Journal of Industrial Engineering International (2019) 15 (Suppl 1):S223–S229
https://doi.org/10.1007/s40092-019-00331-1

ORIGINAL RESEARCH

MILP models and valid inequalities for the two-machine permutation flowshop scheduling problem with minimal time lags

Imen Hamdi[1,2] · Saïd Toumi[1,2,3]

Received: 10 May 2019 / Accepted: 10 October 2019 / Published online: 22 October 2019

Abstract
In this paper, we consider the problem of scheduling on two-machine permutation flowshop with minimal time lags between consecutive operations of each job. The aim is to find a feasible schedule that minimizes the total tardiness. This problem is known to be NP-hard in the strong sense. We propose two mixed-integer linear programming (MILP) models and two types of valid inequalities which aim to tighten the models' representations. One of them is based on dominance rules from the literature. Then, we provide the results of extensive computational experiments used to measure the performance of the proposed MILP models. They are shown to be able to solve optimally instances until the size 40-job and even several larger problem classes, with up to 60 jobs. Furthermore, we can distinguish the effect of the minimal time lags and the inclusion of the valid inequalities in the basic MILP model on the results.

Keywords Flowshop · Total tardiness · Time lags · MILP models · Valid inequalities

Introduction

This paper addresses the two-machine permutation flowshop scheduling problem with minimal time lags where the objective is minimizing the total tardiness. This environment is characterized by n jobs ($i \in \{1, \ldots, n\}$) being processed non-preemptively on two machines $M1$ and $M2$ in this order according to a processing time a_i on $M1$ and b_i on $M2$. For each job, an amount of time must be elapsed before the processing of the second operation on the second machine. This time must be greater than or equal to a non-negative minimal time lag value denoted $\theta_i^{\min}$. Each job has a due date d_i. According to the standard notation provided by Graham et al. (1979), the considered problem is denoted as $F2|\theta_i^{\min}|\sum_i T_i$.

Investigating this problem is motivated by its practical relevance to the production environments. The time lags constraints are used frequently in both service and industrial fields. For example in the chemistry field, the chemical reactions with different processing times may be represented by time lags (Chu and Proth 1996). Also, such constraints appear during the sequence of harvesting operations in the agriculture field (Foulds and Wilson 2005). As well as in a training center for which the sequence of modules taught must require some temporal constraints. Many other real examples are given by Deppner (2004).

Since the publication of Dell'Amico (1996) of his original paper on the two-machine flowshop and jobshop with time lags, this problem has attracted an ever-growing attention in the operations research field. Then, this study was extended to consider various problems with time lags while optimizing different criteria. It is worth noting that the great majority of the researches address the makespan objective (Fondrevelle et al. 2006; Yu et al. 2004; Hamdi and Loukil 2011) in spite that the due date-based criteria are the most met in the real situations.

Interestingly, these criteria and mainly the total tardiness criterion is considered widely with the classical two-machine flowshop problem where most of the researches are

✉ Imen Hamdi
imen.hamdi2007@yahoo.fr

Saïd Toumi
s.toumi@mu.edu.sa

1 MODILS Lab, Faculty of Economics and Management, University of Sfax, Sfax, Tunisia

2 High Institute of Transport and Logistics, University of Sousse, Sousse, Tunisia

3 Department of Business Administration, College of Business Administartion, Majmaah University, Al Majmaah, Saudi Arabia

characterized by applying the branch and bound algorithm accompanied by proposing and improving lower bounds and dominance properties (Kim 1993; Pan and Fan 1997; Pan et al. 2002; Schaller 2005).

However, only few researches considered the time lags. Yu et al. (2004) show the NP hardness of the problem even if all processing times are equal to one. Later, Hamdi et al. (2015) developed a branch and bound algorithm to reach an optimal solution by using some developed lower bounds and dominance criteria. These latter are developed to establish precedence constraints between jobs in an optimal schedule. Some of them will be used later in this research. Also, they proposed some computationally efficient heuristics. Nearly at the same time, Hamdi and Loukil (2015b) extend the research about the permutation flowshop problem with minimal time lags by considering the case of m-machine and optimizing the total number of tardy jobs criterion. They proposed some heuristic procedures based on known and new rules. Then interestingly, they developed lower bounds based on Moore's algorithm and logic-based Benders decomposition as a hybrid approach that mix the MILP and the constraint programming.

It is known that the MILP models can be used to obtain optimal schedules for small size problems. They are often classified based on the choice of the decision variables. Over the literature, a wide variety of permutation flowshop scheduling problems were addressed while using the due date-based criteria. Kharbeche and Haouari (2013) proposed compact mixed-integer programming models and valid inequalities for the two-machine scheduling problem. They developed a lower bound that consistently outperforms the best bound from the literature. Then, their computational study reveals that most of the 30-job hard instances are optimally solved using their proposed MILP models. Also, Ronconi and Birgin (2012) proposed mixed-integer linear programming formulations for the *m*-machine scheduling problem in the flowshop environment with unlimited and zero buffer, for the minimization of total earliness and tardiness.

Even with the existence of other types of the time lags like the maximal time lags (denoted $\theta_{i,k}^{\max}$in the case of *m*-machine to indicate the maximal time lag of the job *i* between the two machines *k* and $k+1$) and the exact time lags (denoted $\theta_{i,k}$), the linear programming method was the scope of numerous papers. Hamdi and Loukil (2015a) proposed a mathematical formulation for the problem $F_\pi|\theta_{i,k}^{\min}, \theta_{i,k}^{\max}|\sum_i T_i$ which was useful to distinguish the effect of the time lags on the results. Moreover, it was used to evaluate the performance of some proposed heuristic methods by determining the percentage deviation from the optimal solution provided by running the proposed MILP with the CPLEX solver. In the same context, Dhouib et al. (2013) proposed a mixed-integer mathematical programming formulation for the permutation flowshop problem with minimal and maximal time lags while the objective is to hierarchically minimize two criteria, the primary criterion is the minimization of the number of tardy jobs and the secondary one minimizes the makespan. It is then used to solve optimally problems until the size 20-jobs and 5-machine and for comparison reason with a proposed simulated annealing algorithm. Later, Hamdi and Loukil (2017) proposed three different MILP models to solve the problem $F_\pi|\theta_{i,k}|\sum_i (E_i + T_i)$ which show that there is a little difference between them when referring to the CPU time in spite that the number of decision variables is almost the same for all the formulations. Thereafter, the obtained optimal solution is used to evaluate the performance of some proposed upper and lower bounds until the size 20-job and 5-machine.

It is worth mentioning that the MILP models are used frequently by the recent researches that concerned by solving scheduling problems with due date-based criteria (i.e., Moreno-Camacho et al. 2018; Huang et al. 2013; Keshavarz et al. 2019).

The contribution of this paper is twofold: First, we propose two mathematical formulations distinguished by the used decision variables which are: the completion time variables and the idle time variables. Second, we present two sets of valid inequalities that can be used to improve the proposed MILP resolution as the CPU time required to solve the problem can be significantly reduced. To this aim, we exploit some dominance rules from the literature for the first set, and we propose a lower bound on the completion time for the second set. Then, these sets are introduced separately and together to the completion time-based formulation to distinguish their effect on the resolution performance. Thus, a total of five MILP formulations are computationally compared by using the CPLEX solver. To the best of our knowledge, this is the first attempt to investigate MILP models for $F2|\theta_i^{\min}|\sum_i T_i$.

The remainder of this paper is organized as follows: mathematical formulations of the considered problem are presented in "MILP models2" section. In "Valid inequalities3" section, we present the two types of valid inequalities: valid inequalities based on dominance rules from the literature, and position-based valid inequalities. Then, computational results are reported in "Computational results4" section. Finally, we discuss concluding remarks in "Conclusion5" section.

MILP models

This section lists two MILP formulations: the completion time-based formulation and the idle time-based formulation, respectively. The considered problem is characterized by *n* jobs being processed on 2 machines always in the same order while a minimal time lag $\theta_i^{\min}$is defined between each couple of operations of each job i. Each machine carries only one job at a time, we assume that all jobs are independent and available for processing at time 0, and preemptions are not allowed .

A common feature for both formulations is that they are based on the positional decision variables. That is, the resolution of the problem consists in assigning each job to only one position in the schedule that leads to the optimal value of the total tardiness.

Completion time-based formulation

This formulation is based on the following decision variables:

- $X_{i,j}$: binary variable that takes value 1 if job i is scheduled in position j, 0 otherwise $\forall i,j \in \{1,2,\dots,n\}$
- $C_{j,k}$: completion time of job in position j on machine k, $\forall j \in \{1,2,\dots,n\}$, $\forall k \in \{1, 2\}$
- T_j: tardiness of the job in position j $\forall j \in \{1,2,\dots,n\}$

The used data are given as follows:

- a_i: processing time of job i on machine 1, $\forall i \in \{1,2,\dots,n\}$
- b_i: processing time of job i on machine 2, $\forall i \in \{1,2,\dots,n\}$
- $\theta_i^{\min}$: minimal time lag between the two machines of each job i $\forall i \in \{1,2,\dots,n\}$

Then the MILP formulation is presented as follows

$$Minimize \sum_j T_j \tag{1}$$

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

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

$$C_{1,1} = \sum_{i=1}^{n} X_{i,1} a_i \tag{4}$$

$$C_{j,1} + \sum_{i=1}^{n} (X_{i,j+1} a_i) \le C_{j+1,1} \quad \forall j = 1,\dots,n-1 \tag{5}$$

$$C_{j,2} \ge C_{j,1} + \sum_{i=1}^{n} X_{i,j} \left(b_i + \theta_i^{\min}\right) \quad \forall j = 1,\dots,n \tag{6}$$

$$C_{j,2} + \sum_{i=1}^{n} (X_{i,j+1} b_i) \le C_{j+1,2} \quad \forall j = 1,\dots,n-1 \tag{7}$$

$$T_j - C_{j,2} + \sum_{i=1}^{n} (d_i X_{i,j}) \ge 0 \quad \forall j = 1,\dots,n \tag{8}$$

$$C_{j,k},\ T_j \ge 0 \quad \forall j = 1,\dots,n,\ \ \forall k = 1,2 \tag{9}$$

$$X_{i,j} \in \{0,\ 1\} \quad \forall i,j = 1,\dots.,n \tag{10}$$

The objective (1) is to minimize the total tardiness. Constraints (2) ensure that each position can be occupied by only one job; however constraints (3) ensure that each job can only be assigned to a sequence position. Constraints (4) determine the competition time of the job in the first position on the first machine. Constraints (5) find the completion time of the other jobs on the first machine. Constraints (6) find the completion time of each job with respect to the minimal time lags between operations and the precedence constraints. Constraints (7) define the precedence constraint of each two successive positions in the second machine. Constraints (8) find the tardiness value of each job. Constraints (9) impose the tardiness and the completion time to be positive values. Then, constraints (10) define $X_{i,j}$ as a binary variable which is equal to 1 if the job i is assigned to position j and 0 else. In this formulation, the number of binary variables is n^2, the number of integer variables is $3n$, and the number of constraints is $9n - 1$.

Idle time-based formulation

This formulation is based on the same data and variables defined in the previous formulation, just we use the variable I_j instead of $C_{j,k}$; it represents the idle time on machine 2 between the completion time of the job in position j and the starting time of the job in position $j + 1$. Then, the formulation is presented as follows.

$$Minimize \sum_j T_j \tag{11}$$

$$\sum_{i=1}^{n} X_{i,j} = 1 \quad \forall j = 1,\dots,n \tag{12}$$

$$\sum_{j=1}^{n} X_{i,j} = 1 \quad \forall i = 1,\dots,n \tag{13}$$

$$\sum_{k=1}^{j-1} \sum_{i=1}^{n} \left(b_i + \theta_i^{\min}\right) X_{i,k} - \sum_{k=2}^{j} \sum_{i=1}^{n} \left(a_i + \theta_i^{\min}\right) X_{i,k} + \sum_{k=2}^{j} I_k \ge 0 \quad \forall j = 2,\dots,n \tag{14}$$

$$\sum_{i=1}^{n}(a_i + b_i + \theta_i^{\min})X_{i,1} - \sum_{i=1}^{n} d_i X_{i,1} \le T_1 \tag{15}$$

$$\begin{aligned}&\sum_{i=1}^{n}(a_i + \theta_i^{\min})X_{i,1} + \sum_{k=1}^{j}\sum_{i=1}^{n} b_i X_{i,k} + \sum_{k=2}^{j} I_k \\ &\quad - \sum_{i=1}^{n} d_i X_{i,j} \le T_j \quad \forall j = 2, \dots, n\end{aligned} \tag{16}$$

$$\begin{aligned}&\sum_{j=1}^{n}\sum_{i=1}^{n} a_i X_{i,j} + \theta_n^{\min} \le \sum_{i=1}^{n}(a_i + \theta_i^{\min})X_{i,1} \\ &\quad + \sum_{j=1}^{n}\sum_{i=1}^{n} b_i X_{i,j} + \sum_{j=1}^{n-1} I_j\end{aligned} \tag{17}$$

$$I_j \ge 0 \ \forall j = 1, \dots, n-1 \tag{18}$$

$$T_j \ge 0 \ \forall j = 1, \dots, n \tag{19}$$

$$X_{i,j} \in \{0, 1\} \ \forall i,j = 1, \dots, n \tag{20}$$

The objective (11) and the first two constraints (12) and (13) are defined as same in the previous MILP model. Constraints (14) state that the idle time of the job in position j is greater than or equal to the completion time of the job in position j on the first machine minus the completion time of the job in position $(j-1)$ on the second machine. Constraints (15) and (16) determine the tardiness value of job in position 1 and the other jobs from position 2 until position n, respectively. Constraints (17) are used to express the relationship among several times such as the processing times, the idle times and the time lags among jobs and the two machines. In this formulation, the number of binary variables is n^2, the number of integer variables is $2n-1$, and the number of constraints is $6n-1$.

Valid inequalities

The two proposed MILP models include almost the same number of decision variables, which can't solve large sized problems. Then by adding cuts, their performance can be enhanced by reducing the number of decision variables and then the search space. Two types of valid inequalities are presented in this section.

Valid inequalities based on dominance rules

For this first type of the valid inequalities, a preprocessing step is needed to define a set of precedence constraints by using some dominance rules. Let this set denote $\xi = \{(i,\ s) \in J^2$: there exists an optimal schedule such that a job s is processed after a job i , then the valid inequalities to be added to the proposed MILP model is:

$$\sum_{j=1}^{n} jX_{i,j} + 1 \le \sum_{j=1}^{n} jX_{i,s}, \quad (i,\ s) \in \xi \tag{21}$$

where $\sum_{j=1}^{n} jX_{i,j}$ indicates the position index of the job i as the binary variable $X_{i,j}$ takes value 1 if and only if job i is assigned to position j. Then, these inequalities are derived by using some specified rules. These rules are proposed by Hamdi et al. (2015) for the same studied problem which aim to sequence a job i in an optimal sequence if some properties holds. They are provided as follows:

Rule 1: For any two adjacent jobs $(i, s) \in J^2$, if

(a) $min\{a_s + \theta_s^{\min},\ b_i + \theta_i^{\min}\} \ge a_i + \theta_i^{\min}$
(b) $b_i + \theta_i^{\min} \le b_s + \theta_s^{\min}$
(c) $d_i \le d_s$

Then there exists an optimal schedule such that i is processed before s.

Rule 2: For job i, if there is a job s satisfying

(a) $a_i + \theta_i^{\min} + b_i \ge a_s + \theta_s^{\min} + b_s$
(b) $\theta_i^{\min} \le \theta_s^{\min}$
(c) $b_i - d_i \le b_s - d_s$

Then, there exists an optimal schedule such that i is not the first job of the sequence. So that, the variable $X_{i,1}$ will be eliminated from the formulation.

Position-based valid inequalities

These inequalities are based on the fact that the completion time of a job in position j is greater than a lower bound (λ_j). Then, we have:

$$C_{j,2} \ge \sum_{i=1}^{n} \lambda_j X_{i,j} \quad \forall j = 1, \dots, n \tag{22}$$

The lower bound value (λ_j) can be calculated easily by using the following proposed formula:

$$\lambda_j = \min_i\{a_i\} + \min_i\{\theta_i^{\min}\} + j \times \{\frac{\sum_i b_i}{n}\} \quad \forall j = 1, \dots, n$$

Here, $\frac{\sum_i b_i}{n}$ is the mean value of the processing times on the second machine; then by relaxing the minimal time lags constraints (as we consider only the minimal value) and adding the minimal value of processing time on the first machine (a_i), we obtain a lower bound on the completion time of the job in position 1. Therefore, to determine the lower bound on the completion time of each job in position j (λ_j), we should multiply $\frac{\sum_i b_i}{n}$ by $j\ \forall j = 1, \ldots, n$ to satisfy the condition $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ in accordance with $C_{1,2} < C_{2,2} < \cdots < C_{n,2}$.

Computational results

Computational experiments are done to assess the effectiveness of the proposed models, mainly the enhancement provided by adding the cuts and the effect of varying the minimal time lags intervals. These models are tested by running the CPLEX 11 solver on a DELL PC/2.20 GHz with 4.00 Go RAM. The tests are conducted on a set of generated instances following the scheme described in Hamdi and Loukil (2015b). The processing times and the minimal time lags are generated from a uniform distribution between 20 and 50 and $[0, \theta^{\min}]$, respectively where $\theta^{\min} \in \{0, 7, 14\}$. The due dates are generated as $d_i = P \times D_{range}$, where P is a lower bound of the makespan which is given as follows: $P = \min_i\{a_i + \theta_i^{\min}\} + \sum_{i=1}^{n} b_i$ and $D_{range} = [0.8, 1.2]$. The number of jobs n is taken to be equal to 10, 15, 20, 25, 40, and 60 jobs. For each combination of n and $\theta^{\min}$, five instances are generated and the average value of the total tardiness is determined. The upper limit of the CPU time for solving a problem is set to 3600 s.

We test the following different formulations:

F_1: Completion time-based formulation
F_2: Idle time-based formulation
F_3: F_1+ valid inequality based on dominance rules
F_4: F_1+ position-based valid inequality
F_5: F_1+ the two sets of inequalities

The results are displayed in Table 1. For each tested problem, we provide the CPU time and the number of nodes required to solve it. The number between parenthesis indicates the number of unsolved instances.

From the above table, we observe the following points:

- In spite that both MILP models include $O(n^2)$ binary variables and $O(n)$ integer variables, the idle-based formulation requires a bit more number of nodes for almost all the sizes, thus, it consumes more CPU time to solve problems.
- It is obvious that the five models are able to solve optimally problems with up to n =40 in less than one hour, and even some instances with up to 60 jobs.
- Increasing the minimal time lags intervals has a distinguishable impact on increasing the number of nodes and the CPU time.
- We could prove the effet of adding constraints on tightening the search space as the number of nodes and the CPU for the three last models tend to be decreased while

Table 1 Impact of the valid inequalities and the minimal time lags on the results

n	$\theta^{\min}$	F_1		F_2		F_3		F_4		F_5	
		Nodes	CPU	Nodes	CPU	Nodes	CPU	Nodes	CPU	Nodes	CPU
10	0	1.152	0.04	2.540	0.04	2.290	0.04	1.210	0.02	872	0.02
	7	920	0.03	2.180	0.08	1.652	0.04	1.089	0.02	914	0.02
	14	2.230	0.14	4.271	1.82	3.121	0.04	1.826	0.04	890	0.03
15	0	4.029	1.18	5.320	1.20	3.540	1.17	2.720	0.04	1.581	0.03
	7	3.244	2.11	7.211	2.16	3.768	1.15	2.542	1.10	2.044	0.07
	14	5.720	2.25	9.017	3.67	6.543	2.50	3.204	1.17	3.000	1.10
20	0	29.276	2.80	41.432	2.97	14.546	2.18	19.768	2.70	15.536	1.89
	7	30.671	3.78	40.713	4.21	20.622	2.66	27.756	2.85	20.429	2.50
	14	35.810	4.31	55.715	5.32	27.783	2.32	29.077	2.86	29.008	2.75
25	0	97.879	8.87	180.661	10.49	95.670	6.81	85.880	5.43	80.545	5.11
	7	109.220	9.10	217.319	10.67	110.989	9.21	93.940	7.21	87.443	6.72
	14	178.702	10.76	275.552	12.63	118.901	9.80	112.850	9.40	100.420	9.06
40	0	1255.61	29.14	1060.34	20.64	1409.44	33.29	816.25	17.28	410.88	11.47
	7	2190.17	45.46	3019.78	49.29	1780.28	47.12	1970.27	50.32	1849.22	38.91
	14	5478.27	67.82	7267.22	70.40	4817.32	58.88	5192.68	60.75	3455.18	41.69
60	0	17,562.278	192.18	27,361.913	210.75 (2)	10,462.667	87.41	11,378.410	160.29	8658.919	128.19
	7	24,445.612	331.39 (3)	68,431.224	410.65 (3)	24,556.901	208.54 (1)	22,798.016	200.48 (2)	18,678.138	158.10
	14	65,122.679	427.10 (3)	91,245.119	481.45 (3)	39,901.778	228.50 (2)	42,560.186	269.10 (2)	28,924.715	170.26 (1)

Table 2 Ratio values of CPU time

F	$\theta^{\min}$	n						Average value
		10	15	20	25	40	60	
F_3	0	1.00	1.00	1.28	1.30	0.87	2.19	1.41
	7	0.75	1.84	1.42	0.98	0.96	1.58	
	14	3.50	0.90	1.85	1.09	1.15	1.86	
F_4	0	2.00	29.5	1.04	1.63	1.68	1.19	3.13
	7	1.50	2.00	1.40	1.26	0.90	1.65	
	14	3.50	1.92	1.50	1.14	1.11	1.58	
F_5	0	2.00	39.3	1.48	1.73	2.48	1.49	5.54
	7	1.50	30.14	1.51	1.35	1.16	2.09	
	14	4.60	2.04	1.56	1.18	1.62	2.50	

adding the valid inequalities. The model F_5, characterized by adding the two sets of the inequalities, is the most efficient as it is less time-consuming than the models F_3 and F_4.

To confirm this last point and to assess the improvement induced by the included type of inequalities to the basic MILP model, we determine for each of the last three formulations (F_3, F_4, and F_5), the ratio of the CPU time required by each problem to the CPU time required by this problem in the original formulation (F_1). Then, the results are shown in Table 2.

The last column in Table 2 indicates the average value of the calculated ratio values for each formulation (F_3, F_4, and F_5) by considering all n and $\theta^{\min}$. Then, adding the two valid inequalities can induce a great improvement for the original formulation by reducing significantly the CPU time until 5.5 times.

Conclusion

In this paper, we proposed two mathematical formulations for the two-machine permutation flowshop problem with minimal time lags scheduling problem. Two sets of valid inequalities are proposed, where the first one is based on some dominance rules from the literature and the second one is based on a lower bound of the job completion time. Different ways are used to integrate these inequalities to the completion time-based model which leads to compare five mathematical models. As of our knowledge, it is the first paper that proposes and compares the computational performance of some mathematical formulations while considering the minimal time lags constraints. There was noticeable improvement in terms of reducing the search space and the time-consuming required to solve the problem mainly the model F_5 characterized by adding the two types of the valid inequalities.

References

Chu C, Proth JM (1996) Single machine scheduling with chain structured precedence constraints and separation time windows. IEEE Trans Robot Autom 12(6):835–844

Dell'Amico M (1996) Shop problems with two machines and time lags. J Oper Res 44(5):777–787

Deppner F (2004) Ordonnancement d'atelier avec contraintes temporelles entre operations. Ph.D. Thesis, Institut National Polytechnique de Lorraine

Dhouib E, Teghem J, Moalla Loukil T (2013) Lexicographic optimization of a permutation flow shop scheduling problem with time lag constraints. Int Trans Oper Res 20(2):213–232

Fondrevelle J, Oulamara A, Portmann MC (2006) Permutation flowshop scheduling problems with maximal and minimal time lags. Comput Oper Res 33(6):1540–1556

Foulds LR, Wilson JM (2005) Scheduling operations for the harvesting of renewable resources. J Food Eng 70(3):281–292

Graham R, Lawler E, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 5:287–326

Hamdi I, Loukil T (2017) The permutation flowshop scheduling problem with exact time lags to minimize the total earliness and tardiness. Int J Oper Res 28(1):70–86

Hamdi I, Loukil T (2015a) Minimizing total tardiness in the permutation flowshop scheduling problem with minimal and maximal time lags. Oper Res Int J 15(1):95–114

Hamdi I, Loukil T (2015b) Upper and lower bounds for the permutation flowshop scheduling problem with minimal time lags. Optim Lett 9(3):465–482

Hamdi I, Oulamara A, Loukil T (2015) A branch and bound algorithm to minimize total tardiness in the two machine flowshop problem with minimal time lags. Int J Oper Res 23(4):387–405

Hamdi I, Loukil T (2011) Minimizing the makespan in the permutation flowshop problem with minimal and maximal time lags. In: Proceeding of the 2011 IEEE international conference of communications, computing and control applications (CCCA'11) 03-05 Mars 2011 a Hammamet, Tunisie

Huang JD, Liu1 JJ, Chen QX and Mao N (2013) Mixed integer fomulation to tardiness objectives in a flow shop with two batches-processing machines and release date. In: CIE43 proceedings, 16–18 October 2013, The University of Hong Kong

Keshavarz T, Salmasi N, Varmazyar M (2019) Flowshop sequence-dependent group scheduling with minimisation of weighted earliness and tardiness. Eur J Ind Eng 13(1):54–80

Kharbeche M, Haouari M (2013) MIP models for minimizing total tardiness in a two-machine flow shop. J Oper Res Soc 64(5):690–707

Kim YD (1993) A new branch-and-bound algorithm for minimizing mean tardiness in two-machine flowshops. Comput Oper Res 20(4):391–401

Moreno-Camacho CA, Montoya-Torres JR, Vélez-Gallego MC (2018) A comparison of mixed-integer linear programming models for workforce scheduling with position-dependent processing times. Eng Optim 50(6):917–932

Pan JC-H, Fan ET (1997) Two-machine flowshop scheduling to minimize total tardiness. Int J Syst Sci 28(4):405–414

Pan JC-H, Chen J-S and Chao C-M (2002) Minimizing tardiness in a two-machine flowshop. Comput Oper Res 29(7):869–885

Ronconi DP, Birgin EG (2012) Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. Just-in-Time Syst 23:91–105

Schaller J (2005) Note on minimizing total tardiness in a twomachine flowshop. Comput Oper Res 32(12):3273–3281

Yu W, Hoogeveen H, Lenstra JK (2004) Minimising makespan in a two machine flowshop with delays and unit time operations is NP-hard. J Sched 7(5):333–348