Yugoslav Journal of Operations Research
32 (2022), Number 1, 45–59
DOI: https://doi.org/10.2298/YJOR210615023A

BRANCH AND BOUND ALGORITHM FOR SOLVING THE TOTAL WEIGHTED TARDINESS CRITERION FOR THE PERMUTATION FLOWSHOP SCHEDULING PROBLEM WITH TIME LAGS

Fatmah ALMATHKOUR
Department of Statistics & Operations Research, Kuwait University
fatmah.almazkoor@ku.edu.kw

Omar BELGACEM
MODILS Laboratory, Faculty of Economics and Management, University of Sfax, Tunisia
omar.belgacem.css@gmail.com

Said TOUMI
Department of Business Administration, College of Business Administration, Majmaah University, Al Majmaah 11952, Saudi Arabia
s.toumi@mu.edu.sa / saidtoumy@gmail.com

Bassem JARBOUI
Higher Colleges of Technology, Abu Dhabi, UAE
bassem_jarboui@yahoo.fr

Received: June 2021 / Accepted: September 2021

Abstract:: This paper deals with the permutation flowshop scheduling problem with time lags constraints to minimize the total weighted tardiness criterion by using the Branch and Bound algorithm. A new lower bound was developed for the flowshop scheduling problem. The computational experiments indicate that the proposed algorithm provides good solution in terms of quality and time requirements.

Keywords: Branch & Bound Algorithm, Total Weighted Tardiness, Time Lags, Permutation Flowshop Scheduling Problem.

MSC: 90B35, 64N10, 70M50, 83C15.

1. INTRODUCTION

In this paper we study the total weighted tardiness criterion in permutation flowshop scheduling problem with minimal time lags, where the objective is to find a job sequence that minimizes the time lags. The problem is denoted as:

$$F_2|\theta_i^{min}|\sum W_j T_j \tag{1}$$

Minimizing the total weighted tardiness is a difficult problem. Although Johnson's theorem is applicable, it is necessary to consider only the schedules where the sequence of jobs is the same on each machine. No constructive algorithm comparable to Johnson is known. Johnson procedure doesn't find an optimal solutions for the criterion.
The weighted tardiness can be obtained by an argument that follows the Johnson rule. To properly define the total weighted tardiness criterion, let:

- $A_{[i]}$: Jobs that are executed on the machine M1.
- $B_{[i]}$: Jobs that are executed on the machine M2.
- Let $X_{[i]}$ the idle time before the operation $B_{[i]}$ on the second machine.
- Let

$$Y_{[j]} = \sum_{i=1}^{j} A_{[i]} - \sum_{i=1}^{j-1} B_{[i]}. \tag{2}$$

Then, the total weighted tardiness is:

$$TWT_{[1]} = A_{[1]} + B_{[1]} = X_{[1]} + B_{[1]} \tag{3}$$

$$TWT_{[2]} = A_{[1]} + B_{[1]} = X_{[2]} + B_{[2]}. \tag{4}$$

$$TWT_j = \sum_{i=1}^{j} B_{[i]} + \sum_{i=1}^{j} X_{[i]} = \sum_{i=1}^{j} B_{[i]} + max(Y_1, Y_2, \dots, Y_j). \tag{5}$$

In the environment of industrial and agricultural production as well as services, Scheduling problems with time lags constraint is applicable. Several studies have investigated the time lags constraint its application to solve various real problems. Such as, [11] reports on two practical harvesting case studies where the context is practiced minimum and maximum time lags on the commercial enterprises in Australia and New Zealand. Then, [2] have used the harvesting of renewable resources on the agricultural context for commercial enterprises and imposed minimum and maximum time lags. Also, [4] studied minimum and maximum time separating successive jobs and showed the importance of the problem in medical practice.
The optimization flowshop with time lags problems are in general difficult to solve.

Several methods are used to find a satisfactory answer to these problems. Scheduling problems are treated by optimization methods that consider the data of the problem as constraints to meet and offer an optimal and feasible solution. The optimality of the data is measured against criteria and objectives established by the higher level of decision [12].
In the literature, many authors tried solving the permutation flowshop scheduling problem with different constraints and for different criteria using a branch and bound algorithm [23], [5], [22], [24], [25], [1]. But few studies deal with a flowshop problem with time lags. The first research to solve the problem with time lags for two machine flowshop and jobshop is given by [6]. Many authors [9], [16], [14], [19], [18], [20], [21], [13], [15] and [17] used an exact methods for solving this problem. Let consider the work of [9] in which the researchers aim to minimize the makespan criterion for m machines with Branch and Bound algorithm. [16] proposed a Branch and Bound algorithm to solve the permutation flowshop scheduling problem with minimal time lags to minimize the total tardiness. Then, [14] developed an upper and lower bounds for the problem with minimal time lags. Also, other researches used the Mixed Integer Linear Programming to solve the permutation flowshop scheduling problem with time lags. [18] used a mixed integer formulation for the flowshop with two batches processing machines and release date to minimize the tardiness. [19] minimized the total tardiness for two machines by using the mixed integer programming. For [20], the goal is to minimize the weighted tardiness and earliness criteria for the flowshop sequence dependent groups scheduling problem by using the mixed integer linear programming. [21] solved the makespan and maximum tardiness criteria for work force scheduling with position dependent processing times based on mixed integer linear programming. Then, [13] proposed a formulation for the permutation scheduling problem with time lags for the total tardiness criterion. The same authors [15] proposed a new mixed integer linear programming models to solve the problem with time lags that emphasize an effective strategy for determining a lower bound for the total earliness and tardiness criteria. [17] proposed a mixed integer linear programming model for a two machine permutation flowshop scheduling problem with minimal time lags, and the models to solve optimally on the instances up to 40 jobs.
The dynamic programming method is based on the Bellman's principle, [3]. The aim of this method is to find the optimal sequence of partial decisions. Considering the work of [10], they developed a Branch and Bound algorithm to solve the permutation flowshop problem with m machines with minimal and maximal time lag constraints. The objective of the study is to minimize a non-classical criterion based on the weighted sum of machine completion times.

Moreover, [7] provided an efficient and effective model. to minimize the number of tardy jobs and to minimize the makespan, the authors proposed mixed integer mathematical programming formulation and other versions of simulated annealing algorithm to solve the permutation flowshop scheduling problem with minimal and maximal time lags. Then, [8] proposed two mathematical programming formulations and a simulated annealing algorithm is developed to solve the permutation flowshop scheduling problem with sequence dependent setup times and time lags constraints to minimise the number of tardy jobs.

2. BRANCH AND BOUND ALGORITHM

Among the exact methods for solving combinatorial optimization problems and in particular scheduling problems, the process of separation and progressive evaluation is the most used.
The corresponding general algorithm is given by Zribi, [26]:

- Divide the search space into sub-spaces (branches).
- Find an upper bound (lower) of an objective function on each sub-space research.
- Eliminate "The bad" subspace (according to the criteria to be optimized).
- Reproducing the preceding steps until getting the global optimum.

Considering the following example of the algorithm:
Step 1: (Initialization)
-Calculate an upper bound UB.
Step 2: (Isolation)
-Select the vertex (node) to separate and create his son.
Step 3: (evaluation)
For all vertices S (already created) do:
-If S is a complete solution then
-Calculate its value for the optimization criterion (total weighted tardiness in the case) and update UB.
-Otherwise calculate the lower bound LB.
-End If.
-End For.
Step 4: (elimination)
-Eliminate all vertices as LB> UB.
Step 5: (Stopping criterion)
-If all vertices have been eliminated, then stop.
-Otherwise go to **step 2**.

3. DESCRIPTION OF THE STUDIED BOUND

Let consider the following notations:
$C_{k,i}$ The completion time of the job sequenced on position i on the machine k
$\sigma_s = \sigma_s^1, \sigma_s^2, \ldots, \sigma_s^s$ The partial schedule until position s.
$\overline{\sigma_s}$ The set of unscheduled job until position s.
C_k, The completion time of the first job on machine k.
$C_{k,i}$ The completion time of the job in position i in the partial schedule

$\hat{C_{k,s}}$ The earliest starting time of the job sequenced on position 1 on the machine k.
$\hat{C_{k,i}}$ The earliest completion time of the job sequenced on position i on the machine k for $s+1 \leq i \leq n$.
$\theta_{h,j}^{min}$ The minimum time lags between machine h-1 and h for the job j.
$\theta_{0,j}^{min} = 0$ is a dummy variable fixed at 0.
d_j The due date of job j

Let:

$$r_{k',j}^{k} = \begin{cases} \sum_{h=k'}^{k}(p_{h,j} + \theta_{h,j}^{min}) & \text{if} 1 \leq k' \leq k \leq m \forall j \in \overline{\sigma_s} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

A graphical presentation of r is presented in Figure 1.
$r_{k',j}^{k}$: The release date is equal to the summation of the minimum time lags between machine h-1 and h for the job j and the processing time of the job j in the case that h including between k and k' and let $k = k'$ in the case where.

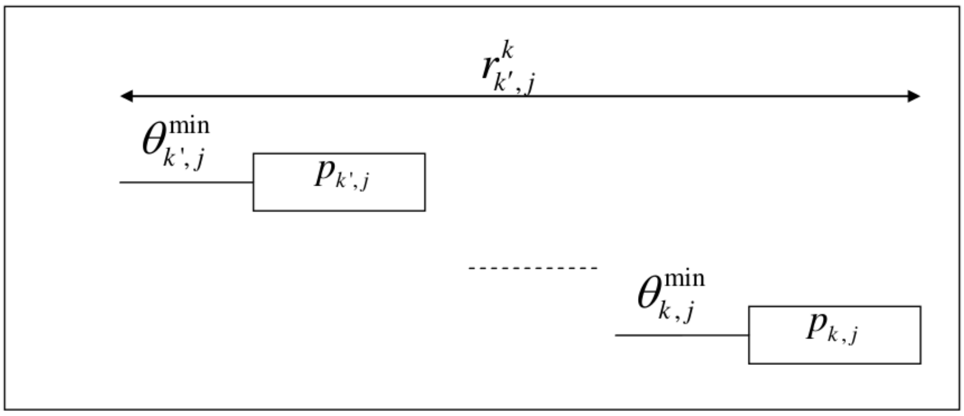

Figure 1: r calculation

$$r_{k,j}^{\prime k-1} = \begin{cases} \sum_{h=k'}^{k-1}(p_{h,j} + \theta_{h+1,j}^{min}) & \text{if} 1 \leq k' \leq k \leq m \forall j \in \overline{\sigma_s} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

A graphical presentation of r' is presented in Figure 2. In this case, let take the release date between two different machines k' and k. This release date can be equal to zero.

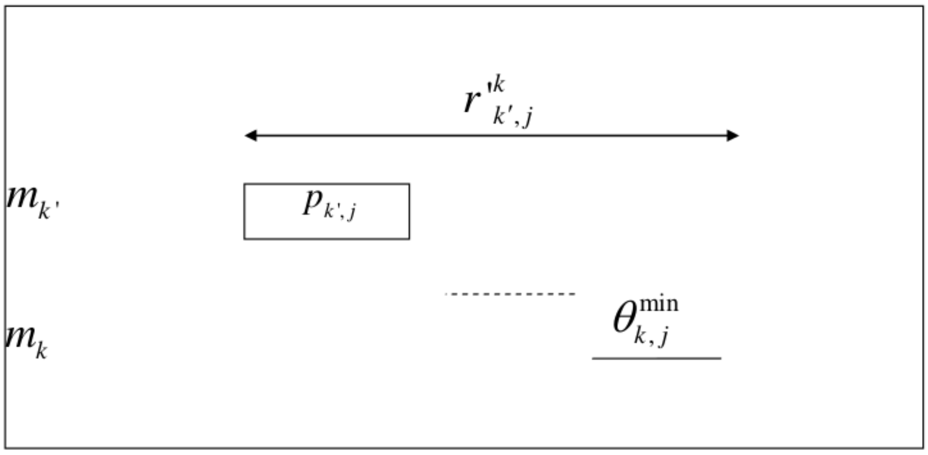

Figure 2: r' calculation

It is found here that we have limited the range of $r'^{k}_{k'}$ on k-1 like criterion of stoping.

$$r''^{k}_{k',j} = \begin{cases} \sum_{h=k'}^{k-1}(p_{k,j} + \theta^{min}_{h+1,j}) & \text{if} 1 \leq k' \leq k \leq m \forall j \in \overline{\sigma_s} \\ p_{k,j} \text{if} k' = k & \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

A graphical presentation of r'' is presented in Figure 3.
$r''^{k}_{k',j}$: Let add here the release date $r'_{k',j}{}^{k}$ to the processing time of the machine k on job j if $k' \leq k$ and in the case of one machine ($k' = k$), the release date will be exactly the processing time of the machine k on job j.

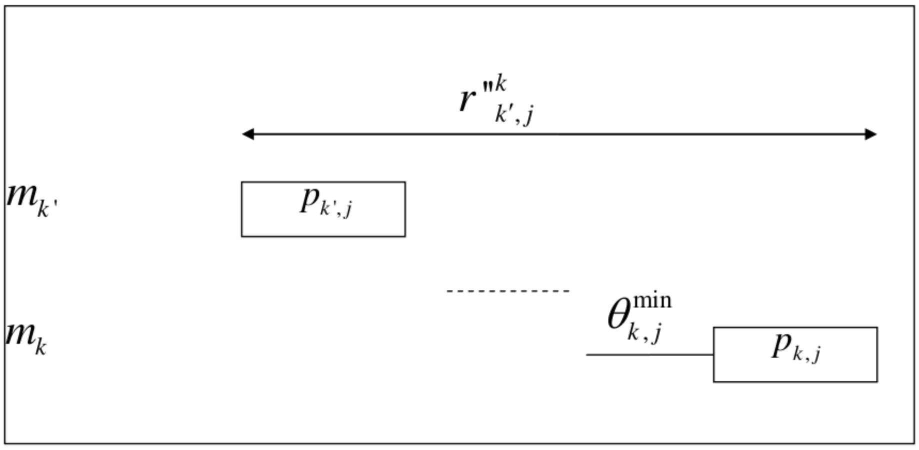

Figure 3: r'' calculation

Let $r^k_{k',[1]}, r^k_{k',[2]}, \ldots, r^k_{k',[n-s]}, r'^k_{k',[1]}, r'^k_{k',[2]}, \ldots, r'^k_{k',[n-s]}$ and $r''^k_{k',[1]}, r''^k_{k',[2]}, \ldots, r''^k_{k',[n-s]}$ are permutations,
that satisfies $r^k_{k',[1]} \leq r^k_{k',[2]} \leq \ldots \leq r^k_{k',[n-s]}$, $r'^k_{k',[1]} \leq r'^k_{k',[2]} \leq \ldots \leq r'^k_{k',[n-s]}$ and $r''^k_{k',[1]} \leq r''^k_{k',[2]} \leq \ldots \leq r''^k_{k',[n-s]}$ respectively $\forall 1 \leq k' \leq k \leq m$.

In the same way, $p_{k,[1]}, p_{k,[2]}, \ldots p_{k,[n]}$ is a permutation of processing times on the machine $k (1 \leq k \leq m)$ where $p_{k,[1]} \leq p_{k,[2]} \leq \ldots \leq p_{k,[n]}$.

Let:

$$\hat{C_{k,s}} \geq C_{k',s} + r'^k_{k',[1]}, 1 \leq k' \leq k \leq m \tag{9}$$

The earliest starting time of the job sequenced on position s on the machine k will be more or equal to the summation of the completion time of the job on machine k' in position i and the release date between the two machine k and k'(we can fall in case where $k = k'$).

Then:

$$\hat{C_{k,s}} = \max_{1 \leq k' \leq k} C_{k',s} + r'^k_{k',[1]}, 1 \leq k \leq m \tag{10}$$

This earliest starting time is a lower bound of the starting time on machine k.

Let also:

$$\hat{C_{k,s}} \geq \hat{C_{k,s}} + \sum_{j=1}^{i-s} p_{k,[j]}, 1 \leq k \leq m, s+1 \leq i \leq n \tag{11}$$

Let us introduce now the earliest starting time of the job sequenced on position i on machine k' and the release date.

$$\hat{C_{k,s}} \geq \hat{C_{k',s}} + r^k_{k'+1,[1]}, 2 \leq k' \leq k \leq m, s+1 \leq i \leq n \tag{12}$$

Let integrate the new release date $r''^{k}_{k',j}$ (already defines) in the new equation and we obtained:

$$\hat{C_{k,s}} \geq \hat{C_{k',s}} + r''^{k}_{k',[1]}, 2 \leq k' \leq k \leq m, s+1 \leq i \leq n \tag{13}$$

Then:

$$\hat{C_{1,i}} \geq \hat{C_{1,s}} + \sum_{j=1}^{i-s} p_{1,[j]}, s+1 \leq i \leq n \tag{14}$$

$$\hat{C_{k,i}} = \max_{2 \leq k' \leq k} \hat{C_{k,s}} + \sum_{j=1}^{i-s} p_{k,[j]}, \hat{C_{k',i}} + r^{k}_{k'+1,[1]}, \hat{C_{k',i-1}} + r''^{k}_{k',[1]} 2 \leq k \leq m \, and \, s{+}1 \leq i \leq n \tag{15}$$

is a lower bound of completion time of the job in position i on machine k.
And

$$LB = \sum_{j \in \sigma} C_{m,j} + \max_{k=1,\ldots,m} \sum_{i=s+1}^{n} \hat{C_{k,i}} + \sum_{j \in \overline{\sigma}} r^{m}_{k+1,j} \tag{16}$$

is a lower bound of total completion time.
Let:

$T_{i,j}$: Total tardiness that equal to the difference between the earliest completion time of the job sequenced on position i on the machine m and the due date of job j.

A lower bound of weighted tardiness can be obtained by solving the following assignment problem:

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} w_j T_{i,j} x_{ij} \tag{17}$$

subject to:

$$\sum_{j=1}^{n} x_{i,j} i = 1, 2, \ldots, n \tag{18}$$

$$\sum_{i=1}^{n} x_{i,j} i = 1, 2, \ldots, n \tag{19}$$

$$x_{i,j} \in (0, 1) \tag{20}$$

$$x_{i,j} = \begin{cases} 1 & \text{if the job } j \text{ is assigned to position } i \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

4. COMPUTATIONAL EXPERIMENTS

The proposed Branch and Bound algorithm was coded in C + +. That is why all these experiments were performed on a Windows XP computer and also on a personal computer with Intel Pentium IV, 3.2 GHz and 512 MB of memory. This study aims to evaluate the performance of this algorithm for optimization, 560 problems instances for this work were generated.

To appreciate the importance of the Branch and Bound algorithm, we take the research experiments on a number of issues generated arbitrarily and divided into 29 classes according to the value of $(m,\, n)$. For each class, 20 instances were generated.

For the case of this study, there are four scenarios of due dates, each of them has been generated in the range [1, 99] with [P (1-T-R/2), P (1-T+R/2)], where T and R are the tardiness factor and P is the value of lower bound on the makespan for the permutation flowshop scheduling problem [23], [24] .

So let consider:

$$P = \max_{1\leq k\leq m} \sum_{i=1}^{n} P_{k,i} + min \sum_{q=1}^{m} P_{q,i} + min \sum_{q=k+1}^{m} P_{q,i}, \max_{i} \sum_{k=1}^{m} P_{k,i} \tag{22}$$

The four scenarios of due dates are presented following the value of T and R:
Scenario 1: T=0.2 and R=0.6
Scenario 2: T=0.2 and R=1.2
Scenario 3: T=0.4 and R=0.6
Scenario 4: T=0.4 and R=1.2

It is also important to note that the maximal and minimal time lags are randomly drawn in [0,300] and [0,200] respectively [9].
Table 1, Table 2, Table 3 and Table 4 show the experimental results which coincides with Scenario 1, Scenario 2, Scenario 3, and Scenario 4, respectively, to minimize the total weighted tardiness.

Based on the first scenario (table 1), we note that up to 14*2, of these instances are solved optimally and without interruption. Note that the instances 10*2, 10*3, 10*4, 10*5 and 10*7 are solved in less than one second. Also, some instances have not been solved in some classes such as (14*3, 16*2, and especially 20*5). We say that for the same number of jobs in progress, when there is an increase in number of machines, the average number of nodes Increases quickly.

Table 1: Computational results according to scenario 1 for minimizing the total weighted tardiness.

Instances	NN	$Nnmax$	T	$Tmax$	Up
10*2	11528,35	27248,5	0,12	0,37	0
10*3	23645,25	41439,95	0,21	0,51	0
10*4	19219,25	25909,45	0,22	0,42	0
10*5	10198,85	16223,25	0,18	0,36	0
10*7	23656,25	35045,1	0,33	0,59	0
10*10	32418,3	50684,7	0,59	1,05	0
12*2	312830,4	1040016,4	5,14	22,07	0
12*3	396005,6	694489,25	8,30	17,38	0
12*4	580231,95	1717267,4	9,01	37,83	0
12*5	266249,6	410014	6,72	12	0
12*7	1730685,75	2288974,95	30,76	45,84	0
12*10	1368502,05	1789957,5	31,85	48,79	0
14*2	3110547,25	14735840,2	68,95	435,58	0
14*3	7737527	17283171,2	355,03	931,91	2
14*4	11246053,5	22082604,1	299,96	721,67	0
14*5	24600808,6	55778842	640,14	1135,77	0
14*7	22317283	82767341	702,19	1210,92	2
14*10	19226259,5	32778099,2	837,24	1815,17	0
16*2	15622155,8	66014936,4	871,63	2999,83	6
16*3	17702616,8	46821665	1411,86	3382,79	8
16*4	21842248,9	27744732,8	1857,12	4162,38	10
18*2	7176075,09	10045034,6	1877,19	2952,65	6
18*3	8432190,89	19084321	1920,62	3001,82	8
18*4	89102837,7	90273645	2100,02	3200,72	10
18*5	99910283,5	100293740	2709,91	2909,92	12
20*2	20334305	28325395,3	1411,37	2235,73	10
20*3	87125632	92615234	1501,19	2100,19	11
20*4	5521486,95	9054885,7	905,47	2340,12	14
20*5	31639598,1	122382556	1816,23	2884,24	17
Mean	**17152520**	**29179976,3**	**737,23**	**1331,33**	

From Table 2, we marked that there is a significant decrease in unsolved instances compared to the first scenario. This leads to a very effective improvement over the first one. This scenario is considered the most efficient compared to all other. For this Scenario, we have also some unsolved node (but not much), especially 20*3, 20*4 and 20*5.

Table 2: Computational results according to scenario 2 for minimizing the total weighted tardiness.

Instances	NN	Nnmax	T	Tmax	Up
10*2	12278,05	29345,6	0,13	0,39	0
10*3	25212,15	43003,42	0,22	0,52	0
10*4	200189,15	27309,55	0,23	0,44	0
10*5	11293,05	17123,15	0,19	0,37	0
10*7	24512,35	36144,1	0,34	0,60	0
10*10	34219,5	52619,7	0,57	1,05	0
12*2	322910,4	1211512,4	4,46	18,96	0
12*3	415211,6	715213,15	6,97	14,47	0
12*4	597145,05	1772510,24	7,51	31,55	0
12*5	283101,16	432192	5,40	9,95	0
12*7	1752210,05	2301110,95	27,61	40,42	0
12*10	1382701,25	1810101,25	28,72	42,40	0
14*2	3142016,25	14772510,5	65,15	423,72	0
14*3	7742601	17301201,3	366,52	1000,70	2
14*4	11293153,5	22120002,1	314,86	759,93	0
14*5	2469234,2	55792135	471,14	1158,79	2
14*7	22345101	82915090	712,12	1232,96	2
14*10	19242279,3	32792123,3	822,79	1770	3
16*2	15645210,2	66052102,1	854,01	294,17	7
16*3	17737910,3	46872901	1382,13	3293,10	9
16*4	21873168,2	27762012,5	1840,12	4104,68	11
18*2	7199101,29	10102291,2	1867,00	2960,62	7
18*3	8475002,39	19094112	1962,63	3010,86	8
18*4	89129910,2	90299123	1995,02	3210,12	9
18*5	99947310,3	101342432	2729,96	2919,54	12
20*2	20352765	28373821,3	1399,08	2217,30	8
20*3	87155987	92643432	1471,17	2010,98	10
20*4	5581382,15	9102662,17	884,54	2304,06	12
20*5	31682652,3	143901237	1793,4	2841,75	17
Mean	**16416405,8**	**29989219,9**	**724,62**	**1230,15**	

Table 3 shows that, in terms of average of resolved nodes, the results in scenario 3 can be classified between the two first scenarios. Moreover, the large number of unresolved nodes belongs to 20*2 and 20*3 classes. In short, this improvement shows the importance of the values of two variables T and R.

Table 3: Computational results according to scenario 3 for minimizing the total weighted tardiness.

Instances	NN	Nnmax	T	Tmax	Up
10*2	13562,86	31009,2	0,2	0,6	0
10*3	28692,02	45215,62	0,34	0,79	0
10*4	210941,87	30156,43	0,36	0,66	0
10*5	11487,12	19254,29	0,29	0,51	0
10*7	28136,64	39122,4	0,05	0,11	0
10*10	37479,3	55139,2	0,73	1,3	0
12*2	32963,8	1305859,3	6,09	23,36	0
12*3	427381,2	731299,49	9,18	18,34	0
12*4	619256,93	1854632,51	10,22	39,55	0
12*5	292204,52	441150	7,29	13,24	0
12*7	1836219,28	2399548,75	31,69	537,62	0
12*10	1429556,69	1915648,53	33,19	49,81	0
14*2	3229751,39	15234167,1	70,43	437,72	0
14*3	7891243	18125884,4	357,07	934,68	2
14*4	11654302,4	22348762,3	309,32	732,27	0
14*5	2473594,58	5632419	461,8	1138,72	0
14*7	22418769	83006714	712,18	1214,93	2
14*10	19338514,8	32814652,3	511,42	1803,33	0
16*2	15725360,4	66112873,8	873,64	3015,05	5
16*3	17842168,3	46954321	1411,14	3397,77	7
16*4	21991018,5	27882941,2	1860,25	4174,08	9
18*2	7225614,37	10187331,5	1867,18	2942,76	8
18*3	8586047,11	19125876	1928,53	3011,3	10
18*4	891982535	90334521	2120,63	3210,75	12
18*5	99947310,3	101342432	2716,99	2929,93	13
20*2	20416729	28421530,8	1413,89	2244,85	8
20*3	87283951	92720562	1510,34	2121,92	9
20*4	5675412,39	9178699,31	907,54	2350,7	10
20*5	31745832,3	143987022	1819,74	2894,16	14
Mean	**44151587,4**	**28354439,5**	**722,48**	**1353,13**	

The last scenario (Table 4) is considered the worst among these four scenarios in terms of average of resolved nodes. We have 134 instances that are unresolved. The most important part of unresolved problem is concentrated in the instances 18*3, 18*4, 18*5, 20*3, 20*4, 20*5.

Table 4: Computational results according to scenario 4 for minimizing the total weighted tardiness.

Instances	NN	Nnmax	T	Tmax	Up
10*2	16219,21	33654,9	0,20	0,65	0
10*3	31084,67	47962,94	0,36	0,85	0
10*4	223699,81	33251,25	0,39	0,69	0
10*5	13670,68	21056,42	0,31	0,62	0
10*7	30159,38	425871	0,07	0,22	0
10*10	39952,1	57926,7	0,75	1,91	0
12*2	34753,9	1415328,6	7,19	24,33	0
12*3	433921,3	744123,55	11,18	20,33	0
12*4	627361,05	1975126,31	13,23	43,58	0
12*5	305986,12	462572	9,32	17,23	0
12*7	1836219,28	2399548,75	39,77	547,66	0
12*10	1516816,23	1999248,62	36,12	52,21	0
14*2	3301541,57	15362189,4	72,43	445,03	0
14*3	832179	18189357,2	367,72	944,32	3
14*4	11796252,1	22439246,3	319,15	743,65	0
14*5	2503958,63	5688912	473,80	1149,95	0
14*7	22524783	83623148	723,88	1232,93	3
14*10	19467231,4	33521689,1	833,32	1824,39	0
16*2	1639217,91	66325004,1	879,69	3030,18	7
16*3	17922546,1	45001236	1421,16	3409,73	9
16*4	22015368,2	27902154,6	1872,20	4185,14	10
18*2	7275411,23	10212367,6	1873,18	2963,12	9
18*3	8625761,53	19295326	1947,10	3032,34	12
18*4	89203498,9	9085642	2142,65	3230,71	13
18*5	10025697,3	104772432	2728,55	2938,97	15
20*2	20552713	28507961,6	1432,97	2256,54	9
20*3	87310255	92779213	1535,51	2220,92	10
20*4	5711326,39	9231457,1	928,55	2359,78	16
20*5	31825694,1	14436587	1844,72	2920,77	18
Mean	**12677354,5**	**21241020,5**	**741,92**	**1365,48**	

5. CONCLUSIONS

In this paper, we used several specific characteristics of the permutation flowshop scheduling problem with time lags to obtain the lower bounds of the total weighted tardiness criterion. Then, a Branch and Bound algorithm was developed. The computational experiments show that the proposed algorithms can be a test for these problems with multiple jobs related to the total weighted tardiness objective and we can find the objective. However, for some problems, the results of the total weighted tardiness version are just motivating. Generally, we can use this algorithm of Branch and Bound for the problems of large sizes. For future work, to improve the algorithm, the development of efficient heuristics are suggested to solve a large number of nodes.

REFERENCES

[1] Azizi, V., and Hu, G., "A branch and bound algorithm to solve a two-machine no-wait flowshop scheduling problem with truncated learning function", *International Journal of Management Science and Engineering Management*, 15 (2) (2020) 89–95.

[2] Basnet, C., Foulds, L. R., Wilson, J., "Scheduling contractors' farm-to-farm crop harvesting operations", *International Transactions in Operational Research*, 13 (1) (2006) 1–15.

[3] Bellman, R., "Dynamic Programming" *Princeton University Press, Princeton, NJ*, 19, 1957.

[4] Chu, C., and Proth, J. M., "Single machine scheduling with chain structured precedence constraints and separation time windows", *IEEE Trans Robot Autom*, 12 (6) (1996) 835–844.

[5] Companys, R., and Mateo, M., " Different behaviour of a double branch-and-bound algorithm $F_m|perm|C_{max}$ and $F_m|block|C_{max}$ problems", *Computers and Operations Research*, 34 (4) (2007) 938–953.

[6] Dell'Amico, M., "Shop problems with two machines and time lags)", *Operations Research*, 44 (5) (1996) 777–787.

[7] Dhouib, E., Teghem, J., and Loukil, T., "Lexicographic optimization of a permutation flowshop with time lags constraint", *International Transactions in Operational Research*, 20 (2) (2013) 213–232.

[8] Dhouib, E., Teghem, J., and Loukil, T., "Minimizing the number of tardy jobs in a permutation flowshop scheduling problem with setup times and time lags constraints", *Journal of Mathematical Modeling and Algorithms*, 12 (1) (2013) 85–99.

[9] Fondrevelle, J., Oulamara, A., and Portmann, M. C., "Permutation flowshop scheduling problems with maximal and minimal time lags", *Comput Oper Res*, 33 (6) (2006) 1540–1556.

[10] Fondrevelle, J., Oulamara, A., and Portmann, M. C., " Permutation flowshop scheduling problems with time lags to minimize the weighted sum of machine completion times", *International Journal of Production Economics*, 112 (1) (2008) 168–176.

[11] Foulds, L.R., and Wilson, J. M., "Scheduling operations for the harvesting of renewable resources", *J Food Eng*, 70 (3) (2005) 281–292.

[12] Gargouri, E., "Ordonnancement coopératif en industries agroalimentaires", *Thèse de Doctorat, Université des Sciences et technologies de Lille 1.*

[13] Hamdi, I., and Loukil, T., "Minimizing total tardiness in the permutation flowshop scheduling problem with minimal and maximal time lags", *Operational Research*, 15 (1) (2015) 95–114.

[14] Hamdi, I., and Loukil, T., "Upper and lower bounds for the permutation flowshop scheduling problem with minimal time lags", *Optimization Letters*, 9 (3) (2015b) 465–482.

[15] Hamdi, I., and Loukil, T., "The permutation flowshop scheduling problem with exact time lags to minimize the total earliness and tardiness", *Internatioal Journal of Opererational Research*, 28 (1) (2017) 70–86.

[16] Hamdi, I., Oulamara, A., and Loukil, T., "A branch and bound algorithm to minimize total tardiness in the two machine flowshop problem with minimal time lags", Enterprise Computing Series, McGraw-Hill, 2000. *Internatioal Journal of Opererational Research*, 23 (4) (2015) 387–405.

[17] Hamdi, I., and Toumi, S., "MILP models and valid inequalities for the two-machine permutation flowshop scheduling problem with minimal time lags", *Journal of Industrial Engineering International*, 15 (2019) 223–229.

[18] Huang, J. D., Liu1, J. J., Chen, Q. X., and Mao, N., " Mixed integer fomulation to tardiness objectives in a flowshop with two batches processing machines and release date", *in: CIE43 proceedings, 16–18 October, The University of Hong Kong*, 2013.

[19] Kharbeche, M., and Haouari, M., "MIP models for minimizing total tardiness in a two-machine flowshop", *Journal of the Operational Research Society*, 64 (5) (2013) 690–707.

[20] Keshavarz, T., Salmasi, N., and Varmazyar, M., "Flowshop sequence dependent group scheduling with minimisation of weighted earliness and tardiness", *European Journal of Industrial Engineering*, 13 (1) (2019) 54–80.

[21] Moreno-Camacho,, CA., Montoya-Torres, J. R., and Vélez-Gallego, M. C., "A comparison of mixed-integer linear programming models for workforce scheduling with position-dependent processing times", *Engineering Optimization*, 50 (6) (2018) 917–932.

[22] Moslehi, G., Mahnam, M., Nayeri, M. A., and Azaron, A., " A branch and bound algorithm to minimise the sum of maximum earliness and tardiness in the single machine", *International Journal of Operational Research*, 8 (4) (2010) 458–482.

[23] Ronconi, D. P., "A branch and bound algorithm to minimize the makespan in a flowshop with blocking", *Annals of Operations Research*, 138 (1) (2005) 53–65.

[24] Toumi, S., Jarboui, B., Eddaly, M., and Rebai, A., "Branch and Bound algorithm for solving blocking flowshop scheduling problem with total tardiness and total weighted tardiness criteria", *International Journal of Operational Research*, 30 (4) (2017) 441–459.

[25] Toumi, S., Jarboui, B., Eddaly, M., and Rebai, A., " Branch and Bound algorithm for solving blocking flowshop scheduling problems with makespan criterion", *International Journal of Mathematics in Operational Research*, 10 (1) (2017) 34–48.

[26] Zribi, N., "Ordonnancement des job-shops flexibles sous contraintes de disponibilité des machines", *Thèse de Doctorat, Université des Sciences et Technologies de Lille*, 2005.