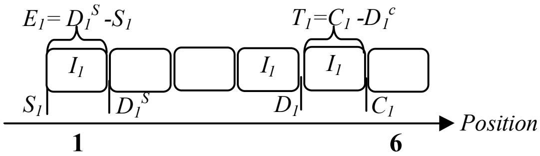

Figure 1. Earliness and Tardiness cost

Figure 1 shows an example to illustrate the calculation way of earliness and tardiness in the first component of objective function and related constraints (4) and (5). As we can see, the completion time of job 1 (C_1) happens after its due date (D_1^c). As a result, tardiness of job 1 happens and its value is equal to $T_1 = C_1 - D_1^c$. Also, the starting time of job 1 (S_1) happens before its ideal starting time (D_1^s). Therefore, earliness of job 1 happens and its value is equal to $E_1 = D_1^s - S_1$. The cost resulted from E/T is obtained by product unitary E/T penalty and the related E/T quantities.

3. Linearization of the proposed model

In this section, we present the linearization procedure and the linearized model.

3.1. Linearization procedure

The linearization procedure that we propose here consists of two steps that are given by the two propositions stated below. Constraints (7) and (8) are non-linear, therefore, these two terms will be linearized using the following auxiliary variables G_{ij}, Q_{ij}, R_{ij}, F_{ij} and B_{ij}. Each proposition for linearization is followed by a proof that illustrates the meaning of each auxiliary (linearization) variable and additional constraints.

Proposition1. The non-linear constraint (7) can be linearized by the following transformation $\max_m \left[\max_j (X_{imj} . j) \right] = Q_{iJ}$, under the following sets of constraints:

$$Q_{i1} = \sum_{m=1}^{M} X_{im1} \qquad \forall i; \qquad (11.1)$$

$$Q_{ij} = Q_{ij-1} . (1 - \sum_{m=1}^{M} X_{imj}) + (j . \sum_{m=1}^{M} X_{imj}) \forall i, j; \quad (11.2)$$

Proof. Consider the following two sections:

(i) In term $\max_m \left[\max_j (X_{imj} . j) \right]$, we find the final position of process for job *i*, thus in constraint (11.2), when for the final position, for example position *j*, $\sum_{m=1} X_{imj} = 1$, then Q_{ij} takes value *j* and for the following positions which are larger than *j*, $\sum_{m=1} X_{imj}$ *s* take value 0, then Q_{ij} finally turns value *j* which implies the final position of process in constraint (11.2).

(ii) The non-linear constraint (11.2) can be linearized by the following transformations $Q_{ij-1} . \sum_{m=1}^{M} X_{imj} = R_{ij}$ and $j . \sum_{m=1}^{M} X_{imj} = F_{ij}$, under the following sets of constraints:

$$R_{ij} \le Q_{ij-1} + A(1 - \sum_{m=1}^{M} X_{imj}) \quad \forall i, j; \qquad (11.3)$$

$$R_{ij} \geq Q_{ij-1} - A(1 - \sum_{m=1}^{M} X_{imj}) \quad \forall i, j; \qquad (11.4)$$

$$R_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \quad \forall i, j; \qquad (11.5)$$

And

$$F_{ij} \leq j + A(1 - \sum_{m=1}^{M} X_{imj}) \quad \forall i, j; \qquad (11.6)$$

$$F_{ij} \geq j - A(1 - \sum_{m=1}^{M} X_{imj}) \quad \forall i, j; \qquad (11.7)$$

$$F_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \quad \forall i, j; \qquad (11.8)$$

This section can be shown for each of the two possible cases that can arise.

1. $\sum_{m=1}^{M} X_{imj} \; . Q_{ij-1} = Q_{ij-1}. \quad \forall i, j;$

Such a situation arises when $\sum_{m=1} X_{imj} = 1$ so, constraints (11.3) and (11.4) implies $R_{ij} \leq Q_{ij-1}$ and $R_{ij} \geq Q_{ij-1}$ and ensures that $R_{ij} = Q_{ij-1}$.

2. $\sum_{m=1} X_{imj} \; .Q_{ij-1} = 0$. Such a situation arises under one of the following three sub-cases:

(a) $\sum_{m=1} X_{imj} = 1$ and $Q_{ij-1} = 0. \quad \forall i, j;$

(b) $\sum_{m=1} X_{imj} = 0$ and $Q_{ij-1} > 0. \quad \forall i, j;$

(c) $\sum_{m=1} X_{imj} = 0$ and $Q_{ij-1} = 0. \quad \forall i, j;$

In all of the three sub-cases given above, R_{ij} takes the value of 0, because in these cases, constraint (11.5) implies $R_{ij} \leq 0$ and ensures that $R_{ij} = 0$. Because R_{ij} has not a strictly positive cost coefficient, the minimizing objective function doesn't ensures that $R_{ij} = 0$. Thus, constraint (11.5) should be added to the mathematical model. The performance of constraints (11.6) - (11.8) is similar to constraints' (11.3) and (11.5).

Proposition2. The non-linear constraint (8) can be linearized by adding the following set of constraints:

$$B_{i1} = \sum_{m=1}^{M} X_{im1} \quad \forall i; \qquad (12.1)$$

$$B_{ij} = B_{ij-1} + (1 - B_{ij-1}).\sum_{m=1}^{M} X_{imj} \quad \forall i, j; \qquad (13.2)$$

$$S_i = J - \sum_{j=1}^{J} B_{ij} + 1 \quad \forall i; \qquad (13.3)$$

Proof. Consider the following two sections:

(i) In term $S_i = \min_m \left[\min_j \left[j + A(1 - X_{imj}) \right] \right]$, we find the first position of process for job i. In constraint (12.2), when for the first position, for example position j, $\sum_{m=1} X_{imj} = 1$, then B_{ij} takes value 1 and since for the following positions which are larger than j, $B_{ij}s$ take value 1, then the summation of $B_{ij}s$ implies the number of positions where job i is work-in-process. Thus S_i returns the first position number in constraint (12.3).

(ii) The non-linear constraint (12.2) can be linearized by the following transformation $B_{ij-1}.\sum_{m=1}^{M} X_{imj} = G_{ij}$,
under the following sets of constraints:

$$G_{ij} \leq B_{ij-1} + A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \quad (12.4)$$

$$G_{ij} \geq B_{ij-1} - A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \quad (12.5)$$

$$G_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \qquad \forall i, j; \quad (12.6)$$

Thus, constraints (12.1) - (12.6) should be added to the mathematical model. The performance of constraints (12.4) - (12.6) is similar to constraints' (11.3) - (11.5).

3.2. The linearized model

We now present the linear mathematical model as follows:

$\min z = (1.1)$

Subject to:

(2), (3), (4), (5), (6), (9), (10)

$$Q_{ij} = Q_{ij-1} - R_{ij} + F_{ij} \qquad \forall i, j; \qquad (7)$$

$$B_{ij} = B_{ij-1} + X_{ij} - G_{ij} \qquad \forall i, j; \qquad (8)$$

$$Q_{i1} = \sum_{m=1}^{M=1} X_{im1} \qquad \forall i; \qquad (11.1)$$

$$C_i = Q_{iJ} \qquad \forall i; \qquad (11.2)$$

$$R_{ij} \leq Q_{ij-1} + A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (11.3)$$

$$R_{ij} \geq Q_{ij-1} - A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (11.4)$$

$$R_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \qquad \forall i, j; \qquad (11.5)$$

$$F_{ij} \leq j + A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (11.6)$$

$$F_{ij} \geq j - A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (11.7)$$

$$F_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \qquad \forall i, j; \qquad (11.8)$$

$$B_{i1} = \sum_{m=1}^{M} X_{imj} \qquad \forall i; \qquad (12.1)$$

$$B_{ij} = B_{ij-1} + \sum_{m=1}^{M} X_{imj} - G_{ij} \qquad \forall i, j; \qquad (12.2)$$

$$S_i = J - \sum_{j=1}^{J} B_{ij} + 1 \qquad \forall i; \qquad (12.3)$$

$$G_{ij} \leq B_{ij-1} + A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (12.4)$$

$$G_{ij} \geq B_{ij-1} - A(1 - \sum_{m=1}^{M} X_{imj}) \qquad \forall i, j; \qquad (12.5)$$

$$G_{ij} \leq A.\sum_{m=1}^{M} X_{imj} \qquad \forall i, j; \qquad (12.6)$$

$$R_{ij} \geq Q_{ij-1} - A(1 - X_{ij}) \qquad \forall i, j; \qquad (12.4)$$

$$G_{ij}, B_{ij}, F_{ij}, Q_{ij}, R_{ij} \geq 0 \qquad \forall i, j; \qquad (13)$$

The number of variables and constraints in the linearized model are presented parametrically in Tables 1 and 2 respectively, based on the variable indices.

Table1. The number of variables in the linearized model

Variable	Count	Variable	Count	Variable	Count
X_{imj}	$N\times M\times J$	Q_{ij}	$N\times J$	R_{ij}	$N\times J$
T_i	N	F_{ij}	$N\times J$	G_{ij}	$N\times J$
E_i	N	B_{ij}	$N\times J$		
Sum = $(N\times M\times J) + 2(N) + 5(N\times J)$					

Table2. The number of constraints in the linearized model

Con.	Count	Con.	Count	Con.	Count
(2)	N	(10)	$2\times N$	(11.8)	$N\times J$
(3)	$M\times J$	(11.1)	N	(12.1)	N
(4)	$N\times J$	(11.2)	N	(12.2)	$N\times J$
(5)	N	(11.3)	$N\times J$	(12.3)	N
(6)	N	(11.4)	$N\times J$	(12.4)	$N\times J$
(7)	$N\times J$	(11.5)	$N\times J$	(12.5)	$N\times J$
(8)	$N\times J$	(11.6)	$N\times J$	(12.6)	$N\times J$
(9)	$N\times M\times J$	(11.7)	$N\times J$	(13)	$5\times N\times J$
Sum = $(N\times M\times J) + 18(N\times J) + (M\times J) + 9(N)$					

4. Computational results

To validate the proposed model and illustrate its various features, numerical example with randomly generated data is solved by branch and bound (B&B) method under Lingo 11.0 software on an Intel® CoreTM2.4 GHz Personal Computer with 4 GB RAM. This example includes three machines and seven jobs. The information related to the example is given in Table 3. Table 3 presents the value of the parameters for each job and contains processing time, due-date and penalty of E/T.

Table3. Job information

Job number	Processing Time	Due date	Ideal Start Time	Earliness Penalty	Tardiness Penalty
1	3	6	4	80$	60$
2	3	5	3	60$	40$
3	4	9	6	80$	90$
4	5	11	7	40$	80$
5	3	11	9	60$	50$
6	8	12	5	30$	60$
7	4	13	10	60$	70$

Table 4 presents the solution obtained for each job and it contains the starting and completion time, number of interruptions and flow time. Furthermore, tardiness/earliness penalty imposed by each job is calculated in Table 4.

Table4. The solution obtained for each job

Job number	Starting Time	Completion Time	Tardiness Penalty	Earliness penalty
1	4	6	0	0
2	3	5	0	0
3	6	9	0	0
4	7	11	0	0
5	9	11	0	0
6	2	12	0	3*30$
7	10	13	0	0

Table5. Objective function and its cost components

OFV	Earliness	Tardiness
90	90	0

The objective function value (OFV) obtained after 15495 iterations in CPU time 9':34" is presented in Table 5. Figure 2 shows the schema of just-in-time in identical parallel machines for this example. It can be seen, the positions in which the jobs are processed, the starting and completion time of each job, as minimum earliness and tardiness cost acquired. In this example, start time job 6 is 2 but idle start time for this job is 5, which causes earliness cost, and the unit earliness penalty of job 6 is 30, therefore the total cost is 3×30=90. We implement the sensitive analysis of model by increasing the unit earliness penalty of job 6 from 30 to 130. The job schedules, objective function and the solution obtained for each job is presented in Figure 3 and Tables 6 and 7, respectively. Increasing in the unit earliness penalty of job 6 causes that model tries to prevent earliness in processing of job 6. As a result, the completion time of job 5 is increased. As we can see, job 6 is processed without any earliness penalty and the completion time of job 5 is increased from 11 to 14. Also the tardiness of job 5 is increased. By comparing the objective function values presented in Tables 5 and 7, we can understand that in spite of processing job 6 without any earliness, increased completion time of job 5 raise the objective function from 90 to 150.

Further to the explained example, we have also solved several numerical examples of different sizes and their results are shown in table 8.

Table6. The solution obtained for each job

Job number	Starting Time	Completion Time	Tardiness Penalty	Earliness penalty
1	4	6	0	0
2	3	5	0	0
3	6	9	0	0
4	7	11	0	0
5	9	14	3*50$	0
6	5	12	0	0
7	10	13	0	0

Table7. Objective function and its cost components

OFV	Earliness	Tardiness
150	0	150

Table8. Several numerical examples and related cost components of objective functions

No. of Jobs	No. of Machines	OFV	Earliness	Tardiness	CPU time
8	2	880	654	226	40':32"
10	2	1128	425	703	1:49':11"
12	3	2340	894	1446	2:22":49"
15	3	3563	1645	1918	5:48':47"
20	4	5642	3475	2167	8:05':52"

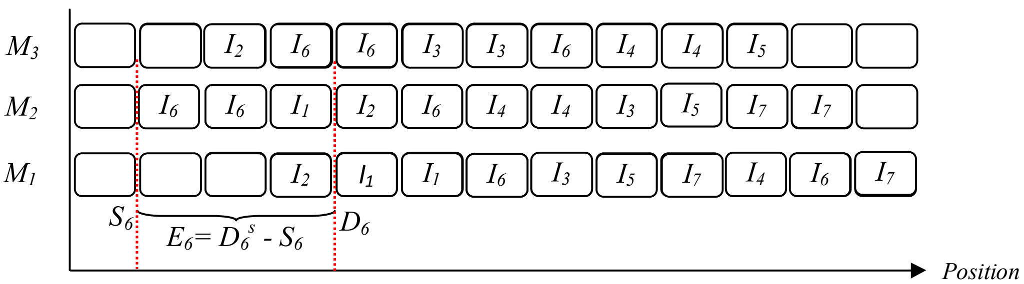

Figure2. Schema of just-in-time in identical parallel machines for this example

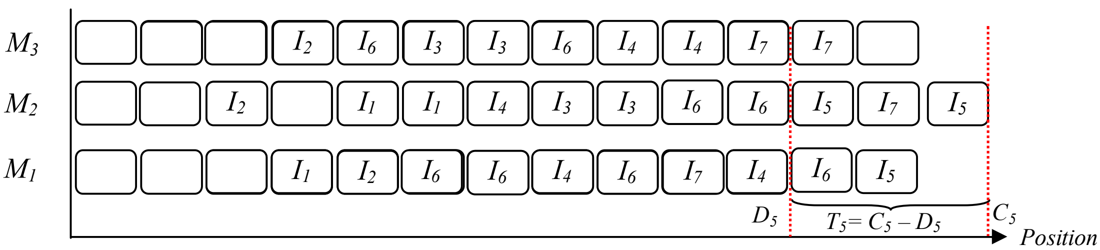

Figure3. Schema of just-in-time in identical parallel machines for this example

5. Conclusion

This paper has presented a new non-linear programming model for a parallel identical machines scheduling problem with preemptive jobs in a just-in-time (JIT) environment. The nonlinear formulation of the proposed model was linearized using an innovative procedure. The performance of the model was illustrated by a numerical example. Sensitive analysis performed on interruption cost illustrated the impact of this feature on the model performance. CPU time required to reach optimal solution for the presented examples shows that obtaining an optimal solution for such hard problems in a reasonable time is computationally intractable. An attractive future research trend is to investigate the preemptive jobs in JIT with parallel uniform or different machines. Also it would be appropriate to consider the problem studied here with the addition of some other assumptions like sequence dependent setup times. It is also interesting to develop heuristics or meta-heuristic algorithms to solve the proposed model for large-sized problems.

REFERENCES

[1] Azizoglu, M., 2003. Preemptive scheduling on identical parallel machines subject to deadlines, *European Journal of Operational Research*, Vol. 148, pp. 205–210.

[2] Biskup, D., 2008. Herrmann, J., Gupta, J.N.D., Scheduling identical parallel machines to minimize total tardiness, *International Journal of Production Economics*, Vol. 115, pp. 134–142.

[3] Cepek, O., Sung, S.C., 2005. A quadratic time algorithm to maximize the number of just-in-time jobs on identical parallel machines, *Computers & Operations Research*, Vol. 32, pp. 3265–3271.

[4] Chen, Z., Powell, W.B., 1999. A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem, *European Journal of Operational Research*, Vol. 116, pp. 220–232.

[5] Davis, J.S., Kanet, J.J., 1993. Single-machine scheduling with early and tardy completion costs. *Naval Research Logistics* Vol. 40, pp. 85-101.

[6] Djellab, Kh., 1999. Scheduling preemptive jobs with precedence constraints on parallel machines, *European Journal of Operational Research*, Vol. 117, pp. 355-367.

[7] Esteve, B., Aubijoux, C., Chartier, A., T'kindt, V., 2006. A recovering beam search algorithm for the single machine Just-in-Time scheduling problem, *European Journal of Operational Research*, Vol. 127, pp. 798–813.

[8] Hendel, Y., Runge, N., Sourd, F., 2009. The one-machine just-in-time scheduling problem with preemption, *Discrete Optimization*, Vol. 6, No. 1, pp. 10-22.

[9] Hendel, Y., Sourd, F., 2006. Efficient neighborhood search for the one-machine earliness/tardiness scheduling problem. *European Journal of Operational Research* Vol. 173, pp. 108-119.

[10] Hendel, Y., Sourd, F., 2005. The single machine just-in-time scheduling problem with preemptions. In: MISTA 2005: proceedings of the 2nd multidisciplinary international conference on scheduling: theory and applications. pp. 140–148.

[11] Hiraishi, K., Levner, E., Vlach, M., 2002. Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs, *Computers & Operations Research*, Vol. 29, pp. 841–848.

[12] Hoogeveen, J.A., Van de Velde, S.L., 1996. A branch-and-bound algorithm for single-machine earliness/tardiness scheduling with idle time. *INFORMS Journal on Computing* Vol. 8, pp. 402-412.

[13] Kravchenko, S.A. and Werner, F., 2009. On a parallel machine scheduling problem with equal processing times, *Discrete Applied Mathematics*, Vol. 157, No. 4, pp. 848-852.

[14] Kravchenko, S., Werner, F., 2009. Preemptive scheduling on uniform machines to minimize mean flowtime, *Computers & Operations Research*, Vol. 36, pp. 2816-2821.

[15] Liao, C.-J., & Cheng, C.-C., 2007. A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Computers and Industrial Engineering* Vol. 52, pp. 404-413.

[16] Luo, X., Chu, Ch., Wang, Ch., 2006. Some dominance properties for single-machine tardiness problem with sequence-dependent setup times. *International Journal of Production Research,* Vol. 44, pp. 3367–3378.

[17] Lushchakova, I.N., 2006. Two machine preemptive scheduling problem with release dates, equal processing times and precedence constraints, *European Journal of Operational Research*, Vol. 171, pp. 107–122.

[18] Nowicki, E., Zdrzalka, S., 1995. A bicriterion approach to preemptive scheduling of parallel machines with controllable job processing times, *Discrete Applied Mathematics*, Vol. 63, pp. 237-256.

[19] Runge, N., Sourd, F., 2009. A new model for the preemptive earliness–tardiness scheduling problem. *Computers & Operation Research*, Vol. 36, pp. 2242-2249.

[20] Sourd, F., Kedad-Sidhoum, S., 2003. The one-machine scheduling with earliness and tardiness penalties. *Journal of Scheduling*, Vol. 6, pp. 533-549.

[21] Sun, K., Li, H., 2010. Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines, *International Journal of Production Economics*, vol. 124, no. 1, pp. 151-158.
[22] Wan, G., Yen, B.P.-C., 2009. Single machine scheduling to minimize total weighted earliness subject to minimal number of tardy jobs, *European Journal of Operational Research*, Vol. 195, pp. 89–97.
[23] Wan, G., Yen, B.P.C., 2002. Tabu search for single machine with distinct due windows and weighted earliness/tardiness penalties. *European Journal of Operational Research*, Vol. 142, pp. 271-281.
[24] Xing, W. and Zhang, J., 2000. Parallel machine scheduling with splitting jobs, *Discrete Applied Mathematics*, Vol. 103, pp. 259-269.