

**A branch and bound method in a permutation flow shop with blocking and setup times****Marcelo Seido Nagano<sup>a\*</sup>, Mauricio Iwama Takano<sup>b</sup> and João Vítor Silva Robazzi<sup>c</sup>**<sup>a</sup>*University of São Paulo, São Carlos School of Engineering, Av. Trabalhador São-Carlense, 400, 13566-590, São Carlos, São Paulo, Brazil*<sup>b</sup>*Federal University of Technology – Paraná, Av. Alberto Carazzai, 1640, 86300-000, Cornélio Procópio, Paraná, Brazil*<sup>c</sup>*Federal Institute of Education, Science and Technology of São Paulo, R. Américo Ambrósio, 269, 14169-263, Sertãozinho, São Paulo, Brazil***CHRONICLE***Article history:*Received May 11 2021  
Received in Revised Format  
June 28 2021  
Accepted October 22 2021  
Available online  
October, 27 2021

## Keywords:

*Scheduling  
Permutation flow shop  
Blocking, Setup  
Total flow time  
Total tardiness  
Branch and bound***ABSTRACT**

In this paper it is presented an improvement of the branch and bound algorithm for the permutation flow shop problem with blocking-in-process and setup times with the objective of minimizing the total flow time and tardiness, which is known to be NP-Hard when there are two or more machines involved. With that objective in mind, a new machine-based lower bound that exploits some structural properties of the problem. A database with 27 classes of problems, varying in number of jobs ( $n$ ) and number of machines ( $m$ ) was used to perform the computational experiments. Results show that the algorithm can deal with most of the problems with less than 20 jobs in less than one hour. Thus, the method proposed in this work can solve the scheduling of many applications in manufacturing environments with limited buffers and separated setup times.

© 2022 by the authors; licensee Growing Science, Canada

**1. Introduction**

In a flow shop scheduling problem, a number of jobs must be processed on each of the  $m$  machines. Every job must go on through the same machines in the same order (i.e., all jobs must be processed first on machine 1, then on machine 2, and so on). The permutation constraint means that all jobs must be processed on every machine in the same exact sequence, all machines can process no more than one job at a time, and interruptions are not accepted. The blocking constraint with zero buffer represents the lack of intermediate queues between machines, which blocks a machine in case a job finishes its processing on it and the next machine is still processing another job. Lastly, the setup constraint means that the setup times are separate from the processing times on the machines.

The total tardiness represents the time exceeded at the end of a task in relation to its due date. When not met, this criterion can result in contractual fines, increasing the total cost, in addition to the loss of reliability, a factor that results in loss of customers and damage to the company's reputation.

According to Garey et al. (1976) the flow shop problem with two or more machines and with the objective to minimize the total flow time is NP-hard.

Many papers consider the setup time either attached to processing time or nonexistent (McCormick et al., 1989; Pan & Wang, 2012; Ronconi, 2004). Considering the setup time detached from the processing time grants more flexibility for the scheduling, allowing, for example, a machine to be prepared right after it finishes the processing of the previous job, even before the

\* Corresponding author Tel.: 55 16 3373-9428; Fax: 55 16 3373-9425  
E-mail: [dmagano@usp.br](mailto:dmagano@usp.br); [dmagano@sc.usp.br](mailto:dmagano@sc.usp.br) (M. S. Nagano)

previous machine has finished processing the job. This ensures a better use of time, consequently a possible reduction in the total flow time and, perhaps, in the tardiness. Moreover, there is a different setup time for each pair of jobs in each machine.

There are papers that considered the problem with unrestrained buffers (Nawaz et al., 1983), which may not represent the reality of an Industry. In an environment with limited buffers, there is a possibility of a machine blockage. In this paper, a zero buffer constraint is considered, i.e., there is no buffer in between machines. The lack of buffers can induce blocking to occur, which is when a job that has already been processed on a specific machine blocks it until the next machine is ready to receive it. Any flow shop problem with limited buffers can be modeled as a flow shop problem with zero buffer, because a machine with zero setup and processing times for all jobs can represent a unit capacity buffer (McCormick et al., 1989). Figure 1 shows a Gantt chart of the problem.

The chart in Fig. 1 starts immediately after the job  $i$  finished being processed on both machines 1 and 2. So, the chart begins with both machines being setup to start processing job  $j$  ( $S_{ij1}$  and  $S_{ij2}$ ). It can be noted that the setup on machine 2 can start even before machine 1 has finished its setup. After the setup on machine 2 is finished, it must wait for machine 1 to finish processing job  $j$  before it can start processing it, so it stays idle for a while. Each machine can start setting up for the next job ( $S_{j,j+1,1}$  and  $S_{j,j+1,2}$ ) immediately after it finishes processing job  $j$ . In this case, observe that machine 1 has finished processing job  $j+1$  prior to the end of the setup process on machine 2. Since there is no buffer between the machines, a *block* occurs. That is, the job must stay on Machine 1 waiting, impeding it from initiating the setup for job  $j+2$ .

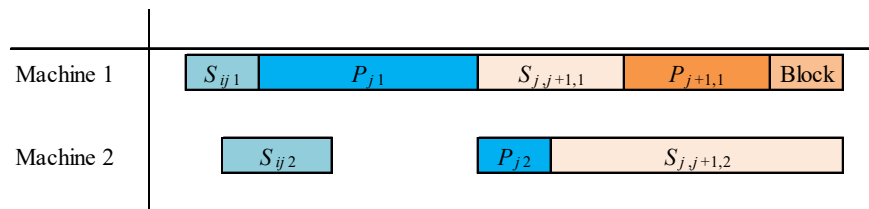


Fig. 1. Example of a blocking flow shop scheduling problem with setup times

Miyata and Nagano (2019) presented a review of 139 papers that address the  $m$ -machine flow shop scheduling problem with blocking constraints. Results show that most of the papers propose metaheuristics or constructive heuristics methods to solve the problem (77% of the papers in total), and most of the papers focus on minimizing makespan (62% of the papers reviewed).

Heuristics methods are usually used to solve large-sized blocking flow shop problems due to their complexity. One of the first authors to explore the blocking environment were Reddi & Ramamoorthy (1972), which consisted of reducing a flow shop problem with two machines, zero buffer, and the objective function of minimizing the makespan to a special case of the Traveling Salesman Problem, which can be solved in polynomial time using the Gilmore-Gomory algorithm (Gilmore & Gomory, 1964). McCormick et al. (1989) developed a constructive heuristic named Profile Fitting (PF), which attempts to schedule the tasks to minimize the sum of idle times, as well as machine blocking times. Ignall and Schrage (1965) developed a branch and bound algorithm to minimize the mean completion time in a two-machine case and the makespan on a flow shop environment with three machines. Bansal (1977) expanded this machine based lower bound to an environment with  $m$  machines with the objective to minimize the total flow time considering all release dates equal to zero. Ahmadi & Bagchi (1990) proposed a machine based lower bound with the preemptive relaxation, which, in terms of value, outperformed the formulation of Bansal (1977). Chung et al. (2002) considered the total flow time problem in a flow shop environment, then introduced a new lower bound that surpassed both previous methods. Results show that the algorithm can handle problems with 15 jobs or less and can solve most problems with up to 20 jobs. Chung et al. (2006) proposed an adaptation of the lower bound proposed by Chung et al. (2002) to minimize the total tardiness. Results show that the algorithm can solve problems with  $n \leq 20$ . Only a few papers explore the use of a branch and bound algorithm to solve the flow shop problem with blocking. Ronconi and Armentano (2001) developed a lower bound based on departure time that was used to minimize both the total tardiness and the makespan in flow shops with zero buffer problems. Ronconi (2005) outran the lower bound for the blocking flow shop problem with the objective of minimizing the makespan proposed by Ronconi & Armentano (2001) in most of the cases. Their lower bound determined an underestimated departure time of the last job in a non-partial sequence. Moslehi & Khorasani (2013) developed a lower bound that minimizes the total flow time in a flow shop problem with zero buffer constraint. The proposed algorithm is capable of solving problems with  $n \leq 18$  and  $m \leq 10$  of 17 instances of the Taillard's (Taillard, 1993) benchmark in less than 20 minutes. Sanches et al. (2016) proposed the use of different initial upper bounds (UB) to reduce the computational time of the branch and bound algorithm in flow shop problems with blocking and the objective of minimizing the makespan. Many heuristic methods were evaluated to provide an initial UB, and a significant reduction of the computational time was noticed in some of them. Nagano et al. (2020) proposed a machine-based lower bound for the problem. The lower bound exploits the machine idleness and the occurrence of blocking and proved to be able to optimally solve small and medium problems. The proposed lower bound was applied to a branch and bound algorithm and compared to the lower bound proposed by Moslehi & Khorasani (2013), results show that the former outperforms the latter.

The flow shop problem with both blocking and setup time constraints was very little explored in the literature. Rios-Mercado & Bard (1999) were one of the pioneers to address this problem. In their work, the lower bound proposed by Ignall & Schrage (1965) was adapted to the problem. A lower bound for the makespan in a flow shop problem with blocking and sequence dependent setup times was created by Takano & Nagano (2017). The proposed lower bound addresses two structural properties of the problem: an upper bound for the machine idle and a lower bound for the machine blocking time. Takano & Nagano (2019) evaluated 28 different constructive heuristics for the permutation flow shop problem with blocking and setup and the objective function to minimize the makespan. Takano & Nagano (2020) also addressed the problem with the minimization of the makespan objective. They proposed two mixed-integer programming (MILP) models and adapted the two models proposed by Ronconi & Birgin (2012) for the problem. Also, an Iterated Greedy (IG) algorithm proposed by Pan & Ruiz (2014) was adapted for the problem and compared to the MILP models. Robazzi et al. (2021) presented an improvement of the branch-and-bound algorithm for the blocking-in-process and setup times permutation flow shop problem with total flow time criterion. Tests show that the algorithm can handle most of the  $n < 20$  problems in less than one hour.

In this paper two different lower bounds for the branch and bound algorithm for permutation flow shop problems with blocking and sequence dependent setup time constraints are presented. The proposed lower bounds aim to optimally solve small and medium size problems. One of which deals with the total flow time criterion and the second one deals with the total tardiness criterion.

The problem is defined, and its equations are detailed in Section 2. The branch and bound algorithm method is described, and the development of the lower bounds are demonstrated and explained in Section 3. The results of the computational tests are displayed in Section 4 and, finally, in section 5 the conclusions are presented.

## 2. Problem Definition

Regardless of the evaluation criterion, a set of equations can be used to define the flow shop problem with blocking and sequence dependent setup time constraints. Thereunto, the following variables are used:

$\sigma$ : arbitrary sequence of jobs;

$k$ : available machine;

$i$ : job that directly precedes job  $j$  in the sequence;

$p_{jk}$ : processing time of the  $j$ -th job in the sequence on machine  $k$ ;

$S_{ijk}$ : setup time of machine  $k$  between the  $i$ -th and the  $j$ -th job in the sequence;

$S_{01k}$ : setup time of machine  $k$  before processing the first job in the sequence;

$R_{jk}$ : completion time of the setup of machine  $k$  to the  $j$ -th job in the sequence; and

$D_{jk}$ : departure time of the  $j$ -th job in the sequence on machine  $k$ .

Equations 1 to 5 show how the departure times are calculated for a given sequence:

$$R_{[1]k} = S_{[0][1]k} \quad \forall 1 \leq k \leq m \quad (1)$$

$$D_{j1} = \max(R_{j2}, R_{j1} + p_{j1}) \quad \forall 1 \leq j \leq n \quad (2)$$

$$D_{jk} = \max(R_{j,k+1}, D_{j,k-1} + p_{jk}) \quad \forall 1 \leq j \leq n; 2 \leq k \leq m - 1 \quad (3)$$

$$D_{jm} = D_{j,m-1} + p_{jm} \quad \forall 1 \leq j \leq n \quad (4)$$

$$R_{jk} = D_{ik} + S_{ijk} \quad \forall 2 \leq j \leq n; 1 \leq k \leq m$$

First, the completion times of the setup of the first job in the sequence on all machines ( $R_{[1]k}$ ) are calculated by Eq. (1). Then, the departure times of the first job on all machines ( $D_{[1]k}$ ) are calculated by Eqs. (2-4). Then, Eq. (5) is used to calculate  $R_{jk}$  for the subsequent job on all machines. Eqs. (2-4) are used again to calculate the departure times ( $D_{jk}$ ) of the next job on all machines. This is repeated until the departure time of the last job on the last machine is calculated.

The sum of all departure times of all jobs on the last machine ( $D_{jm}$ ) is the total flow time, and the sum of all tardiness of all jobs ( $D_{jm} - d_j$ ) is the total tardiness. Therefore, to calculate the total flow time ( $TFT$ ) and the total tardiness ( $TT$ ), Eq. (6) and Eq. (7) are used respectively.

$$TFT = \sum_{j=1}^n D_{jm} \quad (6)$$

$$TT = \sum_{j=1}^n \max(D_{jm} - d_j, 0) \quad (7)$$

where:  $d_j$  is the due date of job  $j$ .

### 3. The Branch and Bound Algorithm

The branch and bound algorithm is a method used to obtain the best feasible solution of a problem. It does so by replacing the original problem by a set of sub-problems, called nodes, which are usually much easier to solve. This replacement process can be repeated until the best feasible solution is obtained. For the scheduling problem, each node is defined by a Partial Sequence (PS). A Non-Partial Sequence (NPS) is the set of jobs that are not included in a PS, and there is one for each node.

Every time a node is branched, two or more new nodes are created by including a job from |NPS| to the |PS| that belongs to the node that is being branched. A lower bound of the optimization criterium is calculated for each new node, which represents the smallest value that can be obtained by that |PS|, regardless of the sequence of the jobs in |NPS|.

In this paper, the depth first rule to select the nodes that will be branched. This rule states that the node with the most jobs in |PS| is branched. the lowest lower bound breaks the ties. This paper aims to evaluate the quality of the proposed lower bounds; therefore, the initial solution is at first defined as a very high value, called big M, and is updated every time a smaller feasible solution is obtained.

#### 3.1. Proposed Lower Bounds

The lower bound used determines the effectiveness of a branch and bound algorithm. Minor computational complexities and tightness of these lower bounds define their efficiency. Thus, the proposed lower bounds focus on both these characteristics, and, for that, they use the following definitions:

- $p_{[j]k}$ : is the  $(j-s)$ -th smallest processing time among all jobs that have not yet been scheduled on machine  $k$ ;
- $S_{[1]k}$ : is the smallest setup time after job  $i$  among all jobs that have not yet been scheduled on machine  $k$ ;
- $S_{[1]jk}$ : is the smallest setup time before job  $j$  among all jobs that have not yet been scheduled on machine  $k$ ;
- $ED_{tk}$ : is an underestimation of the departure time of a job in position  $t > s$  on machine  $k$ . Where  $s$  is the last job in the partial sequence  $\sigma$ ;
- $d_{[j]}$ : is the  $(j-s)$ -th smallest due date among all jobs that have not yet been scheduled.

The lower bound adopts a relaxation in which one machine can process just one job at a time whilst the others can handle all  $n$  jobs at a time. Also, another relaxation assumed is that the setup time used is always the smallest one available. To do that, it is used the following notations:

$$Setup_k(j) = \{S_{[1],|PS+1|,k}, S_{[1],|PS+2|,k}, \dots, S_{[1],j,k}, \dots, S_{[1],n,k}\}$$

$\hat{Setup}_k(j) = Setup_k(j)$  arranged in a non-decreasing order.

$ED_{tk}$  is given by Eq. (8).

$$ED_{tk} = \max \left( \max \left( ST_{s+1,k} + \sum_{j=s+1}^{t-1} p_{[j]k}; ST_{s+1,k+1} + \sum_{j=s+1}^{t-2} p_{[j],k+1} + \hat{Setup}_{k+1}(t-1) \right) + \hat{Setup}_k(t); ST_{s+1,k-1} + \sum_{j=s+1}^t p_{[j],k-1} \right) + p_{[t]k} \quad (8)$$

where  $ST_{s+1,k}$  is the underestimate of the starting time of the processing of job  $(s+1)$  on machine  $k$ , as shown in Eq. (9).

$$ST_{s+1,k} = \max(D_{sk} + S_{s,[1],k}; ST_{s+1,k-1} + p_{[s+1],k-1}) \quad (9)$$

For  $k = 1$  and  $k = m$ , the formulation excludes the  $k - 1$  and  $k + 1$  terms, respectively. In Fig. 2 it is shown how  $ED_{tk}$  and  $ST_{s+1,k}$  are represented in a GANTT chart.

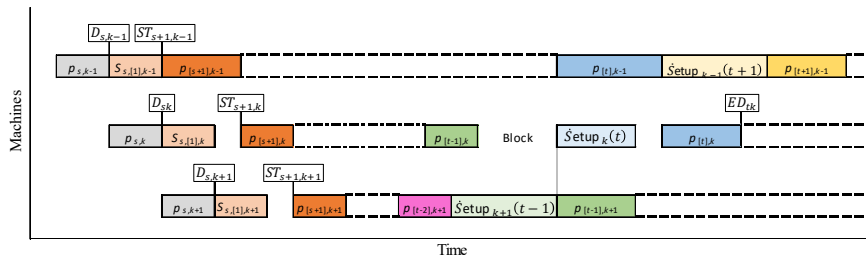


Fig. 2.  $ED_{tk}$  and  $ST_{s+1,k}$  representation in a GANTT chart

A different lower bound is proposed for each of the criterions.

### 3.1.1. Total Flow Time

According to Pinedo (2008) the SPT rule is optimal for total flow time criterion on a single machine, thus it is possible to obtain a lower bound for the flow shop problem. The lower bound of a partial sequence  $\sigma$  of size  $s$  is estimated as follows:

$$LB^{TFT} = TFT(\sigma) + \max_{1 \leq k \leq m} LB_k^{TFT} \quad (10)$$

$$LB_k^{TFT} = \sum_{t=s+1}^n \left( ED_{tk} + \sum_{r=k+1}^m p_{tr} \right) \quad (11)$$

### 3.1.2. Total Tardiness

The proposed lower bound for the total tardiness is based on the idea proposed by Chung et al. (2006) to underestimate the tardiness.

$$LB^{TT} = TT(\sigma) + \max_{1 \leq k \leq m} LB_k \quad (12)$$

$$LB_k^{TT} = \sum_{t=s+1}^n \max \left( ED_{tk} + \sum_{r=k+1}^m q_{[t-s],r} - d_{[t-s]}; 0 \right) \quad (13)$$

## 4. Computational Results

In this section the computational tests performed to evaluate the effectiveness of the proposed lower bounds (LB) are described. The problem instances database from Ronconi & Armentano (2001), in which the processing times were randomly generated in a uniform distribution varying from 1 to 99, were used for the computational tests. This database consists of 27 different problem classes, every one varying in number of jobs and machines, each with 20 unique problems, totaling 540 instances.

Setup times are not included in the database provided by Ronconi & Armentano (2001). Thus, to evaluate the impact of the setup times in the lower bounds, four different databases for the setup times were randomly generated for these tests. To form these databases, the range of values over which the setup times were uniformly distributed were:

- Database 1: from 01 to 09;
- Database 2: from 01 to 49;
- Database 3: from 01 to 99;
- Database 4: from 01 to 124.

The ranges of the setup times were defined roughly to 10%; 50%; 100%; and 150% of the processing time respectively. Thereby, it is possible to evaluate the influence of the setup time on the lower bounds. E.g., if the results for the problems of database 1 and the results of database 4 are similar, consequently it can be established that the value of the setup time does not influence or have little influence in determining the best lower bound. Each of the setup time databases was combined with the processing time database, totaling four different databases, each with 540 instances. This results in 2160 different problems.

As proposed by Ronconi & Armentano (2001), the due dates of the jobs were uniformly distributed between  $P(1 - TF - DR/2)$  and  $P(1 - TF + DR/2)$ . Where,  $TF$  is the Tardiness Factor,  $DR$  is the Dispersion Range of the due dates, and  $P$  is a lower bound of the problem with unlimited buffer.  $P$  is defined by Eq. (14), which is an adaptation of the equation proposed by Ronconi & Armentano (2001). The adaptation was required because the original equation did not consider setup times, thus, the due date would be too short to evaluate the method.

$$P = \max \left( \max_{1 \leq k \leq m} \sum_{j=1}^n p_{jk} + \min_{1 \leq j \leq n} \sum_{q=1}^{k-1} p_{jq} + \min_{1 \leq j \leq n} \sum_{q=k+1}^m p_{jq}; \max_{1 \leq j \leq n} \sum_{k=1}^m p_{jk} \right) + \frac{\sum_{k=1}^m \sum_{j=1}^n \sum_{i=1}^n S_{ijk}}{nm} \quad (14)$$

These scenarios represent different configurations by changing the values of  $TF$  and  $DR$ , as follows:

- Scenario 1: Low Tardiness Factor ( $TF = 0.2$ ) and small Dispersion Range ( $DR = 0.6$ );
- Scenario 2: Low Tardiness Factor ( $TF = 0.2$ ) and wide Dispersion Range ( $DR = 1.2$ );
- Scenario 3: High Tardiness Factor ( $TF = 0.4$ ) and small Dispersion Range ( $DR = 0.6$ );
- Scenario 4: High Tardiness Factor ( $TF = 0.4$ ) and wide Dispersion Range ( $DR = 1.2$ ).

The experimentation codes were written in C and ran on an Intel Core i7-8700K 3.7GHz, 16GB RAM DDR4 3000MHz using Microsoft Visual Studio Community 2017. A time limit of 3600 seconds was set for the execution of the algorithm.

To analyze the efficiency of the different lower bounds, the relative percentage deviation was calculated for both the average number of created nodes and the computational times, applying the following equation:

$$RPD_{variation} = \frac{\sigma_{variation} - \sigma^*}{\sigma^*} * 100 \quad (15)$$

In Equation 15,  $\sigma_{variation}$  is the mean value of either the number of created nodes or the computational time obtained with the LB tested and  $\sigma^*$  is the best value obtained among all the LB variations.

#### 4.1. Total Flow Time

Three variations of the proposed lower bound ( $LB^{TFT}$ ) were compared with one another to test the efficiency of the method. This was necessary as no other lower bound for the  $F_m|block, S_{ijk}|\sum C_{jm}$  was found in the literature. The goal of these tests is to analyze which varieties of the  $LB^{TFT}$  solve the problems in less computational time. The three proposed variations are:

- $LB_1^{TFT}$ :  $S_{i[1]k}$  and  $S_{[1]jk}$  were defined as zero and were not included in the searches;
- $LB_2^{TFT}$ :  $Setup_k(j)$  was defined as zero and was not included in the searches;
- $LB_3^{TFT}$ : LB was fully calculated.

Every problem was solved using each one of the  $LB^{TFT}$  variations, and all algorithms were conducted under the same circumstances. The average CPU time consumed, the average number of nodes that needed to be created, and the number of unsolved problems considering the setup times databases 1, 2, 3, and 4 are displayed on Tables 1, 2, 3, and 4, respectively.

**Table 1**  
Comparison of the variations of  $LB^{TFT}$  – Database 1

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$
10	2	581	576	<b>530</b>	0.01	<b>0.008</b>	0.01	<b>0</b>	<b>0</b>	<b>0</b>
10	3	8760	8510	<b>8390</b>	0.071	<b>0.06</b>	0.069	<b>0</b>	<b>0</b>	<b>0</b>
10	4	44300	40400	<b>40200</b>	0.293	<b>0.223</b>	0.28	<b>0</b>	<b>0</b>	<b>0</b>
10	5	382000	<b>381000</b>	382000	1.904	<b>1.722</b>	1.944	<b>0</b>	<b>0</b>	<b>0</b>
10	7	696000	<b>661000</b>	672000	3.28	<b>3.185</b>	3.398	<b>0</b>	<b>0</b>	<b>0</b>
10	10	<b>1560000</b>	1580000	1600000	<b>7.891</b>	8.873	8.768	<b>0</b>	<b>0</b>	<b>0</b>
12	2	2170	2180	<b>1930</b>	0.061	<b>0.043</b>	0.058	<b>0</b>	<b>0</b>	<b>0</b>
12	3	99500	82700	<b>81300</b>	0.84	<b>0.575</b>	0.689	<b>0</b>	<b>0</b>	<b>0</b>
12	4	5510000	5260000	<b>5210000</b>	26.32	<b>26.221</b>	28.437	<b>0</b>	<b>0</b>	<b>0</b>
12	5	13500000	<b>13200000</b>	13300000	66.189	<b>63.332</b>	67.047	<b>0</b>	<b>0</b>	<b>0</b>
12	7	41600000	41500000	<b>40900000</b>	223.579	<b>205.25</b>	224.6	<b>0</b>	<b>0</b>	<b>0</b>
12	10	90900000	<b>77800000</b>	79400000	539.634	<b>437.121</b>	490.666	<b>0</b>	<b>0</b>	<b>0</b>
14	2	16900	18100	<b>15100</b>	0.644	<b>0.456</b>	0.578	<b>0</b>	<b>0</b>	<b>0</b>
14	3	8360000	<b>2840000</b>	8130000	57.063	<b>21.574</b>	57.114	<b>0</b>	<b>0</b>	<b>0</b>
14	4	<b>73100000</b>	111000000	109000000	<b>427.835</b>	550.287	595.402	<b>1</b>	<b>2</b>	<b>2</b>
14	5	205000000	<b>193000000</b>	<b>193000000</b>	1191.026	<b>1076.282</b>	1129.617	<b>3</b>	<b>2</b>	<b>3</b>
14	7	299000000	307000000	<b>289000000</b>	2101.254	<b>1976.018</b>	2098.374	<b>8</b>	<b>7</b>	<b>8</b>
14	10	589000000	614000000	<b>575000000</b>	3510.736	<b>3495.732</b>	3511.206	<b>19</b>	<b>19</b>	<b>19</b>
16	2	104000	117000	<b>92200</b>	5.135	<b>3.696</b>	4.657	<b>0</b>	<b>0</b>	<b>0</b>
16	3	3460000	3530000	<b>3180000</b>	129.559	<b>89.907</b>	119.861	<b>0</b>	<b>0</b>	<b>0</b>
16	4	165000000	<b>141000000</b>	160000000	1105.745	<b>878.71</b>	1065.442	<b>4</b>	<b>3</b>	<b>4</b>
18	2	843000	1030000	<b>750000</b>	55.86	<b>40.276</b>	51.584	<b>0</b>	<b>0</b>	<b>0</b>
18	3	<b>27100000</b>	37800000	34600000	840.463	<b>711.829</b>	873.607	<b>0</b>	<b>1</b>	<b>1</b>
18	4	<b>204000000</b>	270000000	225000000	2410.37	<b>2313.624</b>	2359.329	<b>9</b>	<b>10</b>	<b>9</b>
20	2	6180000	7690000	<b>5380000</b>	495.834	<b>381.78</b>	452.126	<b>0</b>	<b>0</b>	<b>0</b>
20	3	72400000	62400000	<b>51000000</b>	2653.069	<b>2306.895</b>	2488.527	<b>12</b>	<b>9</b>	<b>10</b>
20	4	<b>256000000</b>	281000000	260000000	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
Average		76439526.33	80479313.56	<b>76138653.70</b>	720.54	<b>673.84</b>	712.35	2.81	<b>2.70</b>	2.81

**Table 2**  
Comparison of the variations of  $LB^{TFT}$  – Database 2

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$
10	2	1420	1310	<b>1100</b>	0.028	<b>0.018</b>	0.02	<b>0</b>	<b>0</b>	<b>0</b>
10	3	20800	<b>19600</b>	24400	0.191	<b>0.152</b>	0.187	<b>0</b>	<b>0</b>	<b>0</b>
10	4	72100	48800	<b>45100</b>	0.544	<b>0.387</b>	0.432	<b>0</b>	<b>0</b>	<b>0</b>
10	5	<b>206000</b>	242000	214000	1.382	1.355	<b>1.333</b>	<b>0</b>	<b>0</b>	<b>0</b>
10	7	519000	<b>496000</b>	513000	3.305	<b>2.867</b>	3.095	<b>0</b>	<b>0</b>	<b>0</b>
10	10	1310000	<b>1200000</b>	1210000	<b>6.753</b>	6.917	7.048	<b>0</b>	<b>0</b>	<b>0</b>
12	2	12400	12900	<b>9590</b>	<b>0.183</b>	0.248	0.277	<b>0</b>	<b>0</b>	<b>0</b>
12	3	<b>703000</b>	821000	800000	5.097	4.963	<b>4.076</b>	<b>0</b>	<b>0</b>	<b>0</b>
12	4	4090000	<b>2430000</b>	2790000	29.105	<b>17.793</b>	21.131	<b>0</b>	<b>0</b>	<b>0</b>
12	5	<b>5250000</b>	6080000	6140000	39.324	<b>37.569</b>	41.504	<b>0</b>	<b>0</b>	<b>0</b>
12	7	<b>24600000</b>	25500000	25400000	147.206	<b>142.323</b>	153.871	<b>0</b>	<b>0</b>	<b>0</b>
12	10	64900000	51100000	<b>50300000</b>	447.156	<b>335.646</b>	359.675	<b>0</b>	<b>0</b>	<b>0</b>
14	2	69600	74100	<b>54900</b>	2.657	<b>1.658</b>	2.177	<b>0</b>	<b>0</b>	<b>0</b>
14	3	8220000	6020000	<b>5760000</b>	97.614	<b>70.584</b>	72.341	<b>0</b>	<b>0</b>	<b>0</b>
14	4	66200000	<b>32600000</b>	34200000	550.658	<b>300.324</b>	369.081	1	<b>0</b>	<b>0</b>
14	5	178000000	<b>135000000</b>	161000000	1312.986	<b>985.334</b>	1193.124	4	2	4
14	7	<b>302000000</b>	394000000	315000000	2661.016	2828.815	<b>2654.844</b>	<b>11</b>	<b>11</b>	<b>11</b>
14	10	484000000	484000000	<b>447000000</b>	3365.358	<b>3238.832</b>	3308.02	18	16	17
16	2	520000	595000	<b>407000</b>	27.855	<b>19.407</b>	22.54	<b>0</b>	<b>0</b>	<b>0</b>
16	3	42300000	33900000	<b>31400000</b>	724.443	<b>539.128</b>	646.615	2	1	1
16	4	262000000	198000000	<b>192000000</b>	2491.518	<b>1731.481</b>	1995.478	9	7	7
18	2	4530000	5420000	<b>3550000</b>	314.373	<b>221.093</b>	251.807	<b>0</b>	<b>0</b>	<b>0</b>
18	3	122000000	174000000	<b>113000000</b>	3171.886	<b>2801.018</b>	2947.936	15	12	14
18	4	120000000	150000000	<b>99800000</b>	3576.416	<b>3407.678</b>	3501.395	19	16	18
20	2	21800000	30700000	<b>18600000</b>	2053.918	<b>1574.208</b>	1736.984	3	2	2
20	3	59100000	86000000	<b>57200000</b>	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
20	4	297000000	280000000	<b>275000000</b>	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
Average		76645345.19	77713359.63	<b>68200707.04</b>	1045.59	<b>943.33</b>	981.30	4.52	<b>3.96</b>	4.22

**Table 3**  
Comparison of the variations of  $LB^{TFT}$  – Database 3

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$
10	2	2350	2050	<b>1580</b>	0.039	0.032	<b>0.03</b>	<b>0</b>	<b>0</b>	<b>0</b>
10	3	25300	17200	<b>17100</b>	0.274	<b>0.177</b>	0.205	<b>0</b>	<b>0</b>	<b>0</b>
10	4	49500	49300	<b>38400</b>	0.542	<b>0.421</b>	0.446	<b>0</b>	<b>0</b>	<b>0</b>
10	5	153000	124000	<b>118000</b>	1.284	<b>0.973</b>	1.068	<b>0</b>	<b>0</b>	<b>0</b>
10	7	405000	296000	<b>293000</b>	2.272	<b>2.124</b>	2.431	<b>0</b>	<b>0</b>	<b>0</b>
10	10	<b>698000</b>	808000	771000	<b>4.728</b>	5.416	5.95	<b>0</b>	<b>0</b>	<b>0</b>
12	2	12000	11500	<b>8660</b>	0.317	<b>0.218</b>	0.26	<b>0</b>	<b>0</b>	<b>0</b>
12	3	255000	<b>200000</b>	250000	3.771	<b>2.31</b>	2.953	<b>0</b>	<b>0</b>	<b>0</b>
12	4	<b>603000</b>	748000	639000	10.668	<b>8.469</b>	9.621	<b>0</b>	<b>0</b>	<b>0</b>
12	5	3140000	<b>2330000</b>	2360000	35.531	<b>22.606</b>	27.407	<b>0</b>	<b>0</b>	<b>0</b>
12	7	11300000	11100000	<b>10700000</b>	97.695	<b>82.675</b>	90.788	<b>0</b>	<b>0</b>	<b>0</b>
12	10	33200000	<b>20600000</b>	22100000	266.644	<b>171.832</b>	218.601	<b>0</b>	<b>0</b>	<b>0</b>
14	2	76200	76200	<b>55600</b>	2.665	<b>1.983</b>	2.378	<b>0</b>	<b>0</b>	<b>0</b>
14	3	<b>1480000</b>	2290000	1590000	<b>36.153</b>	37.268	40.787	<b>0</b>	<b>0</b>	<b>0</b>
14	4	24100000	<b>10500000</b>	16500000	328.047	<b>162.731</b>	239.095	<b>0</b>	<b>0</b>	<b>0</b>
14	5	104000000	<b>46500000</b>	65900000	1052.792	<b>554.925</b>	795.59	2	<b>0</b>	1
14	7	<b>216000000</b>	223000000	219000000	2088.795	<b>1897.037</b>	2051.695	7	7	7
14	10	365000000	358000000	<b>337000000</b>	3335.304	<b>3050.372</b>	3238.649	15	14	14
16	2	1270000	1060000	<b>728000</b>	57.811	<b>33.615</b>	37.202	<b>0</b>	<b>0</b>	<b>0</b>
16	3	31500000	33000000	<b>28100000</b>	619.392	<b>455.64</b>	527.823	1	1	1
16	4	104000000	110000000	<b>92800000</b>	2197.687	<b>1757.055</b>	1922.422	6	6	7
18	2	3480000	4190000	<b>2640000</b>	253.645	<b>167.221</b>	196.373	<b>0</b>	<b>0</b>	<b>0</b>
18	3	61500000	89000000	<b>58100000</b>	3342.785	<b>2933.593</b>	3050.296	14	11	10
18	4	<b>140000000</b>	183000000	<b>140000000</b>	3600.001	<b>3546.032</b>	3592.311	20	19	19
20	2	21600000	30500000	<b>18600000</b>	2077.88	<b>1618.644</b>	1799.302	6	1	2
20	3	<b>62200000</b>	93400000	64100000	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
20	4	144000000	147000000	<b>125000000</b>	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
Average		49261087.04	50659342.59	<b>44718901.48</b>	985.80	<b>878.27</b>	927.91	4.11	<b>3.67</b>	3.74

**Table 4**  
Comparison of the variations of  $LB^{TFT}$  – Database 4

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$
10	2	1860	1560	<b>1200</b>	0.033	<b>0.022</b>	0.024	<b>0</b>	<b>0</b>	<b>0</b>
10	3	13200	<b>11600</b>	17900	0.174	<b>0.156</b>	0.187	<b>0</b>	<b>0</b>	<b>0</b>
10	4	51900	52600	<b>40100</b>	0.524	<b>0.417</b>	0.419	<b>0</b>	<b>0</b>	<b>0</b>
10	5	111000	96500	<b>91900</b>	1.108	<b>0.834</b>	0.94	<b>0</b>	<b>0</b>	<b>0</b>
10	7	305000	<b>242000</b>	249000	2.849	<b>2.016</b>	2.415	<b>0</b>	<b>0</b>	<b>0</b>
10	10	691000	<b>526000</b>	554000	4.369	<b>4.197</b>	4.892	<b>0</b>	<b>0</b>	<b>0</b>
12	2	17000	16500	<b>11900</b>	<b>0.242</b>	0.318	0.341	<b>0</b>	<b>0</b>	<b>0</b>
12	3	168000	135000	<b>121000</b>	3.159	<b>2.036</b>	2.497	<b>0</b>	<b>0</b>	<b>0</b>
12	4	859000	646000	<b>593000</b>	10.668	<b>5.908</b>	8.88	<b>0</b>	<b>0</b>	<b>0</b>
12	5	1060000	<b>944000</b>	1260000	16.945	<b>9.8</b>	17.235	<b>0</b>	<b>0</b>	<b>0</b>
12	7	6850000	<b>6480000</b>	6530000	70.271	<b>57.843</b>	67.667	<b>0</b>	<b>0</b>	<b>0</b>
12	10	21100000	<b>17900000</b>	18400000	222.954	<b>164.728</b>	196.393	<b>0</b>	<b>0</b>	<b>0</b>
14	2	90300	95800	<b>65900</b>	1.999	<b>1.874</b>	2.691	<b>0</b>	<b>0</b>	<b>0</b>
14	3	4120000	3060000	<b>2000000</b>	62.388	43.66	<b>41.695</b>	<b>0</b>	<b>0</b>	<b>0</b>
14	4	<b>26700000</b>	43300000	31200000	361.938	401.366	<b>358.702</b>	<b>0</b>	<b>0</b>	<b>0</b>
14	5	16800000	<b>14400000</b>	<b>14400000</b>	376.944	<b>255.084</b>	337.281	<b>0</b>	<b>0</b>	<b>0</b>
14	7	<b>20800000</b>	23400000	21500000	2362.521	<b>2197.001</b>	2275.122	<b>8</b>	<b>9</b>	<b>8</b>
14	10	260000000	259000000	<b>236000000</b>	3452.525	<b>3085.414</b>	3310.143	<b>16</b>	<b>11</b>	<b>14</b>
16	2	464000	560000	<b>365000</b>	25.5	<b>18.455</b>	20.377	<b>0</b>	<b>0</b>	<b>0</b>
16	3	10900000	10900000	<b>7580000</b>	435.885	<b>279.153</b>	317.886	<b>0</b>	<b>0</b>	<b>0</b>
16	4	81400000	<b>73200000</b>	74600000	2695.172	<b>1872.869</b>	2437.614	<b>9</b>	<b>4</b>	<b>6</b>
18	2	3680000	4570000	<b>2730000</b>	265.789	<b>178.755</b>	203.287	<b>0</b>	<b>0</b>	<b>0</b>
18	3	45400000	67100000	<b>44600000</b>	2864.808	<b>2692.193</b>	2823.943	<b>13</b>	<b>9</b>	<b>12</b>
18	4	132000000	<b>121000000</b>	130000000	3531.263	<b>3492.868</b>	3507.656	<b>19</b>	<b>19</b>	<b>19</b>
20	2	14700000	22700000	<b>12300000</b>	1417.233	<b>1157.732</b>	1159.092	<b>1</b>	<b>1</b>	<b>1</b>
20	3	64900000	93100000	<b>62300000</b>	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
20	4	140000000	149000000	<b>100000000</b>	<b>3600.001</b>	<b>3600.001</b>	<b>3600.001</b>	<b>20</b>	<b>20</b>	<b>20</b>
Average		38532676.30	41593983.70	<b>35592996.30</b>	940.27	<b>856.47</b>	899.90	3.93	<b>3.44</b>	3.70

The RPD was calculated for all databases separately, and the results are shown in Table 5.

**Table 5**  
RPD of the variations of the  $LB^{TFT}$  for all databases

Database	Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$	$LB_1^{TFT}$	$LB_2^{TFT}$	$LB_3^{TFT}$
Database 1	<b>0.00</b>	0.06	<b>0.00</b>	0.07	<b>0.00</b>	0.06	0.04	<b>0.00</b>	0.04
Database 2	0.12	0.14	<b>0.00</b>	0.11	<b>0.00</b>	0.04	0.14	<b>0.00</b>	0.07
Database 3	0.10	0.13	<b>0.00</b>	0.12	<b>0.00</b>	0.06	0.12	<b>0.00</b>	0.02
Database 4	0.08	0.17	<b>0.00</b>	0.10	<b>0.00</b>	0.05	0.14	<b>0.00</b>	0.08
Average	0.08	0.13	<b>0.00</b>	0.10	<b>0.00</b>	0.05	0.11	<b>0.00</b>	0.05

From Tables 1, 2, 3, and 4 it is possible to notice that the higher is the range of values for the setup times, the smaller is the number of nodes. This occurred for all variations of the  $LB^{TFT}$ . Therefore, it can be concluded that this element is not significant for the tests. From Table 5 it is possible to acknowledge that the lower bound  $LB_3^{TFT}$  was the one that reduced the most the number of nodes for all databases. However,  $LB_2^{TFT}$  solved more problems in less computational time.  $LB_1^{TFT}$  could not perform very well for all databases.

#### 4.2. Total Tardiness

Three variations of the proposed lower bound ( $LB^{TT}$ ) were compared with one another to test the efficiency of the method. This had to be done because no other lower bound for the  $F_m|block, S_{ijk}|\sum T_j$  was found in the literature. The objective of the tests is to analyze which varieties of the  $LB^{TT}$  solve the problems in less computational time. The three proposed variations are similar to the ones proposed for the Total Flow Time criterion.

As the range of the setup time did not influence the results of the total flow time problems, only the 1-125 distribution was used for the total tardiness tests. Each of the problems was combined with the four different scenarios, totaling 2160 different problems.

Every problem was solved using every one of the variations of  $LB^{TT}$ , and all algorithms were conducted under the same circumstances. The average CPU time consumed, the average number of nodes that needed to be created, and the number of unsolved problems considering database 1, 2, 3, and 4 for the due date are displayed on Tables 6, 7, 8, and 9, respectively.



**Table 6**  
Comparison of the variations of  $LB^{TT}$  – Scenario 1

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$
10	2	4780.00	5240.00	<b>4750.00</b>	0.05	<b>0.05</b>	0.07	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	3	13500.00	12200.00	<b>11100.00</b>	0.09	<b>0.08</b>	0.09	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	4	49000.00	<b>36900.00</b>	37500.00	0.21	<b>0.16</b>	0.21	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	5	126000.00	<b>120000.00</b>	133000.00	<b>0.48</b>	0.49	0.60	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	7	128000.00	<b>125000.00</b>	139000.00	0.53	<b>0.49</b>	0.64	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	10	220000.00	<b>210000.00</b>	220000.00	1.13	<b>1.01</b>	1.25	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	2	55000.00	61400.00	<b>54300.00</b>	0.74	<b>0.61</b>	0.76	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	3	161000.00	219000.00	<b>160000.00</b>	<b>1.81</b>	1.96	1.96	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	4	626000.00	<b>514000.00</b>	591000.00	3.76	<b>3.16</b>	3.74	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	5	1550000.00	<b>1490000.00</b>	1600000.00	8.00	<b>7.03</b>	8.38	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	7	4470000.00	<b>4310000.00</b>	4470000.00	21.71	<b>17.32</b>	21.06	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	10	17000000.00	<b>11900000.00</b>	17000000.00	76.24	<b>49.86</b>	79.68	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	2	1140000.00	1240000.00	<b>1110000.00</b>	20.11	<b>15.75</b>	20.26	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	3	3100000.00	3120000.00	<b>3040000.00</b>	42.07	<b>32.61</b>	42.36	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	4	89800000.00	88700000.00	<b>83900000.00</b>	296.07	<b>280.34</b>	307.70	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
14	5	29600000.00	<b>29300000.00</b>	29600000.00	167.34	<b>140.54</b>	177.43	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	7	<b>11100000.00</b>	<b>11100000.00</b>	13400000.00	494.00	<b>465.26</b>	613.69	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
14	10	<b>21800000.00</b>	22800000.00	21900000.00	1333.92	<b>1271.46</b>	1387.24	3.00	<b>1.00</b>	3.00
16	2	22900000.00	26000000.00	<b>22500000.00</b>	503.51	<b>401.18</b>	494.22	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	3	36000000.00	42500000.00	<b>35600000.00</b>	959.38	<b>775.63</b>	990.13	2.00	<b>0.00</b>	2.00
16	4	192000000.00	194000000.00	<b>187000000.00</b>	1749.47	<b>1524.07</b>	1797.74	<b>3.00</b>	<b>3.00</b>	5.00
18	2	104000000.00	148000000.00	<b>101000000.00</b>	3207.50	<b>3173.92</b>	3227.69	14.00	<b>12.00</b>	14.00
18	3	109000000.00	149000000.00	<b>106000000.00</b>	3462.08	<b>3357.03</b>	3463.34	18.00	<b>17.00</b>	18.00
18	4	257000000.00	318000000.00	<b>249000000.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>
20	2	91200000.00	139000000.00	<b>87400000.00</b>	3438.04	<b>3431.20</b>	3434.36	<b>19.00</b>	<b>19.00</b>	<b>19.00</b>
20	3	84000000.00	118000000.00	<b>81100000.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>
20	4	280000000.00	315000000.00	<b>265000000.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>
Average		61227528.89	71476434.81	<b>60358172.22</b>	984.75	<b>953.75</b>	995.36	4.48	<b>4.22</b>	4.56

**Table 7**  
Comparison of the variations of  $LB^{TT}$  – Scenario 2

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$
10	2	484.00	488.00	<b>480.00</b>	0.01	<b>0.00</b>	0.01	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	3	2590.00	2600.00	<b>2580.00</b>	0.03	<b>0.02</b>	0.02	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	4	<b>33600.00</b>	33700.00	34600.00	<b>0.12</b>	0.13	0.13	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	5	95700.00	96000.00	<b>95600.00</b>	0.36	<b>0.29</b>	0.35	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	7	<b>95300.00</b>	<b>95300.00</b>	<b>95300.00</b>	0.36	<b>0.33</b>	0.42	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	10	<b>321000.00</b>	<b>321000.00</b>	<b>321000.00</b>	1.45	<b>1.23</b>	1.58	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	2	4330.00	4650.00	<b>4290.00</b>	0.06	<b>0.05</b>	0.06	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	3	<b>217000.00</b>	<b>217000.00</b>	<b>217000.00</b>	0.88	<b>0.73</b>	0.89	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	4	340000.00	<b>319000.00</b>	320000.00	1.81	<b>1.17</b>	1.47	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	5	1600000.00	<b>1460000.00</b>	1580000.00	5.54	<b>5.28</b>	6.10	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	7	<b>1290000.00</b>	1890000.00	<b>1290000.00</b>	<b>5.73</b>	6.83	6.13	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	10	<b>14900000.00</b>	<b>14900000.00</b>	<b>14900000.00</b>	63.70	<b>58.60</b>	67.56	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	2	20100.00	20500.00	<b>19900.00</b>	0.41	<b>0.28</b>	0.43	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	3	<b>448000.00</b>	451000.00	<b>448000.00</b>	<b>3.13</b>	3.18	3.24	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	4	<b>28500000.00</b>	28700000.00	28700000.00	96.81	<b>88.98</b>	101.37	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	5	<b>67100000.00</b>	<b>67100000.00</b>	<b>67100000.00</b>	234.25	<b>218.57</b>	233.38	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	7	142000000.00	146000000.00	<b>140000000.00</b>	503.47	<b>494.63</b>	517.26	<b>2.00</b>	<b>2.00</b>	<b>2.00</b>
14	10	197000000.00	199000000.00	<b>196000000.00</b>	1023.60	<b>981.50</b>	1056.56	<b>2.00</b>	<b>2.00</b>	<b>2.00</b>
16	2	69600.00	71500.00	<b>69300.00</b>	1.03	<b>0.75</b>	1.05	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	3	<b>4150000.00</b>	4170000.00	<b>4150000.00</b>	29.35	<b>22.99</b>	29.96	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	4	125000000.00	126000000.00	<b>121000000.00</b>	565.00	<b>524.82</b>	574.08	<b>2.00</b>	<b>2.00</b>	<b>2.00</b>
18	2	2040000.00	2050000.00	<b>2030000.00</b>	16.75	<b>12.86</b>	18.49	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
18	3	<b>15400000.00</b>	15500000.00	<b>15400000.00</b>	139.11	<b>104.06</b>	139.63	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
18	4	<b>288000000.00</b>	313000000.00	290000000.00	1946.61	<b>1795.10</b>	1999.94	9.00	<b>7.00</b>	9.00
20	2	9780000.00	12000000.00	<b>9590000.00</b>	384.27	<b>321.47</b>	389.54	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
20	3	<b>131000000.00</b>	141000000.00	<b>131000000.00</b>	1065.12	<b>934.17</b>	1069.22	<b>4.00</b>	<b>4.00</b>	<b>4.00</b>
20	4	239000000.00	279000000.00	<b>237000000.00</b>	2383.76	<b>2246.78</b>	2390.50	11.00	<b>10.00</b>	11.00
Average		46978063.11	50126027.33	<b>46717335.19</b>	313.80	<b>289.81</b>	318.87	1.15	<b>1.04</b>	1.15

**Table 8**  
Comparison of the variations of  $LB^{TT}$  – Scenario 3

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$
10	2	<b>1160.00</b>	1310.00	1170.00	0.01	<b>0.01</b>	0.02	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	3	<b>4030.00</b>	4290.00	4160.00	0.03	<b>0.03</b>	0.04	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	4	<b>74800.00</b>	76300.00	<b>74800.00</b>	0.27	<b>0.21</b>	0.25	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	5	192000.00	192000.00	<b>190000.00</b>	0.63	<b>0.56</b>	0.66	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	7	295000.00	<b>294000.00</b>	295000.00	1.05	<b>0.94</b>	1.14	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	10	588000.00	<b>587000.00</b>	588000.00	2.44	<b>2.30</b>	2.63	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	2	15800.00	17400.00	<b>15500.00</b>	0.24	<b>0.19</b>	0.28	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	3	<b>98000.00</b>	172000.00	119000.00	<b>0.72</b>	0.79	0.80	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	4	289000.00	<b>273000.00</b>	281000.00	9.16	<b>9.08</b>	9.61	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	5	351000.00	265000.00	<b>236000.00</b>	13.06	10.43	<b>10.01</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	7	1000000.00	1000000.00	<b>999000.00</b>	37.13	<b>34.11</b>	38.33	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	10	3600000.00	3650000.00	<b>3500000.00</b>	148.18	<b>147.94</b>	153.43	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	2	146000.00	167000.00	<b>143000.00</b>	3.31	<b>2.47</b>	3.23	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	3	404000.00	421000.00	<b>403000.00</b>	7.69	<b>5.66</b>	8.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	4	5000000.00	<b>4900000.00</b>	5000000.00	179.28	<b>166.34</b>	195.97	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	5	15100000.00	17300000.00	<b>14600000.00</b>	<b>523.33</b>	574.06	532.01	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
14	7	42000000.00	42800000.00	<b>40200000.00</b>	1437.28	<b>1421.48</b>	1439.72	<b>7.00</b>	<b>7.00</b>	<b>7.00</b>
14	10	43000000.00	45500000.00	<b>42700000.00</b>	2017.61	<b>1945.54</b>	2024.71	<b>9.00</b>	<b>8.00</b>	<b>9.00</b>
16	2	158000.00	186000.00	<b>156000.00</b>	44.50	<b>36.61</b>	46.67	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	3	592000.00	644000.00	<b>577000.00</b>	137.76	<b>103.98</b>	144.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	4	<b>8860000.00</b>	9910000.00	10100000.00	501.02	<b>465.89</b>	571.22	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
18	2	1900000.00	2280000.00	<b>1870000.00</b>	733.53	<b>581.72</b>	747.42	<b>0.00</b>	<b>0.00</b>	1.00
18	3	6600000.00	<b>5250000.00</b>	6460000.00	1527.82	<b>1051.67</b>	1590.11	2.00	<b>1.00</b>	3.00
18	4	<b>16900000.00</b>	21700000.00	20700000.00	2838.19	<b>2423.27</b>	2935.56	11.00	<b>7.00</b>	12.00
20	2	7130000.00	10900000.00	<b>6800000.00</b>	3274.35	<b>3219.59</b>	3319.16	17.00	<b>16.00</b>	17.00
20	3	9150000.00	14200000.00	<b>7780000.00</b>	3600.00	<b>3474.41</b>	3600.00	20.00	<b>17.00</b>	20.00
20	4	11900000.00	24500000.00	<b>11100000.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>3600.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>
Average		64337732.96	76093048.15	<b>64163838.15</b>	764.39	<b>714.05</b>	776.85	3.26	<b>2.89</b>	3.37

**Table 9**  
Comparison of the variations of  $LB^{TT}$  – Scenario 4

Size		Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
n	m	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$
10	2	687.00	717.00	<b>683.00</b>	0.01	<b>0.01</b>	0.01	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	3	<b>2220.00</b>	2290.00	<b>2220.00</b>	0.02	<b>0.02</b>	0.02	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	4	<b>45600.00</b>	47900.00	49300.00	0.17	<b>0.15</b>	0.19	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	5	195000.00	192000.00	<b>191000.00</b>	0.61	<b>0.57</b>	0.67	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	7	301000.00	324000.00	<b>292000.00</b>	1.05	<b>0.99</b>	1.07	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
10	10	<b>617000.00</b>	621000.00	623000.00	<b>2.46</b>	2.70	2.69	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	2	5540.00	5970.00	<b>5460.00</b>	0.08	<b>0.07</b>	0.09	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	3	92400.00	92300.00	<b>91200.00</b>	0.57	<b>0.53</b>	0.80	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	4	<b>219000.00</b>	220000.00	224000.00	7.66	<b>6.70</b>	7.27	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	5	323000.00	<b>314000.00</b>	315000.00	12.63	<b>10.82</b>	12.72	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	7	<b>868000.00</b>	<b>868000.00</b>	<b>868000.00</b>	33.34	<b>29.25</b>	32.62	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
12	10	<b>2910000.00</b>	3050000.00	<b>2910000.00</b>	<b>122.75</b>	125.32	132.39	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	2	66500.00	70100.00	<b>65200.00</b>	1.37	<b>0.98</b>	1.39	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	3	319000.00	326000.00	<b>318000.00</b>	4.97	<b>3.44</b>	5.34	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	4	9540000.00	9590000.00	<b>9160000.00</b>	263.49	<b>262.85</b>	268.08	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
14	5	5610000.00	5610000.00	<b>5600000.00</b>	212.44	<b>207.40</b>	223.20	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
14	7	33400000.00	33100000.00	<b>32500000.00</b>	1190.85	<b>1144.73</b>	1191.39	<b>4.00</b>	<b>4.00</b>	<b>4.00</b>
14	10	52400000.00	52500000.00	<b>50400000.00</b>	2233.82	<b>2193.73</b>	2261.98	<b>10.00</b>	<b>10.00</b>	<b>10.00</b>
16	2	262000.00	278000.00	<b>260000.00</b>	7.12	<b>5.13</b>	7.64	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	3	2060000.00	2100000.00	<b>2050000.00</b>	137.63	<b>121.73</b>	147.73	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
16	4	<b>14700000.00</b>	14800000.00	<b>14700000.00</b>	694.45	<b>664.44</b>	722.31	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
18	2	924000.00	985000.00	<b>919000.00</b>	38.10	<b>26.67</b>	38.70	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
18	3	1330000.00	1410000.00	<b>1320000.00</b>	463.03	<b>331.32</b>	476.44	1.00	<b>0.00</b>	1.00
18	4	50800000.00	<b>45000000.00</b>	49300000.00	2655.60	<b>2319.31</b>	2683.73	12.00	<b>9.00</b>	12.00
20	2	2360000.00	3030000.00	<b>2300000.00</b>	1067.56	<b>906.62</b>	1085.97	4.00	<b>2.00</b>	4.00
20	3	17100000.00	18200000.00	<b>15700000.00</b>	2502.55	<b>2261.50</b>	2522.73	12.00	<b>9.00</b>	12.00
20	4	27100000.00	27400000.00	<b>25000000.00</b>	3340.09	<b>3162.65</b>	3351.77	17.00	<b>16.00</b>	17.00
Average		81852998.04	80550565.81	<b>78751372.70</b>	555.35	<b>510.73</b>	562.18	2.30	<b>1.93</b>	2.30

The RPD was calculated for all databases separately, and the results are shown in Table 10.

**Table 10**  
RPD of the variations of the  $LB^{TT}$  for all databases

Database	Ave. Node Count			Ave. CPU Time (ms)			Number of Unsolved		
	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$	$LB_1^{TT}$	$LB_2^{TT}$	$LB_3^{TT}$
Scenario 1	0.01	0.18	<b>0.00</b>	0.03	<b>0.00</b>	0.04	0.06	<b>0.00</b>	0.08
Scenario 2	0.01	0.07	<b>0.00</b>	0.08	<b>0.00</b>	0.10	0.11	<b>0.00</b>	0.11
Scenario 3	<b>0.00</b>	0.19	<b>0.00</b>	0.07	<b>0.00</b>	0.09	0.13	<b>0.00</b>	0.17
Scenario 4	0.04	0.02	<b>0.00</b>	0.09	<b>0.00</b>	0.10	0.19	<b>0.00</b>	0.19
Average	0.02	0.12	<b>0.00</b>	0.07	<b>0.00</b>	0.08	0.12	<b>0.00</b>	0.14

From Tables 6, 7, 8, and 9 it was possible to notice that wider dispersion rates of the due dates facilitate obtaining the optimal solution. On the other hand, higher tardiness factors make it more difficult to achieve the optimal solution, mostly in problems with more jobs. From Table 10 it was possible to point out that, similar to the results obtained for the total flow time problems,  $LB_2^{TT}$  solved more problems in less computational time. However,  $LB_3^{TT}$  was the lower bound that most reduced the number of nodes. In both cases  $LB_1^{TT}$  could not perform very well. Since the most noteworthy measure for the B&B algorithm is the computational time, it can be assumed that the search of  $\hat{Setup}_k(j)$  is inadequate for both problems considered in this paper.

## 5. Conclusion

In this paper it is considered a permutation flow shop scheduling problem with blocking and setup time constraint and with the total flow time and the total tardiness criteria, which are known to be NP-hard for problems with two or more machines. A machine based lower bound that considers the existence of blocking and idle times in the process is suggested for each of the problems ( $LB^{TFT}$  and  $LB^{TT}$ ), and they are both implemented in a branch and bound algorithm. Four databases for the setup times were generated, each with a specific range of values. The analysis shows that this range of values does not affect the effectiveness of the lower bounds.

Three different approaches to applying the proposed  $LB^{TFT}$  were compared with one another, because no other lower bound for the total flow time criterion was found in the literature. They were named  $LB_1^{TFT}$ ,  $LB_2^{TFT}$ , and  $LB_3^{TFT}$ . Overall,  $LB_1^{TFT}$  was the lower bound that got the highest computational time and was the variation that solved fewer problems within 3600 seconds, also got the second highest number of nodes. So, it can be deduced that it is important for the  $LB^{TFT}$  to incorporate both  $S_{i[1]k}$  and  $S_{[1]jk}$  in the search.

Overall,  $LB_2^{TFT}$  got the smallest value for both the computational time and the number of unsolved problems and  $LB_3^{TFT}$  was the one that had to create less nodes to solve the problems. This can imply that it is not interesting for the  $LB^{TFT}$  to incorporate  $\hat{Setup}_k(j)$  in the search.

Similar to what happened to the total flow time criterion, no other lower bound for the total tardiness criterion was found in the literature. So, as with the total flow time problems, three different approaches to applying the proposed  $LB^{TT}$  were compared with one another.

As the tests showed that the range value of the setup time did not affect the performance of the lower bounds in the total flow time problems, only one of the databases was used for the total tardiness tests. The results were also very similar.  $LB_2^{TT}$  was the one that got the smallest values for both the computational time and the number of unsolved problems. This can mean that including  $\hat{Setup}_k(j)$  in the search is not interesting for the  $LB^{TT}$  either.

However, for the total tardiness,  $LB_3^{TT}$  was the one that took the longest to solve the problems and was the variation that had the greater number of unsolved problems within 3600 seconds, even though it was the one that got the smallest number of nodes. So, it can be deduced that including both  $S_{i[1]k}$  and  $S_{[1]jk}$  in the calculus of the lower bound is very important for the  $LB^{TT}$ .

Also, regarding the total tardiness criterion, the lower bound showed to be more efficient for higher dispersion rate of the due dates.

For future works, it is suggested the development of a dominance rule to eliminate a larger number of nodes and the study of the use of efficient heuristics as initial upper bounds for the branch and bound algorithm. These proposals aim to enhance the algorithm, while applied along with  $LB_2^{TFT}$  and  $LB_2^{TT}$ . Moreover, other exploration rules, like the best bound rule, or a hybrid technique, may be analyzed. Acknowledging the efficiency of the proposed lower bounds, an additional suggestion is to apply the rules used to incorporate blocking constraints on these lower bounds into a general case with a limited buffer greater than or equal to zero.

## References

- Ahmadi, R. H., & Bagchi, U. (1990). Improved lower bounds for minimizing the sum of completion times of  $n$  jobs over  $m$  machines in a flow shop. *European Journal of Operational Research*, 44(3), 331-336.
- Bansal, S. P. (1977). Minimizing the sum of completion times of  $n$  jobs over  $m$  machines in a flowshop: A branch and bound approach. *A IIE Transactions*, 9(3), 306-311.
- Chung, C. S., Flynn, J., & Kirca, O. (2002). A branch and bound algorithm to minimize the total flow time for  $m$ -machine permutation flowshop problems. *International Journal of Production Economics*, 79(3), 185-196.
- Chung, C. S., Flynn, J., & Kirca, O. (2006). A branch and bound algorithm to minimize the total tardiness for  $m$ -machine permutation flowshop problems. *European Journal of Operational Research*, 174(1), 1-10.
- Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117-129.
- Gilmore, P. C., & Gomory, R. E. (1964). Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Operations Research*, 12(5), 655-679.
- Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. *Operations Research*, 13(3), 400-412.
- Mccormick, S., Pinedo, M., J. Shenker, S., & Wolf, B. (1989). Sequencing in an assembly line with blocking to minimize cycle time. *Operations Research*, 37(6), 925-935.
- Miyata, H. H., & Nagano, M. S. (2019). The blocking flow shop scheduling problem: A comprehensive and conceptual review. *Expert Systems with Applications*, 137(1), 130-156.
- Moslehi, G., & Khorasani, D. (2013). Optimizing blocking flow shop scheduling problem with total completion time criterion. *Computers & Operations Research*, 40(7), 1874-1883.
- Nagano, M. S., Robazzi, J. V. S., & Tomazella, C. P. (2020). An improved lower bound for the blocking permutation flow shop with total completion time criterion. *Computers & Industrial Engineering*, 146(1), 106511.
- Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the  $m$ -machine,  $n$ -job flow-shop sequencing problem. *Omega*, 11(1), 91-95.
- Pan, Q. K., & Wang, L. (2012). Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. *Omega*, 40(2), 218-229.
- Pan, Q.-K., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44(1), 41-50.
- Pinedo, M. L. (2008). *Scheduling: Theory, Algorithms, and Systems*. 3<sup>rd</sup> edn. Springer Publishing Company, New York.
- Reddi, S., & Ramamoorthy, C. (1972). On the flow-shop sequencing problem with no wait in process. *Journal of the Operational Research Society*, 23(3), 323-331.
- Rios-Mercado, R. Z., & Bard, J. F. (1999). A Branch-and-Bound Algorithm for Flowshop Scheduling with Setup Times. *IIE Transactions on Scheduling & Logistics*, 31(8), 721-731.
- Robazzi, J. V. S., Nagano, M. S., & Takano, M. I. (2021). A Branch-and-Bound Method to Minimize the Total Flow Time in a Permutation Flow Shop with Blocking and Setup Times. In: Rossit D.A., Tohmé F., Mejia Delgadillo G. (eds) Production Research. ICPR-Americas 2020. *Communications in Computer and Information Science*, vol 1407. Springer, Cham.
- Ronconi, D. P. (2004). A note on constructive heuristics for the flowshop problem with blocking. *International Journal of Production Economics*, 87(1), 39-48.
- Ronconi, D. P. (2005). A branch-and-bound algorithm to minimize the makespan in a flowshop with blocking. *Annals of Operations Research*, 138(1), 53-65.
- Ronconi, D. P., & Armentano, V. A. (2001). Lower bounding schemes for flowshops with blocking in-process. *Journal of the Operational Research Society*, 52(11), 1289-1297.
- Ronconi, D. P., & Birgin, E. G. (2012). Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. *Just-in-Time Systems*, 61(1), 91-105.
- Sanches, F. B., Takano, M. I., & Nagano, M. S. (2016). Evaluation of heuristics for a branch and bound algorithm to minimize the makespan in a flowshop with blocking. *Acta Scientiarum-Technology*, 38(3), 321-326.
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2), 278-285.
- Takano, M. I., & Nagano, M. S. (2017). A branch-and-bound method to minimize the makespan in a permutation flow shop with blocking and setup times. *Cogent Engineering*, 4(1), 1389638.
- Takano, M. I., & Nagano, M. S. (2019). Evaluating the performance of constructive heuristics for the blocking flow shop scheduling problem with setup times. *International Journal of Industrial Engineering Computations*, 10(1), 37-50.
- Takano, M. I., & Nagano, M. S. (2020). Solving the permutation flow shop problem with blocking and setup time constraints. *International Journal of Industrial Engineering Computations*, 11(3), 469-480.

