

Bounds for the Permutation Flowshop Scheduling Problem with Exact Time Lags while Minimizing the Total Earliness and Tardiness

Imen Hamdi, Taïcir Loukil*

Abstract- We consider the problem of n -jobs scheduling in an m -machine permutation flowshop with exact time lags between consecutive operations of each job. The exact time lag is defined as the time elapsed between every couple of successive operations of the same job which is equal to a prescribed value. The aim is to find a feasible schedule that minimizes the total tardiness and earliness. We propose a mathematical formulation, which is then solved by running the commercial software CPLEX to provide an optimal solution for small size problems. As the problem is shown to be strongly NP-hard, we propose two upper bounds and two lower bounds useful for large size problems. We then evaluate their effectiveness through an extensive computational experiment.

Keywords: Scheduling; Exact time lags; Tardiness and earliness; Lower bounds; Upper bounds

I. INTRODUCTION

We consider the permutation flowshop scheduling problem in which the elapsed time between each couple of operations of each job must be equal to a prescribed value ($\theta_{i,k}$). The permutation flowshop problem with exact time lags is a particular case of the permutation flowshop with minimal and maximal time lags. It corresponds to the case where the minimal and maximal time lags are equal. According to Fondrevelle et al. [1], the exact time lag constraints generalize the classical no-wait constraints, for which the waiting time between successive operations equals 0. The no-wait requirement can be found in industries where products must be processed continuously through the stages in order to prevent degradation. Then, the problem can be formulated as follows: a set $i \in \{1, 2, \dots, n\}$ of jobs have to be processed on a set $k \in \{1, 2, \dots, m\}$ of machines. Each machine can process one job at a time and preemption is not allowed. An exact time lag between each couple of operations is added. For each job, we define $p_{i,k}$ the processing time of job i on machine k , $\theta_{i,k}$ the exact time lag of job i between machine k and machine $k+1$, and d_i the due date of job i . The aim is to find a feasible schedule that minimizes the total tardiness and earliness.

*Manuscript received December 23, 2013; revised January 8, 2014. This work is supported by LOGIQ Research Unit and by ISSAT Kasserine Institute, Tunisia.

I.H. is with the High Institute of Industrial Management, University of Sfax, 3018 Tunisia (e-mail: imen.hamdi2007@yahoo.fr).

T.L. is with High Institute of Industrial Management, University of Sfax, 3018 Tunisia (e-mail: taicir.loukil@fsegs.rnu.tn).

Tardiness of job i is defined as $T_i = \max(C_{i,m} - d_i, 0)$ where $C_{i,m}$ is the completion time of job i on the last machine m . Earliness of job i is defined as $E_i = \max(d_i - C_{i,m}, 0)$. Recently, many researches are interested in analyzing the earliness on the manufacturing costs. According to Chandra et al. [2], some reasons for reducing earliness be limited storage space for finished goods, and the limited shelf life of products as in the case of chemicals and pharmaceuticals industries. Our objective in this research is then to determine a sequence of all jobs that insure minimum total tardiness and earliness. Since Koulamas [3] has shown the NP-hardness of $F||\sum T$ problem for $k \geq 3$, the problem $F_\pi|\theta_{i,k}|\sum(E+T)$ is also NP-hard. A permutation flowshop scheduling problem with exact time lags ($F_\pi|\theta_{i,k}|L$) is studied by Fondrevelle et al. [1], some special cases are studied and a dominance relation is provide. Also lower and upper bounds are developed and integrated in a branch and bound procedure.

Scheduling models with this criterion are compatible especially with Just In Time production where jobs are scheduled to complete as close as possible to their due dates. Most of existing researches that deal with this criterion are done for the single machine scheduling (see Kanet and Sridharan [4], Valente [5], Abdul razaq and Potts [6], Li [7], Valente and Alves [8], and Liaw [9]) who consider the earliness/tardiness problem with equal release dates and no idle time where the main proposed approach is the branch and bound. Also some heuristics are developed, the performance of various heuristics, including dispatching rules, a greedy procedure and a decision theory algorithm, are analyzed in (Valente and Alves [8]). Metaheuristic approaches are proposed by Feldmann and Diskup [10].

Common due dates is pioneered by many researches e.g. Kanet [11], and Baker and Scudder [12] where two kinds of due dates are defined: the unrestrictive one if its optimal value has no influence on the optimal sequence and according to Feldmann and Diskup [10] for a given due date which is greater than or equal to the sum of processing times of all jobs is always unrestrictive. However, if the common due date may influence on the optimal sequence of jobs, it is called restrictive one. Recently Janiak et al. [13] introduce the common due window concept, they present a survey of studies on scheduling problems with a common due window assignment and earliness/tardiness penalty functions. They turn to analyze the classical models with job-independent and job-dependent earliness/tardiness penalty functions and some other more complicated models. Then, they de-

scribe the computational complexity of the problems and the main features of the approaches developed to solve them.

Focusing on the flowshop environment, some researches are done while minimizing the total tardiness and earliness. For 2-machines flowshop problem with common due dates problems, we can mention the research of Sung and Min [14]. For the m -machine flowshop problems; recently, Bulbul et al. [15] generalize the tardiness/ earliness problem for the flowshop problem with intermediate inventory holding costs, they formulate it as an integer problem. They develop heuristics to minimize the total costs, then, they exploit the duality between Dantzig-Wolfe reformulation and lagrangian relaxation to enhance these heuristics. Chandra et al. [2] treat the permutation flowshop problem with earliness and tardiness penalties, they provide a partial characterisation of the optimal solution and develop a comprehensive approach for solving the problem over the entire range of due dates.

Furthermore, variety of flowshop problems with tardiness earliness-based objective are adressed in the literature. The no-wait permutation flow shop scheduling ($F|No\ Wait|\sum T + E$) problem is studied by Ning et al. [16]. They propose an heuristic, and then the NEH algorithm is used to get the optimized solutions. Debora and Ernesto [17] consider no storage constraints and with blocking inprocess. They present Mixed-Integer models which are then evaluated and compared using commercial software. Zhu and Heady [18] develop a mixed integer programming in a multi-machine scheduling problem that can easily provide the optimal solution to problems involving about nine jobs and three machines. Arabameri and Salmasi [19] investigate the no-wait flowshop sequence-dependent setup time scheduling problem with minimization of weighted earliness and tardiness penalties. They propose a mixed integer linear programming model. Then, they develop several meta-heuristic algorithms based on tabu search and particle swarm optimization algorithms, and they generate a timing algorithm to find the optimal schedule and calculate the objective function value of a given sequence.

In this paper, we consider the permutation flowshop scheduling problem with exact time lags while minimizing the total earliness and tardiness. This problem is not studied sufficiently over the literature. We propose a mathematical formulation useful to generate an optimal solution by running the software CPLEX. Then, two new upper bounds are proposed. Also, two lower bounds are developed where the first one is based on relaxing the integrality constraints and the second one on summing two derived tardiness lower bound and earliness lower bound. The organization of the remainder of this paper is as follows: In Section 2, we present the proposed mathematical formulation. In Section 3, we present the developed upper bounds. In section 4, the two derived lower bounds are presented. Then, computational results are reported in Section 5, and finally in section 6 we discuss concluding remarks.

II. MATHEMATICAL FORMULATION

The considered problem is characterized by n jobs being processed on m machines always in the same order while an exact time lag $\theta_{i,k}$ is defined between each couple of operations of each job i . Preemptions are not allowed, that is, when a job starts to be processed on a machine, it cannot be interrupted.

The used notations are described as follows:

A. Decision Variables

- $X_{i,j} = 1$ if job i is scheduled in position j , 0 otherwise $\forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, n\}$
- $C_{j,k}$: completion time of job in position j on machine k , $\forall j \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, m\}$

B. Data

- $p_{i,k}$: processing time of job i on machine k , $\forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, m\}$
- $\theta_{i,k}$: exact time lag between the stopping of the k^{th} operation and the starting of the $(k+1)^{th}$ operation of the job i .
- d_i : due date of job i

Then the mathematical formulation is presented as follows

$$\text{Minimize } \sum_j E_j + T_j \quad (1)$$

$$\sum_{i=1}^n X_{i,j} = 1 \quad \forall j \quad (2)$$

$$\sum_{j=1}^n X_{i,j} = 1 \quad \forall i \quad (3)$$

$$C_{j,k+1} = C_{j,k} + \sum_{i=1}^n X_{i,j} (p_{i,k+1} + \theta_{i,k}) \quad \forall j \text{ and } \forall k \quad (4)$$

$$C_{j,k} + \sum_{i=1}^n (p_{i,k} X_{i,j}) \leq C_{j+1,k} \quad \forall j \text{ and } \forall k \quad (5)$$

$$T_j \geq C_{j,m} - \sum_{i=1}^n (d_i X_{i,j}) \quad \forall j \text{ and } \forall i \quad (6)$$

$$E_j \geq \sum_{i=1}^n (d_i X_{i,j}) - C_{j,m} \quad \forall j \text{ and } \forall i \quad (7)$$

$$C_{j,k}, T_j, \text{ and } E_j \geq 0 \quad \forall j \text{ and } \forall k \quad (8)$$

$$X_{i,j} \in \{0, 1\} \quad (9)$$

Constraints (2) and (3) are classical assignment constraints which ensure that each job can only be allocated to a sequence position and that each sequence position can only be filled by one job $\forall i \in \{1, 2, \dots, n\}$ and $\forall j \in \{1, 2, \dots, n\}$. Constraints (4) involved in obtaining the completion time of each job with respect to the

precedence constraint and the exact time lags between operations $\forall j \in \{1, 2, \dots, n\}$ and $\forall k \in \{1, 2, \dots, m-1\}$. Constraints (5) state that the completion time of job in position j plus its processing time on machine k have to be smaller than or equal to the completion time of the next job in the same machine $\forall j \in \{1, 2, \dots, n-1\}$ and $\forall k \in \{1, 2, \dots, m\}$. With constraints (6) and (7), we define the tardiness and the earliness for each job $\forall i \in \{1, 2, \dots, n\}$ and $\forall j \in \{1, 2, \dots, n\}$. Constraints (8) force the tardiness, the earliness and the completion time to be positive values $\forall j \in \{1, 2, \dots, n\}$ and $\forall k \in \{1, 2, \dots, m\}$. Then, constraints (9) specify $X_{i,j}$ as a binary variable which is equal to 1 if the job i is assigned to position j and 0 else $\forall i \in \{1, 2, \dots, n\}$ and $\forall j \in \{1, 2, \dots, n\}$.

III. UPPER BOUNDS

In this section we propose two upper bounds which are based on two different rules: the Earliest Due Date (EDD) and the Shortest Sum Processing Time (SSPT) rule.

- EDD rule: this rule is known and it states to arrange the jobs in nondecreasing order of the due dates d_i ($d_{\pi(1)} \leq d_{\pi(2)} \leq \dots \leq d_{\pi(n)}$). Then, by obtaining a sequence we calculate the total tardiness by applying the algorithm described later.
- SSPT rule: This rule is similar to *SPT*, but instead of using the individual job processing time on each machine, we consider for each job the total processing times of a job plus the exact time lags between each consecutive couple of operations.

The first step in the proposed algorithm consists in obtaining the sequence of jobs by using the rule (*EDD* or *SSPT*). This sequence is scheduled in the second step. In the third step, an exchange step between each two adjacent jobs is done in the seek of enhancing the obtained result. We begin by the two first adjacent jobs and so on. Each time, we compare the new found value of the total tardiness and earliness with the previous one. If there is no enhancement, no exchange is done. Then, the first sequence can be modified. The algorithm is detailed as follows:

Algorithm

Step 1. Determine the scheduling sequence (π) by using the rule (*EDD* or *SSPT*)

Step 2. Schedule the sequence of jobs (π) as follows:

2.1 The first job is scheduled as soon as possible

$$C_{\pi(1),1} = p_{\pi(1),1}$$

For $k = 1$ to $m-1$, do

$$C_{\pi(1),k+1} = C_{\pi(1),k} + \theta_{\pi(1),k} + p_{\pi(1),k+1}$$

end

2.2 Schedule the other jobs as soon as possible

For $i = 2$ to n , do

$$C_{\pi(i),1} = C_{\pi(i-1),1} + p_{\pi(i),1}$$

For $k = 1$ to $m-1$, do

$$C_{\pi(i),k+1} = \max\{C_{\pi(i),k} + \theta_{\pi(i),k}, C_{\pi(i-1),k+1}\} + p_{\pi(i),k+1}$$

end

end

2.3 Make sure that the exact time lag constraints are satisfied

For $k = m-1$ to 1

$$\text{if } C_{\pi(i),k+1} - C_{\pi(i),k} > \theta_{\pi(i),k} + p_{\pi(i),k+1}$$

$$\text{then, } C_{\pi(i),k+1} = C_{\pi(i),k} + \theta_{\pi(i),k} + p_{\pi(i),k+1}$$

end

2.4 Determine the absolute deviation of each job's completion time and its due date, then the total earliness and tardiness (*ET*)

For $i = 1$ to n , do

$$X_i = |C_{i,m} - d_i|$$

$$\text{then } ET = \sum_{i=1}^n X_i$$

end

Step 3. An exchange operation between each two adjacent jobs.

For $i = 1$ to $n-1$, do

$$\text{temp} \leftarrow \pi(i)$$

$$\pi(i) = \pi(i+1)$$

$$\pi(i+1) = \text{temp}$$

3.1 Determine the new value of the total earliness and tardiness (*ET1*)

For $i = 1$ to n

$$ET_1 = \sum_{i=1}^n |C_{i,m} - d_i|$$

end

3.2 We compare the new value (*ET1*) with the previous value (*ET*). If the new value is enhanced ($ET_1 < ET$), then we keep the permutation and the total tardiness and earliness value will be equal to (*ET1*). Else, we restart with the previous positions and the total earliness and tardiness value still equal to (*ET*).

If $ET_1 \geq ET$, then

$$\text{temp} \leftarrow \pi(i)$$

$$\pi(i+1) = \pi(i)$$

$$\pi(i) = \text{temp}$$

$$\text{Total tardiness and earliness} = ET$$

end

end

IV. LOWER BOUNDS

The permutation flowshop scheduling problem with exact time lags is not studied sufficiently over the literature. Fondrevelle et al. [1] derive lower and upper bounds which are then integrated in a branch and bound procedure to minimize the maximum lateness. Developing tight lower bounds seems to be crucial mainly when with the branch and bound algorithm as they allow an efficient pruning in the search tree. Here, we are concerned by minimizing the total earliness and tardiness, which is an *NP-hard* problem in the strong sense.

In this section we develop two lower bounds. The first one is a Linear Program (LP) relaxation which consist in relaxing the integrality constraints. The second one is based on developing a lower bound on the total tardiness and a lower bound on the total earliness, and then by

summing both of bounds we find the lower bound on the total earliness and tardiness.

A. LP relaxation (LB1)

An Integer Linear Program (ILP) is a LP such that the variables must take integer values. The ILP is an eloquent to formulate optimization problems and they can catch a large number of combinatorial optimization problems. Obtaining an optimal solution for the ILP is an NP-hard problem. A relaxation of the original problem by removing the integrality constraints seems to be interesting.

The integrality constraints are easy to state but make the problem much more difficult to solve. The LP relaxation consists in removing the integrality constraints, that is allowing variables to take on non-integral values and replaced by appropriate continuous constraints. Therefore, the NP-hard optimization problem can be reduced to a linear program which is solvable in polynomial time.

B. The sum of tardiness lower bound and earliness lower bound (LB2)

As it was described previously, this lower bound consists in adding a lower bound on the total tardiness (LBT) and a lower bound on the total earliness (LBE). Even when adding a lower bound on the total tardiness and an estimated value of the total earliness, the sum of both values will result in a lower bound on the total tardiness and earliness. They are described as follows.

A lower bound on the total tardiness can be given as : $LBT = \sum_{i=1}^n [\max_{1 \leq k \leq m} \{ \sum_{s=1}^i p_{s,k} + \sum_{l=1}^{k-1} \min_i \{ p_{i,l} \} + \sum_{l=1}^{k-1} \min_i \{ \theta_{i,l} \} \} - d_i]^+$. Obviously it consists in determining a lower bound on the completion time of each job, by subtracting the due date and finding a positive value we can derive the tardiness of each job.

The lower bound on the total earliness can be obtained by relaxing the restriction on the starting time of the first machine at time 0. We begin by determining an Estimated Completion Time (ECT) for each job for an EDD sequence as : $ECT_i = \sum_{k=1}^{m-1} (p_{i,k} + \theta_{i,k}) + p_{i,m} + \sum_{s=1}^{i-1} p_{s,1}$ (here i and s note the position in the sequence). Then we can determine the Estimated Earliness Value (EEV) for each early job as: $EEV_i = [d_i - ECT_i]^+$. Let x note the minimal earliness value among all the job's earliness values; then by translating the corresponding job to the right, its earliness will be equal to 0 and the earliness values of the other will be minimized. By translating even one job to the right, all the schedule have to be translated to respect the exact time lags between machines which is possible in this case as we relax the restriction of beginning processing time at time 0 for the first machine. Finally the lower bound on the earliness value is calculated as: $LBE = \sum_{i=1}^n [d_i - (ECT_i + x)]^+$. Then, the second lower bound on the total earliness and tardiness is obtained as: $LB2 = LBT + LBE$

V. COMPUTATIONAL RESULTS

We conduct a computational analysis to evaluate the performance of the proposed bounds. The mathematical formulation is tested by running CPLEX 11., and the algorithms are implemented with MATLAB 7.6. The computational experiments are run on a DELL PC/2.20GHz with 4.00Go RAM.

Instances are generated as the same way in Fondrevelle et al. [1] (for one of the tested classes): The processing times are generated from a uniform distribution between 20 and 50 and the time lags in the interval [0, 100] . For the due dates, we follow the method proposed by Potts and Wassenhove [20], they are generated in a range [Px, Py] where P is a lower bound on the makespan and defined as

$$P = \max_{1 \leq k \leq m-1} \{ \sum_{i=1}^n p_{i,k} + \sum_{l=1}^{m-1} \min_i (p_{i,l} + \theta_{i,l}) \}$$

Where $x = 1 - T - R/2$ and $y = 1 - T + R/2$. T is the tardiness factor, which is set to 0.2 and 0.6; while R is the due date range that assumed the values 0.25 and 0.75. The different used combinations of (T, R) are: (1)=(0.2, 0.25); (2)=(0.2, 0.75); (3)=(0.6, 0.25); and (4)=(0.6, 0.75). We set four different configurations for number of jobs $n \in \{5, 10, 15, 20\}$, and two different configurations for number of machines $m \in \{5, 10\}$.

For each problem size, ten instances are generated for each combination of T and R; then the average of the total tardiness and earliness is determined. A total of 200 runs are executed. The results are summarized in the following Table 1. To evaluate the proposed upper and lower bounds, we determine the percentage deviation from the optimal solution, which is calculated as $\% = \frac{Op-LB}{LB} \times 100$ for the lower bounds and as $\% = \frac{UB-Op}{Op} \times 100$ for the upper bounds.

Table 1: Computational results

(n, m)	(T, R)	Op		LB%		UB%	
		ET	CPU	LB1	LB2	EDD	SSPT
(5, 5)	(1)	277	0.07	17.62	1.51	0.08	2.22
	(2)	323	0.09	13.50	0.67	1.01	1.61
	(3)	611	0.18	7.32	0.82	1.25	1.15
	(4)	623	0.09	10.0	0.21	0.20	0.91
(10, 5)	(1)	450	0.16	18.61	2.19	0.87	0.91
	(2)	683	0.51	15.53	1.91	0.04	2.13
	(3)	1489	1.45	12.01	0.56	0.26	1.14
	(4)	1749	1.86	9.11	0.06	0.32	1.05
(15, 5)	(1)	841	0.39	13.52	2.01	0.72	2.78
	(2)	833	3.44	12.11	1.34	0.45	1.24
	(3)	2565	314.1	9.50	0.57	1.08	0.17
	(4)	2703	197.3	10.33	0.41	0.03	0.29
(15, 10)	(1)	3390	644.4	18.42	0.04	0.12	1.25
	(2)	2213	636.4	14.31	1.09	0.73	2.37
	(3)	8555	813.1	8.25	2.76	0.01	1.63
	(4)	9766	245.2	11.20	1.12	0.23	1.22
(20, 5)	(1)	2210	145.8	16.31	1.26	1.05	1.53
	(2)	2894	914.5	10.28	2.03	0.42	2.23
	(3)	3344	866.6	9.32	1.59	0.07	0.62
	(4)	3788	1232.4	11.91	0.77	0.05	0.11

For each problem size, it is obvious that the best results are found with the smallest values of tardiness parameters. The larger the values of the tardiness parameters, the smaller will be the range of due dates. This result in larger optimal value for total tardiness and earliness. The proposed lower bound (LB2) is shown to be good as the percentage deviation doesn't exceed 2.76 and so better than (LB1) for which the percentage deviation range between 7.3 and 18.4. Also the upper bounds are shown to be very good, just the one based on the *EDD* rule is better for almost all the problems. As the completion times are sorted in non decreasing order for any sequence, then when sort the due dates also in non decreasing order will result in the minimal absolute deviation between each job's completion time and its due date. On the other hand, we can conclude that the deviation percentage from the optimal solution decreases with the increasing values of the parameters. The lower bounds are more better for problems with potentially high total tardiness and earliness. We expect that they can provide the optimal solution with larger size problems and with increasing tardiness parameters values.

The problems with larger values of tardiness parameters consume more *CPU* time (*second*), and the *CPLEX* is shown to be unable to solve problems with up the size ($n=20, m=5$) in one hour (3600 s). The upper bounds and lower bound (LB2) are solved in less than 1 second and we couldn't distinguish a meaningful difference between all problems in the *CPU* time.

An other experiment is done to distinguish the effect of the exact time lags on the deviation percentage for large size problems. Here, we consider the deviation of the second lower bound (LB2) from the upper bound provided by the procedure based on the *EDD* rule ($\% = \frac{UB-LB}{LB} \times 100$). For each problem size, we define four classes of problems according to the time lags interval. The first class is a classical permutation flowshop problem without time time lags. For the other classes the exact time lags are generated from the interval $[0, \theta_{i,k}]$ where $\theta_{i,k} \in \{25, 50, 100\}$. The parameters (T, R) are fixed to (0.2, 0.25).

Table 2: Effect of the time lags on the deviation

(n, m)	(50, 10)	(100, 20)	(200, 50)	(500, 50)	
	0	19.1 %	15.1 %	16.3 %	10.3 %
(θ)	25	15.4 %	14.1 %	13.1 %	10.7 %
	50	16.2 %	12.2 %	13.5 %	15.8 %
	100	13.7 %	11.5 %	10.3 %	10.6 %

From Table 2, we can confirm that the exact time lags have a significant effect on the deviation percentage. The higher value is found with the classical case where the exact time lags is equal to zero, then it decreases with the increasing values of the time lags intervals and with the increasing problem size. The developed lower bounds are more performant with higher values of the time lags.

VI. CONCLUSION

The permutation flowshop problem with exact time

lags to minimize the total earliness and tardiness is considered in this paper. We propose a mathematical formulation, two upper bounds and two lower bounds. An extensive computational experiment is done to evaluate the effectiveness of these bounds. It is measured by the percentage deviation from the optimal solution. The results reveal that the proposed procedures are shown to be very efficient to derive good results by using some parameters. Also, the results show that the tardiness parameters and the exact time lags intervals have an important effect on the objective values and the deviation percentage. When the exact time lags and the parameters (T, R) increase, the objective value increases and the deviation percentage decreases.

As we mentioned previously in this paper, few research works deal with shop scheduling problems with exact time lags. Several directions for further work can be of major interest: It could be interesting to develop other lower bounds which could be tight by using a decomposition approach.

References

- [1] J. Fondrevelle, A. Allahverdi, A. Oulamara, M.C. Portmann, "Permutation flowshop with exact time lags to minimize maximum lateness". Rapport de recherche LORIA 2005.
- [2] P. Chandra, P. Mehta, D. Tirupati, "Permutation flow shop scheduling with earliness and tardiness penalties", International Journal of Production Research Vol. 47, No. 20, 2009, pp. 5591-5610.
- [3] C. Koulamas, "The Total Tardiness Problem: Review and Extensions", Operations Research, Vol. 42, No. 6, 1994, pp.1025-1041.
- [4] J. Kanet, V. Sridharan, "Scheduling with inserted idle time: Problem taxonomy and literature review", Operations Research, Vol. 48, No. 1,2000, pp. 99-110.
- [5] J.M.S. Valente, R.A.F.S. Alves, "Improved lower bounds for the early/tardy scheduling problem with no idle time", Journal of the Operational Research Society, Vol. 56, 2005, pp. 604-612.
- [6] T.S Abdul-Razaq, C.N. Potts, "Dynamic programming state-space relaxation for single machine scheduling", Journal of the Operational Research Society Vol. 39, No. 2,1988, pp. 141-152.
- [7] G. Li, "Single machine earliness and tardiness scheduling", European Journal of Operational Research, Vol. 96, No. 3, 1997, pp. 546-558.
- [8] J.M.S. Valente, "Improved Lower Bounds for the Single Machine Earliness/Tardiness Scheduling Problem with Release Dates", International Journal of Operations Research, Vol. 2, No. 2, 2005, pp. 9-16.
- [9] C.F. Liaw, C.F., "A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem", Computers and Operations Research, Vol. 26, No. 7, 1999, pp. 679-693.

- [10] M. Feldmann, D. Diskup, "Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches", *Computers & Industrial Engineering*, Vol. 44, No. 2, 2003, pp. 307-323.
- [11] J.J Kanet, "Minimizing the average deviation of job completion times about a common due date", *Naval Res. Log. Q.*, Vol. 28, No. 4, 1981, pp. 643-651.
- [12] K.R. Baker, G.D. Scudder, "On the assignment of optimal due dates", *Journal of the Operational Research Society*, Vol. 40, No. 1, 1989, pp. 93-95.
- [13] A. Janiak, T. Kwiatkowski, M. Lichtenstein, "Scheduling problems with a common due window assignment: A survey", *International Journal of Applied Mathematics and Computer Science*, Vol. 23, No. 1, 2013, pp. 231-241.
- [14] C. Sung, J. Min, "Scheduling in a two-machine flowshop with batch processing machine(s) for earliness/tardiness measure under a common due date", *European Journal of Operational Research*, Vol. 131, No. 1, 2001, pp. 95-106.
- [15] K. Bülbül, P. Kaminsky, C. Yano, "Flow Shop Scheduling with Earliness, Tardiness, and Intermediate Inventory Holding Costs", *Naval Research Logistics*, Vol. 51, No. 3, 2004, pp. 407-445.
- [16] W. Ning, Y. Lei, H.W. Zhong, G.R. Yan, "Researches on the No-Wait Permutation Flow Shop Earliness and Tardiness Scheduling Problem", *Journal of Beijing University of Posts and Telecom*, Vol. 32, 2009, pp. 72-76.
- [17] P.R. Dehora, G.B. Ernesto, "Mixed-Integer Programming Models for Flowshop Scheduling Problems Minimizing the Total Earliness and Tardiness", *Springer Optimization and Its Applications*, 2012, pp. 91-105.
- [18] Z. Zhu, R.B. Heady, "Minimizing the sum of earliness/tardiness in multi-machine scheduling : a mixed integer programming approach", *Computers and Industrial Engineering*, Vol. 38, No. 2, 2000, pp. 297-305.
- [19] S. Arabameri, N. Salmasi, "Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem", *Computers and Industrial Engineering*, Vol. 64, No. 4, 2013, pp. 902-916.
- [20] C.N Potts, L.N. Van Wassenhove, "decomposition algorithm for the single machine total tardiness problem", *Operations Research Letters*, Vol. 1, No. 5, 1982, pp. 177- 181.

Date of modification: 28-02-2014

Description of the changes: I did modifications in the results of LB2 and the two upper bounds in Table 1 and Table 2 as i did many experiments with different data and i discover later that the data used to obtain these bounds are different from the ones used to obtain the optimal solution and LB1. So, i correct this mistake.