RAIRO-Oper. Res. 46 (2012) 125–147
DOI: 10.1051/ro/2012010

RAIRO Operations Research
www.rairo-ro.org

BRANCH-AND-BOUND ALGORITHM FOR TOTAL WEIGHTED TARDINESS MINIMIZATION ON PARALLEL MACHINES UNDER RELEASE DATES ASSUMPTIONS

Imed Kacem[1], Nizar Souayah[2] and Mohamed Haouari[2]

Abstract. This paper deals with the parallel-machine scheduling problem with the aim of minimizing the total (weighted) tardiness under the assumption of different release dates. This problem has been proven to be NP-hard. We introduce some new lower and upper bounds based on different approaches. We propose a branch-and-bound algorithm to solve the weighted and unweighted total tardiness. Computational experiments were performed on a large set of instances and the obtained results showed that our algorithms are efficient.

Keywords. Scheduling, weighted tardiness, parallel machines, branch-and-boun.

Mathematics Subject Classification. 90B35, 68M20, 90C57.

1. Introduction

In this paper, we study the identical parallel-machine scheduling problem to minimize the total (weighted) tardiness with release dates. This problem is denoted as the $P|r_j|\sum(w_j)T_j$ in the standard scheduling terminology.

The objective of this paper is to develop an exact method to solve this problem. The tardiness of a job j is defined as $T_j = \max(0, C_j - d_j)$, and the corresponding weighted tardiness is defined as $w_jT_j = w_j \max(0, C_j - d_j)$, where C_j, d_j and w_j are respectively the completion time, the due date and the weight of job j. Each machine cannot process more than one job at a given time and no job can

Received October 12, 2011. Accepted April 17, 2012.

[1] Université de Lorraine, LCOMS, UFR MIM, Ile du Saulcy, 57000 Metz, France.
imed.kacem@gmail.com

[2] École Polytechnique de Tunisie, 2078 La Marsa, Tunisia

Article published by EDP Sciences

© EDP Sciences, ROADEF, SMAI 2012

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

be preempted. The problem considered in this paper is at least NP-Hard, since the unweighted case on a single-machine was proved to be NP-hard by Rinnooy Kan [17].

Despite the importance of this criterion, a lot of research has been carried out on the single-machine problem. Given the aim of this paper, we can cite the following references.

Tiang *et al.* [22] presented some optimality properties based on the optimality of a special case, where the release dates and the due dates are *agreeable*.

Chu [9], developed some dominance rules and provided a lower bound by using the job preemption for the single-machine with release dates when the weights are equal. He also proposed a branch-and-bound algorithm for the same problem $1|r_j|\sum T_j$ and he incorporated the previous lower bound and an upper bound derived from a constructive heuristic algorithm. This procedure can solve difficult problems with up to 30 jobs and easy problems with up to 230 jobs.

Baptiste *et al.* [5] presented a branch-and-bound algorithm to minimize the total tardiness on a single-machine with arbitrary release dates. The lower bound used is an improvement of Chu's lower bound [9]. Some dominance properties are introduced based on the Emmons rules [11]. The computational results showed that the proposed branch-and-bound algorithm outperforms the best known procedures, especially the algorithm proposed by Chu. Several tests have been done to show the efficiency of the dominace properties.

Su and Chen [21] adapted some dominance properties from the literature to build a branch-and-bound algorithm and a heuristic procedures. They proposed a lower bound by combining Baptiste's lower bound procedure with Lawler's dynamic programming. The computational results showed the efficiency of their approach that can solve hard problems with up 50 jobs and easy problems with up to 500 jobs.

Recently, Baptiste *et al.* [6] presented some lower bounds for the parallel machine scheduling problem with release dates. The authors computed a lower bound for the problem $P|r_j|\sum(w_j)T_j$ using some relaxation techniques: release dates relaxation, reducing the problem to flow problems. They presented also a time mixed-integer formulation.

In the litterature, some papers dealt with the tardiness minimization problem on parallel-machines without release dates. We can cite: Elmaghraby and Park [10], Azizoglu and Kirka [3], Yalaoui and Chu [23] and Shim and Kim [18] that have proposed exact resolution.

Several heuristic algorithms have been developed also for the same problem, we have Alidaee and Rosa [1], Koulamas [12], Lee and Pinedo [13] and Mosheiov and Oron [14]. Finally, it is worthy to mention several references on tardiness minimization for shop scheduling (for example, Chen and Lin [7], Anderson and Nyirenda [2]).

This work proposes an exact algorithm for the identical parallel-machine scheduling problem in order to minimize the total weighted tardiness with release dates. In such an algorithm, several lower and upper bounds are incorporated.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

The paper is organized as follows. Section 2 describes the problem formulation. Section 3 presents the proposed lower bounds. Section 4 describes two resolution methods: heuristics and a branch-and-bound algorithm. Section 5 reports the computational results. Finally, some concluding remarks are given in Section 6.

2. Problem formulation

We have to schedule a set J of n jobs on m parallel machines. Each job j has a processing time p_j, a due date d_j, a weight w_j and a release date r_j. Each machine can process only one job at a given time and preemption is not allowed. A job can be scheduled on the machines at time t if t is greater or equal than its release date. For a given sequence, a job j is called tardy if its completion time C_j is strictly larger than its due date, its tardiness is $T_j = \max(0, C_j - d_j)$ and its weighted tardiness is $w_j \max(0, C_j - d_j)$.

Therefore, the total tardiness of the n jobs is given in equation (2.1).

$$\sum_{j=1}^{n} T_j = \sum_{i=1}^{n} \max(0, C_j - d_j) \tag{2.1}$$

and the corresponding total weighted tardiness is given in equation (2.2)

$$\sum_{j=1}^{n} w_j T_j = \sum_{i=1}^{n} w_j \max(0, C_j - d_j). \tag{2.2}$$

The objective is to find a sequence that minimizes $\sum_{j=1}^{n} T_j$ and $\sum_{j=1}^{n} w_j T_j$. Therefore, we can formulate our problems, as follows:

$$(P) \quad \text{minimize} \sum_{j=1}^{n} T_j$$

$$T_j \geq (C_j - d_j) \tag{2.3}$$

$$C_j \geq r_j + p_j, \tag{2.4}$$

$$(P_w) \quad \text{minimize} \sum_{j=1}^{n} w_j T_j$$

$$T_j \geq (C_j - d_j) \tag{2.5}$$

$$C_j \geq r_j + p_j, \tag{2.6}$$

where $(C_j)_{1 \leq j \leq n}$ is a feasible completion time vector.

3. Lower bounds

In this section we present several lower bound procedures for the problem (P) (noted $P|r_j|\sum T_j$) and the problem (P_w) (noted $P|r_j|\sum w_j T_j$). Note that these

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

procedures can be applied to compute lower bounds on the tardiness of a partial schedule. Let $SP = ([1], [2], \ldots, [k])$ denote a partial schedule of k jobs where $[h]$ denotes the job scheduled in the hth position. Note that $J - \{[1], [2], \ldots, [k]\}$ are the unscheduled jobs and jobs of SP are scheduled one by one according to the earliest available machine rule.

We apply the same rules used for the problem $P||\sum w_j T_j$ (see [20]) in order to take the scheduled jobs into account.

The lower bounds relying on similar techniques are described in the same section.

3.1. Solving assignment problem

3.1.1. The total tardiness case: $LB1$

This lower bound is based on the splitting relaxation (*i.e.*, every job can be performed simultaneously on several machines). The principle is based on an assignment problem involving a lower bound on the completion times of jobs. The lower bound is computed in two steps. In the first step, we compute an earliest completion time $C_{[j]}$ of job scheduled in position j (*i.e.*, for every j, $C_{[j]}$ is a lower bound on the jth smallest completion time in any feasible schedule). In the second step, we define an assignment problem between jobs and the above completion times $C_{[j]}$ that we solve optimally. The lower bound $LB1$ is given by

$$LB1 = \sum_{j=1}^{n} \max(0, C_{[j]} - d_{[j]}),$$

where $(d_{[1]}, d_{[2]}, \ldots, d_{[n]})$ obtained by sorting d_j in the nondecreasing order.

In order to compute the completion times $C_{[j]}$, we use the lower bound introduced by Nessah and Chu [15] for the problem $Pm|r_j|\sum C_j$. This bound is the improved lower bound of Yalaoui and Chu [24] based on the ASRPT rule.

Let $C_j(S^*)$ be the completion time of the job j obtained by the ASRPT rule, $z_j = p_j + r_j$ and $(z_{[1]}, z_{[2]}, \ldots, z_{[n]})$ obtained by sorting z_j in the nondecreasing order.

The completion time used for the assignment problem is given by

$$C_{[j]} = \max(C_{[j]}(S^*), z_{[j]}). \tag{3.1}$$

ASRPT rule: at any time, a job with the SRPT (shortest remaining processing time) among available jobs is simultaneously processed on all the available machines. The processing is interrupted if another job becomes available with a processing time strictly shorter than the remaining processing time of the job in process.

Example. Let us consider an instance of 5 jobs and 2 machines where the data are given in Table 1.

The schedule obtained according to ASRPT rule is shown in Figure 1.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

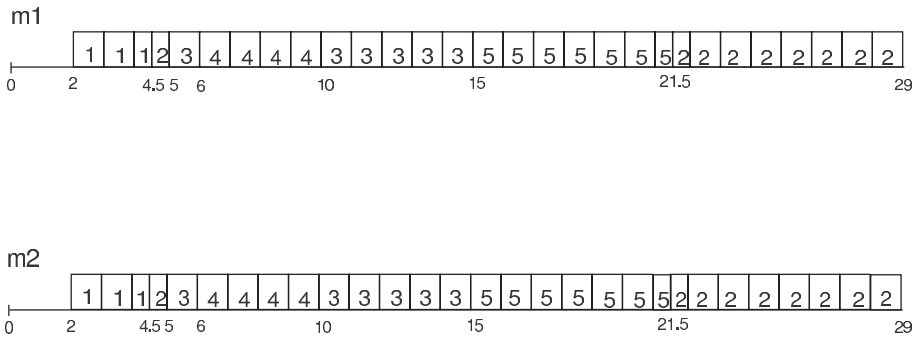


FIGURE 1. Scheduling of jobs obtained by ASRPT rule.

TABLE 1. Instance data

j	1	2	3	4	5
r_j	2	2	5	6	9
p_j	5	16	12	8	13
d_j	7	18	14	8	19
z_j	7	18	17	14	22

TABLE 2. Completion times given by ASRPT rule.

j	1	2	3	4	5
C_j^*	4.5	29	15	10	21.5

From Table 2 and Figure 1, we have the completion times $C_j(S^*)$ given by the ASRPT rule. The completion times used on the lower bound are ($C_{[j]} = \max(C^*_{[j]}, z_{[j]})$): $C_{[1]} = 7$, $C_{[2]} = 14$, $C_{[3]} = 17$, $C_{[4]} = 21.5 \approx 22$, $C_{[5]} = 29$.

By sorting the due dates on the nondecreasing order, we obtain the following lower bound

$$LB1 = \sum_{j=1}^{5} \max(0, C_{[j]} - d_{[j]}) = 23.$$

3.1.2. The total weighted tardiness case: LB2

It is based on an assignment problem involving lower bounds on the completion times of jobs. The lower bound $LB2$ is computed in two steps. In the first step, we compute an earliest completion time $C_{[j]}$ of job scheduled in position j (we use the same completion time given by the ASRPT rule like for the previously lower bound). In the second step, we define an assignment problem between jobs and the above completion times $C_{[i]}$ that we solve by using the algorithm proposed by Baptiste and Le Pape [4]. The cost of assigning job j to date $C_{[i]}$ is $w_j * \max(0, C_{[i]} - d_j)$.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

3.2. Lagrangian relaxation based lower bounds

3.2.1. Lower bound LB3 for $P|r_j|\sum T_j$

The principle of this lower bound is to solve the lagrangian relaxation of the constraints (2.3) and (2.4) in problem (P). From Potts [16], the lagrangian relaxation of the problem (P), yields the problem $(P_{\lambda\gamma})$

$$(P_{\lambda\gamma}) \; : \quad L(\lambda,\gamma) = \min \sum_{j=1}^{n} (T_j + \lambda_j(C_j - d_j - T_j) + \gamma_j(r_j + p_j - C_j))$$
$$\gamma_j \leq \lambda_j \quad \forall \quad j = 1,\ldots,n$$
$$T_j \geq 0 \qquad\quad \forall \quad j = 1,\ldots,n$$

where $\lambda = (\lambda_1,\ldots,\lambda_n)$ and $\gamma = (\gamma_1,\ldots,\gamma_n)$ are the vectors of the corresponding nonnegative multipliers.

We impose to λ_j to satisfy the condition $\lambda_j \leq 1$ and $\gamma_j \leq \lambda_j$ for all j in order to avoid that the Lagrangian function decreases as T_j and C_j increases which gives $L(\lambda,\gamma) = -\infty$.

The problem $(P_{\lambda\gamma})$ is equivalent to

$$\left(P'_{\lambda\gamma}\right) \; : \quad \min \sum_{j=1}^{n} (\lambda_j - \gamma_j)C_j + \sum_{j=1}^{n} (\gamma_j(r_j + p_j) - \lambda_j d_j)$$
$$0 \leq \gamma_j \leq \lambda_j \leq 1 \quad \forall \quad j = 1,\ldots,n.$$

To find the best possible λ and γ, we have to apply the subgradient method. We use the variant proposed by Sherali and Ulular [19] referred to as the *Average Direction Search (ADS)* strategy.

3.2.2. Lower bound LB4 for $P|r_j|\sum w_jT_j$

In this section we extend the lower bound $LB3$ obtained for the unweighted problem $P_m|r_j|\sum T_j$ to the problem $P_m|r_j|\sum w_jT_j$. We can formulate our problem, denoted by (P_w), as follows:

$$(P_w) \quad \text{minimize} \sum_{j=1}^{n} w_jT_j$$
$$T_j \geq (C_j - d_j) \tag{3.2}$$
$$C_j \geq r_j + p_j, \tag{3.3}$$

where $(C_j)_{1\leq j\leq n}$ is a feasible completion time vector.

The lagrangian relaxation of the constraints (3.2) and (3.3) in problem (P_w) yields the problem $(P_{w\lambda\gamma})$

$$(P_{w\lambda\gamma}) \quad \min \sum_{j=1}^{n} (w_j - \lambda_j)T_j + (\lambda_j - \gamma_j)C_j + \gamma_j(r_j + p_j) - \lambda_j d_j$$
$$\gamma_j \leq \lambda_j \leq w_j \quad \forall \quad j = 1,\ldots,n$$

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

where $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\gamma = (\gamma_1, \ldots, \gamma_n)$ are the vectors of the corresponding nonnegative multipliers.

The problem $(P_{w\lambda\gamma})$ is equivalent to

$$\left(P'_{w\lambda\gamma}\right) : \quad \min \sum_{j=1}^{n} (\lambda_j - \gamma_j) C_j + \gamma_j (r_j + p_j) - \lambda_j d_j$$
$$0 \leq \gamma_j \leq \lambda_j \leq w_j \quad \forall j = 1, \ldots, n.$$

To find the best possible λ and γ, we have to apply the subgradient method.

3.2.3. Lower bound LB5 for $P|r_j|\sum w_j T_j$

In this section we describe a lower bound for the problem (P_w) which exploits the lower bounds presented previously. This bound is computed in four steps.

Step 1. We consider the problem of minimizing total weighted tardiness on parallel machines denoted by $P||\sum w_j T_j$ treated in [20]. This problem can be formulated as follows

$$(Q) \quad \text{minimize} \sum_{j=1}^{n} w_j T_j$$
$$T_j \geq (C_j - d_j) \tag{3.4}$$
$$T_j \geq 0,$$

where $(C_j)_{1 \leq j \leq n}$ is a feasible completion time vector.

We recall the lower bound $LB1$ computed for the problem (Q). This lower bounds is obtained by lagrangian relaxation of constraint (3.4), then the problem (Q) is reduced to

$$(PLR) : \quad \min \sum_{j=1}^{n} \lambda_j (C_j - d_j)$$
$$\lambda_j \leq w_j \qquad \forall j.$$

$\lambda = (\lambda_1, \ldots, \lambda_n)$ is the vector of the corresponding nonnegative multipliers. This lower bound is computed by choosing special values of λ_j. These values of lagrangian multipliers λ_j will be used in the next steps.

Step 2. Our goal in this step is to find a lower bound for the problem $P_m|r_j|\sum w_j T_j$. This is formulated as follows

$$(P_w) \quad \text{minimize} \sum_{j=1}^{n} w_j T_j$$
$$T_j \geq (C_j - d_j) \tag{3.5}$$
$$C_j \geq r_j + p_j. \tag{3.6}$$

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

The objective function can be written as follows:

$$\sum_{j=1}^{n} w_j T_j = \sum_{j=1}^{n} \lambda_j T_j + \sum_{j=1}^{n} (w_j - \lambda_j) T_j.$$

Let $(P1_w)$ be the following problem:

$$(P1_w) \quad \text{minimize} \sum_{j=1}^{n} \lambda_j T_j$$
$$T_j \geq (C_j - d_j)$$
$$C_j \geq r_j + p_j,$$

and $(P2_w)$ the following problem

$$(P2_w) \quad \text{minimize} \sum_{j=1}^{n} (w_j - \lambda_j) T_j$$
$$T_j \geq (C_j - d_j)$$
$$C_j \geq r_j + p_j.$$

Let be $lb5_1$ a lower bound for the problem $(P1_w)$ and $lb5_2$ a lower bound for the problem $(P2_w)$. Therefore, the lower bound $LB5$ for the problem (P_w) is $LB5 = lb5_1 + lb5_2$.

In the next steps, we compute the lower bounds $lb5_1$ and $lb5_2$.

Step 3. We consider the problem $(P1_w)$.

As $T_j \geq C_j - d_j$, then $\sum_{j=1}^{n} \lambda_j T_j \geq \sum_{j=1}^{n} \lambda_j (C_j - d_j)$. Therefore, a lower bound for the problem with objective function $\sum_{j=1}^{n} \lambda_j (C_j - d_j)$ is a lower bound for the problem $(P1_w)$. Then, our problem now is

$$(P1_{w_1}) \quad \text{minimize} \sum_{j=1}^{n} \lambda_j (C_j - d_j)$$
$$C_j \geq r_j + p_j. \tag{3.7}$$

We compute two lower bounds for this problem $lb5_{1_1}$ and $lb5'_{1_1}$, then we set $lb5_1 = \max(lb5_{1_1}, lb5'_{1_1})$.

First lower bound $lb5_{1_1}$

The lagrangian relaxation of the constraints (3.7) in problem $(P1_{w_1})$ yields the problem

$$\text{minimize} \sum_{j=1}^{n} \lambda_j (C_j - d_j) + \beta_j (r_j + p_j - C_j)$$
$$\beta_j \leq \lambda_j. \tag{3.8}$$

Therefore, to find the best possible β_j, we apply the Eastman's lower bound and the multiplier adjustment method (see [20]).

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

SECOND LOWER BOUND $lb5'_{1_1}$

This lower bound has the same principle of lower bound $LB2$: solving an assignment problem on the completion time using the Baptiste's algorithm. The cost of assigning job j to date $C_{[i]}$ is $\lambda_j(C_{[i]} - d_j)$. The completion times $C_{[i]}$, are given by Nessah and Chu [15] like in the lower bound $LB1$.

Step 4. In this step, we compute a lower bound $lb5_2$ for the problem $(P2_w)$ by using the same techniques as for $lb5_1$. We compute two lower bounds based on Lagrangian relaxation and assignment problem, then we take the maximum of them.

Finally, the lower bound $LB5 = lb5_1 + lb5_2$.

3.3. SPLITTING BASED LOWER BOUND

The principle of this lower bound is based on the splitting of jobs. The idea is to decompose each job j into p_j pieces. Hence, we transform the original instance problem with n jobs to an instance problem with $\sum_{j=1}^{n} p_j$ pieces. Each piece has a processing time p_j^k, a weight w_j^k,a due date d_j^k and a release date r_j^k $\forall k = 1, \ldots, p_j$ as follows:

$$\begin{aligned} p_j^k &= 1 \\ w_j^k &= \frac{w_j}{p_j} \\ d_j^k &= d_j - (p_j - k) \\ r_j^k &= r_j + k - 1. \end{aligned}$$

Then, we define an assignment problem between pieces and their completion time C_j^k which we solved with the Hungarian algorithm to obtain the lower bound LB_{Split}.

The completion times C_j^k are obtained as follows. We bound the position of each piece (j,k) by a lower bound $\underline{C_j^k}$ and an upper bound for its completion time given by $\overline{C_j^k}$.

$$\underline{C_j^k} \leq C_j^k \leq \overline{C_j^k} \qquad \forall k = 1, \ldots, p_j. \tag{3.9}$$

For each piece, the possible completion times are the values of C_j^k given by (3.9). The weighted tardiness of assignment piece (j,k) to completion time C_j^k is defined as follows:

$$\text{weighted tardiness} = \begin{cases} w_j^k \max(0, C_j^k - d_j^k) & \text{if } \underline{C_j^k} \leq C_j^k \leq \overline{C_j^k} \\ +\infty & \text{otherwise.} \end{cases}$$

The inconvenience of this lower bound is the computational time because the size of the problem is great. Indeed, when using the splitting, we generate a large number of pieces. Therefore, in order to exploit this lower bound in a branch and bound

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

algorithm, we will adapt this scheme to the unweighted problem $P_m|r_j|\sum T_j$. Indeed, the data generation of the unweighted problem is more adapted to this lower bound because when applying the splitting we obtain a set of pieces containing sharply fewer elements than with the data generation for the weighted problem (P_w).

4. Branch-and-bound algorithm

In this section, we suggest a branch-and-bound algorithm for the problems (P) and (P_w) considered in this study. The algorithm starts by computing an initial solution which provides the first upper bound *UB*. In order to obtain *UB*, we apply a heuristic that consists in choosing the best solution from those given by the following priorities rules:

(1) the nondecreasing order of due dates (*EDD*);
(2) the nondecreasing order of $p_i + r_i$ (*ECT*);
(3) the minimum slack order (nondecreasing order of $d_i - p_i$).

Note that the assignment of the jobs is done on the earliest available machine. As the computational tests prove that these heuristics do not usually give a good solution, we have improved this initial upper bound by a classical genetic algorithm which the tests demonstrate its effectiveness.

We have adapted this procedure to the problems (P) and (P_w) to obtain an upper bound.

For branching, every node represents a partial schedule and the depth-first strategy is used. It consists in scheduling a new job after a partial schedule on the earliest machine available. Before the creation of a new node, a lower bound is calculated. If the value of the lower bound obtained is larger than or equal to the value of the upper bound, then this node is removed.

5. Computational results

The objective of the computational experiments described in this section is to evaluate the efficiency of all lower bounds as well as the performance of the branch-and-bound algorithm proposed for each problem treated. Computational experiments were done on a personal computer with an Intel Pentium IV 1.8 GHz processor and 3G RAM, running under Windows XP. The lower bounds and the branch-and-bound algorithm were coded in the *C* language.

5.1. Data generation

The test examples were randomly generated. For the total tardiness problem, the instances have been generated with the scheme of Chu [9] for single machine by adaptation to our problem. For total weighted tardiness we adopt the scheme of Chu [8].

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 3. Problem groups.

β α	0.0	0.5	1.0	1.5
0.05	1	4	7	10
0.25	2	5	8	11
0.5	3	6	9	12

For the experiments, 10 problems for each of all combinations. For each combination the parameters are as follows:

- three levels of the number of machines $m = 2$, $m = 3$ and $m = 4$;
- eight levels for the number of jobs ($n = 10, 15, 20, 25, 30, 40, 50$);
- four levels for $\alpha = 0.0, 0.5, 1.0, 1.5$;
- three levels for $\beta = 0.05, 0.25, 0.5$.

For the total tardiness, we generate, for each job, an integer processing time p_i using a uniform distribution from $[1, 10]$, an integer release date r_i in the interval $[0, \alpha P]$ and an integer due date in the interval $[p_i + r_i, p_i + r_i + \beta P]$, where $P = \sum_{i=1}^{n} p_i$.

For the total weighted tardiness, we generate, for each job, an integer processing time p_i using a uniform distribution from $[1, 100]$, an integer weight w_i using uniform distribution from $[1, 10]$, an integer release date r_i in the interval $[0, \alpha P]$ and an integer due date in the interval $[p_i + r_i, p_i + r_i + \beta P]$, where $P = \sum_{i=1}^{n} p_i$.

We generate 120 instances for each pair of (n, m) divided into 12 representative groups with all the possible combinations of the two parameters α and β as shown in Table 3.

5.2. Results of lower bounds

5.2.1. Lower bounds for $P_m|r_j|\sum T_j$

To evaluate the performance of the lower bounds presented in this paper, we computed the average value obtained from 10 instances for each combination.

From Figures 2–4, we can conclude that the lower bound LB_{split} based on splitting of jobs outperforms the lower bounds $LB1$ and $LB3$. Moreover, by comparing the gap between each lower bound and the upper bound obtained by genetic algorithm in Tables 4–9, we can conclude that LB_{split} has the best gap where the gap is calculated as follows gap $= \frac{UB-LB}{UB} \times 100$. These tables show also the influence of the two parameters α and β on problem hardness. Indeed, we can identify the difficult groups and easy groups to solve next.

Finally, we can conclude that LB_{split} represents a good alternative to build a branch and bound algorithm.

5.2.2. Lower bounds for $P|r_j|\sum w_j T_j$

In this section we present the same analysis given for the problem $P_m|r_j|\sum T_j$. The results for the performance of lower bounds are given in Figures 5–7. We

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 4. Mean gap between lower bound and upper bound $P_m|r_j|\sum T_j$ with $m = 2$.

Jobs	10			15			20			25			30			40			50		
Groups	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}
1	17.24	12.88	1.35	10.07	15.52	0.57	8.23	22.00	0.46	6.18	14.70	0.33	5.47	18.19	0.28	4.32	23.24	0.29	3.57	28.77	0.29
2	40.33	17.16	8.12	26.63	22.78	5.02	31.46	14.98	3.96	27.66	13.18	3.46	28.73	16.28	4.08	29.05	18.98	4.43	31.48	25.33	6.17
3	43.33	40.71	19.05	44.56	29.33	10.59	69.39	35.25	15.79	45.36	44.08	14.80	65.86	57.42	20.93	77.50	55.67	21.71	64.49	56.53	25.09
4	70.40	100.00	36.02	65.51	97.89	30.69	71.60	96.07	31.90	60.11	100.00	25.73	66.03	100.00	26.92	78.11	95.40	38.11	75.38	99.66	33.63
5	50.00	50.00	22.22	46.09	50.00	27.30	30.00	30.00	22.00	10.00	10.00	19.00	0.00	0.00	10.00	20.00	20.00	38.75	20.00	20.00	59.62
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00
7	70.00	80.00	42.08	70.71	80.00	29.10	65.77	70.00	38.14	87.03	96.07	58.19	90.00	90.00	63.00	70.00	70.00	53.37	36.96	40.00	30.14
8	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	1.67	0.00	0.00	0.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	50.00	50.00	22.50	46.67	50.00	43.33	30.00	30.00	21.67	53.00	60.00	43.14	50.00	50.00	32.50	40.00	40.00	26.67	30.00	30.00	30.00
11	0.00	0.00	0.00	0.00	0.00	0.00	20.00	20.00	20.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mean gap	28.44	29.23	12.6	25.85	28.79	12.21	28.04	27.35	12.96	24.94	29	14.21	26.34	28.49	13.97	26.58	26.94	16.11	21.82	25.02	15.4

TABLE 5. Mean gap between lower bound and upper bound for $P_m|r_j|\sum T_j$ with $m = 3$.

Jobs	10			15			20			25			30			40			50		
Groups	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}
1	43.04	14.14	6.27	22.08	9.34	1.59	16.93	16.47	1.07	12.93	15.19	0.98	10.27	18.17	0.71	8.77	21.56	0.68	7.05	26.07	0.74
2	23.96	37.29	12.50	75.70	65.66	23.19	67.25	49.03	14.14	64.85	36.81	12.91	57.12	34.92	12.11	60.24	25.81	10.66	54.53	27.90	11.05
3	10.00	4.00	4.00	0.00	0.00	0.00	10.00	10.00	6.00	10.00	2.58	0.97	10.00	5.71	2.86	20.00	12.86	11.54	20.00	18.00	9.00
4	40.00	40.00	30.00	69.00	70.00	41.95	87.78	90.00	58.97	40.00	40.00	31.67	49.09	50.00	19.14	48.85	60.00	42.73	28.75	50.00	35.71
5	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	10.00	10.00	10.00	30.00	30.00	23.33	20.00	20.00	20.00	0.00	0.00	0.00	20.00	20.00	18.00	10.00	10.00	10.00	10.00	10.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mean gap	11.41	9.61	6.06	16.4	14.58	7.51	17.66	16.29	9.18	10.65	7.88	3.87	12.20	10.73	4.4	12.32	10.85	6.3	10.02	11	4.71

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 6. Mean gap between lower bound and upper bound for $P_m|r_j|\sum T_j$ with $m = 4$.

Jobs	10			15			20			25			30			40			50		
Groups	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}	LB1	LB3	LB_{Split}
1	61.90	18.66	9.73	38.60	14.23	4.24	30.44	13.13	3.18	22.40	15.80	2.19	19.08	15.69	1.70	14.27	19.06	1.23	12.11	20.65	1.01
2	0.00	0.00	0.00	19.55	11.41	5.61	18.57	17.14	7.86	48.95	43.70	19.45	40.00	30.80	11.83	49.00	45.41	21.55	59.22	46.96	22.48
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.00	20.00	16.67	10.00	10.00	4.44
4	30.00	30.00	26.67	10.00	10.00	7.50	10.00	10.00	6.67	10.00	10.00	10.00	10.00	10.00	10.00	20.00	10.00	10.00	10.00	10.00	10.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mean gap	7.65	4.05	3.03	5.67	2.96	1.44	4.91	3.35	1.47	7.61	6.62	3.46	5.75	4.7	1.95	8.6	7.87	4.11	7.6	7.3	3.15

TABLE 7. Mean gap between lower bound and upper bound for $P|r_j|\sum w_jT_j$ with $m = 2$.

Jobs	10			15			20			25			30			40			50		
Groups	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5
1	59.26	82.5	7.27	57.54	88.88	4.96	52.07	87.28	2.32	49.13	87.1	1.53	48.26	85.58	1.71	51.42	87.01	1.07	46.36	85.5	1.19
2	63.85	77.31	44.19	65.14	72.51	27.61	62.55	73.07	16.35	64.58	78.46	22.58	69.17	87.54	13.83	70.11	83.87	15.99	68.29	81.23	18.43
3	37.99	31.45	27.98	78.53	83.03	75.39	67.79	66.3	65.54	50	48.62	50	59.6	55.86	52.96	80	70.27	75.43	79.22	70.63	65.03
4	89.47	100	89.47	90.74	100	91.41	82.26	100	82.26	86.27	99.42	85.14	88.37	100	88.37	89.07	98.81	89.07	89.91	100	90.04
5	78.11	80	78.11	56.55	60	66.55	30	30	30	10	10	10	40	40	40	30	30	30	30	30	30
6	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	0	0	0
7	70	70	70	76.4	80	67.25	68.91	70	68.91	87.09	90	87.09	86.71	90	86.71	88.12	90	88.12	39.07	40	29.07
8	10	10	10	0	0	10	0	0	0	0	0	0	10	10	10	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	30	30	30	68.4	70	59.85	48.22	50	48.22	47.83	50	47.83	87.65	90	87.65	70	70	70	49.88	50	49.88
11	20	20	20	10	10	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean gap	38.22	41.77	31.4	41.94	47.03	35.25	34.31	39.72	26.13	32.91	38.63	25.35	41.64	47.41	32.6	40.72	45	31.64	33.55	38.11	23.63

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 8. Mean gap between lower bound and upper bound for $P|r_j|\sum w_jT_j$ with $m = 3$.

Jobs	10			15			20			25			30			40			50		
Groups	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5
1	73.04	77.22	26.82	61.67	85.49	8.28	60.70	88.82	5.22	59.59	90.77	4.01	55.89	89.56	3.82	55.38	87.58	2.81	53.70	86.13	2.11
2	66.77	67.48	65.45	88.50	93.08	85.63	84.74	93.14	77.74	82.98	76.93	52.83	80.19	72.05	49.28	76.16	78.47	57.85	80.00	74.46	54.07
3	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	10.00	0.00	0.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00	20.00	20.00	20.00
4	60.00	60.00	50.00	70.00	70.00	70.00	80.00	80.00	80.00	79.64	80.00	79.64	50.00	50.00	50.00	70.00	70.00	70.00	50.00	50.00	50.00
5	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	40.00	40.00	42.37	40.00	40.00	40.00	10.00	10.00	10.00	0.00	0.00	0.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	20.00	20.00	20.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mean gap	21.65	22.05	17.88	21.68	24.05	16.99	20.45	23.5	15.25	18.52	20.64	11.37	17.17	19.3	10.25	17.62	20.50	11.72	16.97	19.21	10.51

TABLE 9. Mean gap between lower bound and upper bound for $P|r_j|\sum w_jT_j$ with $m = 4$.

Jobs	10			15			20			25			30			40			50		
Groups	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5	LB2	LB4	LB5
1	85.05	82.56	33.69	72.76	84.44	17.08	68.24	82.58	14.47	62.10	88.53	8.73	66.00	88.68	6.45	57.82	87.79	4.97	56.89	90.28	3.81
2	0.00	0.00	0.00	10.00	10.00	10.00	10.00	8.55	8.76	20.00	20.00	20.00	49.74	44.76	49.74	60.00	55.22	54.80	59.94	53.13	52.05
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	10.00	10.00	10.00	20.00	20.00	20.00	20.00	20.00	20.00	40.00	40.00	40.00	0.00	0.00	0.00	10.00	10.00	10.00	10.00	10.00	10.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mean gap	7.92	7.72	3.64	8.57	9.53	3.92	8.18	9.25	3.6	10.17	12.37	5.72	9.64	11.11	4.68	10.65	12.75	5.82	10.57	12.78	5.49

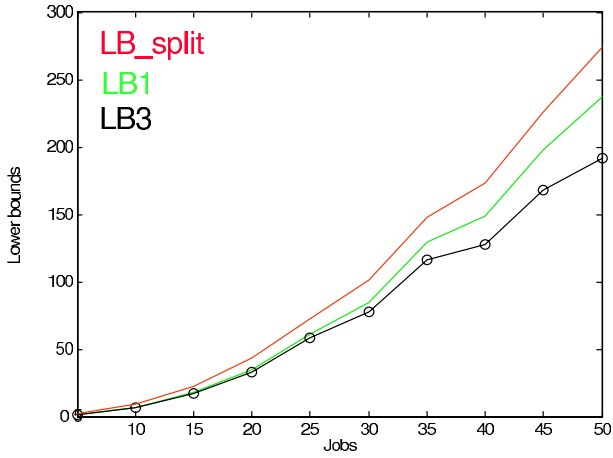


FIGURE 2. Mean values of lower bounds for $P_m|r_j|\sum T_j$ with 2 machines.

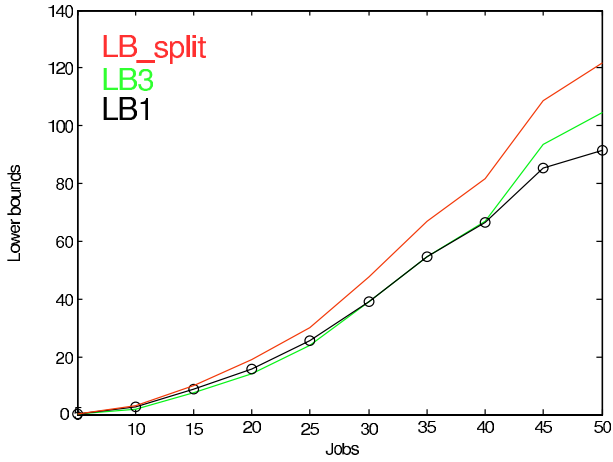


FIGURE 3. Mean values of lower bounds for $P_m|r_j|\sum T_j$ with 3 machines.

can notice that the behaviours of lower bounds $LB2$ and $LB4$ are close to those obtained from the same techniques, for the problem $P_m|r_j|\sum T_j$. Indeed, the techniques used for these lower bounds do not give a good result.

The lower bound $LB5$ dominates largely the other bounds. This remark is confirmed by comparing the gap between each lower bound and upper bound given in Tables 10–15. From these experimental results, we can conclude that lower bound $LB5$ is the best candidate to build a branch-and-bound algorithm in the next step.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

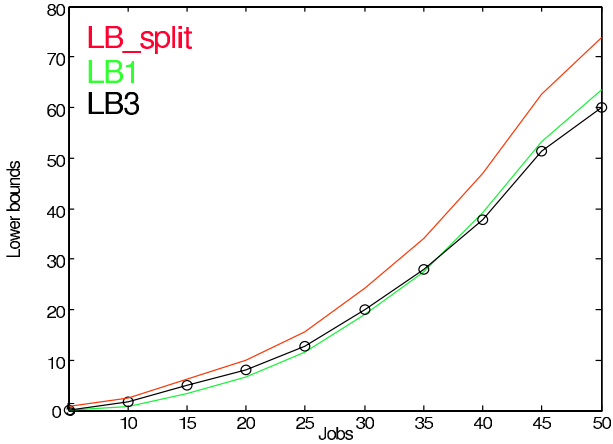


FIGURE 4. Mean values of lowers bounds for $P_m|r_j|\sum T_j$ with 4 machines.

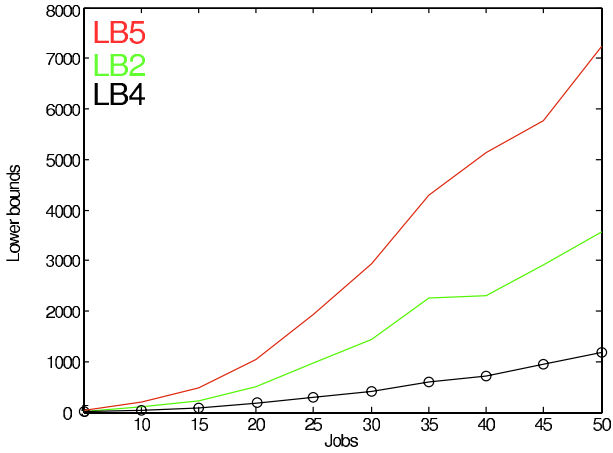


FIGURE 5. Mean values of lower bounds for $P|r_j|\sum w_j T_j$ with 2 machines.

5.3. The performances of the branch-and-bound algorithm

The computational results of the branch-and-bound algorithms for the two problems treated are given in Tables 10–24 where "Node", "Time", "NS" and "Gap" represent respectively, the average of nodes generated, the computation time in seconds the number of problems not solved over the 10 problems of the corresponding class in a period of 30 min and the gap between the value of the lower bound computed at the root and the optimal solution (if can be obtained). A "–" in the columns of "Gap" indicates that all instances are not solved in time and we do not calculate the gap since we do not have the optimal solution.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

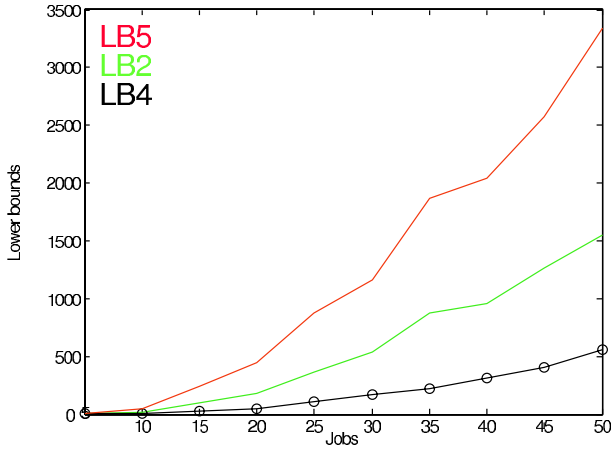


FIGURE 6. Mean values of lower bounds for $P|r_j|\sum w_jT_j$ with 3 machines.

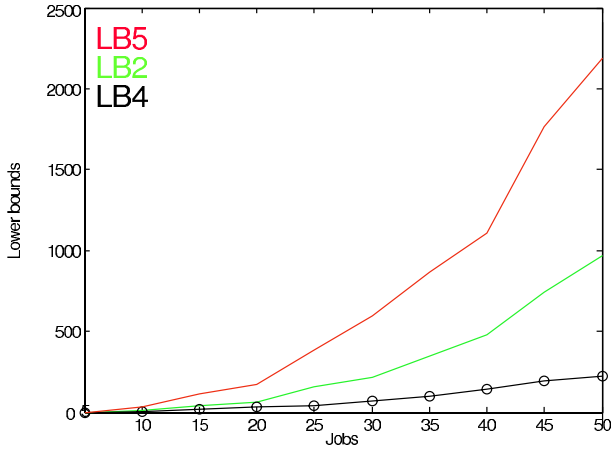


FIGURE 7. Mean values of lower bounds for $P|r_j|\sum w_jT_j$ with 4 machines.

5.3.1. Branch and bound algorithm for $P_m|r_j|\sum T_j$

The branch and bound algorithm (BAB) takes into account the upper bound given by the genetic algorithm and the lower bound LB_{split} presented previously.

Throughout the computational tests, we note that the effectiveness of the BAB algorithm depends on the average tardiness factor α and the relative range of due dates β. Indeed, the problem difficulty is higher for groups 1 and 2 that is corresponding to value of $\alpha = 0.0$. It can be shown especially when the number of jobs is large ($n = 40, 50$). Indeed, when α is small ($\alpha = 0$) the problem becomes without release dates, then all the jobs are available at time 0 and the majority

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 10. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 10$.

$n = 10$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	470.60	6.34	1.35	0	10 720.00	157.57	6.27	0	30 955.70	38.83	9.73	0
2	979.50	13.98	7.90	0	6043.50	87.38	12.50	0	1.00	0.00	0.00	0
3	1772.90	34.26	19.05	0	49.10	0.68	4.00	0	1.00	0.00	0.00	0
4	610.00	15.62	35.61	0	1375.40	48.41	30.00	0	5442.20	89.97	26.67	0
5	149.00	3.13	22.22	0	489.50	15.95	10.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	304.20	12.74	42.08	0	69.90	3.70	10.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	106.10	9.69	22.50	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 11. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 15$.

$n = 15$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	97 633.70	416.67	0.57	2	361 618.10	1445.88	1.08	8	417 860.10	1800.02	–	10
2	389 261.80	1049.79	4.08	3	317 012.30	1460.95	31.75	7	25 539.00	182.09	3.70	1
3	70 943.70	364.92	6.67	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
4	8687.90	67.17	30.24	0	27 622.10	636.82	45.24	3	1769.90	180.00	0.00	1
5	23 464.40	213.01	22.93	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	2389.80	62.32	29.10	0	2334.60	186.21	14.81	1	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	753.50	42.03	43.33	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 12. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 20$.

$n = 20$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	173 817.00	1440.03	0.00	8	218 229.60	1620.02	0.00	9	275 503.20	1800.03	–	10
2	253 412.10	1800.03	–	10	220 751.50	1800.03	–	10	52 820.10	360.01	0.00	2
3	128 489.10	737.40	11.94	4	486.30	1.30	6.00	0	1.00	0.00	0.00	0
4	57 659.90	1011.78	36.41	4	38 746.20	1541.00	28.57	8	6194.20	180.02	0.00	1
5	2839.40	37.82	21.25	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	9672.50	732.61	21.15	3	2320.60	360.13	0.00	2	1.00	0.00	0.00	0
8	2806.80	137.93	1.67	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	1040.50	288.39	16.67	1	355.20	180.04	0.00	1	1.00	0.00	0.00	0
11	1430.70	182.41	11.11	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 13. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 25$.

$n = 25$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	76 171.90	1260.66	0.13	7	161 299.30	1800.09	–	10	212 678.40	1800.07	–	10
2	126 332.00	1800.12	–	10	140 603.40	1800.07	–	10	48 966.00	540.47	14.29	3
3	45 374.90	720.09	8.33	4	42 335.00	180.00	0.00	1	1.00	0.00	0.00	0
4	48 660.60	1744.86	12.00	9	8223.10	517.38	12.50	2	3787.10	180.05	0.00	1
5	3410.50	106.00	7.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	8322.60	1635.37	22.22	9	1.00	0.00	0.00	0	471.20	180.16	0.00	1
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	2764.70	1081.21	0.00	6	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	572.80	180.33	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 14. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 30$.

$n = 30$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	86 684.00	1620.70	0.12	9	98 198.80	1800.16	–	10	120 435.00	1800.11	–	10
2	95 722.70	1800.17	–	10	92 488.60	1800.18	–	10	55 225.90	659.62	4.76	3
3	60 634.00	1261.33	16.67	7	555.10	2.83	2.86	0	1.00	0.00	0.00	0
4	33 965.10	1800.43	–	10	6958.80	720.24	0.00	4	776.90	180.03	0.00	1
5	5717.10	360.03	0.00	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	4755.00	1277.93	33.33	7	695.50	360.25	0.00	2	1.00	0.00	0.00	0
8	508.30	180.03	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	490.90	403.62	15.63	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 15. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 40$.

$n = 40$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	47 660.80	1800.77	–	10	51 791.70	1800.37	–	10	53 340.90	1800.75	–	10
2	51 920.70	1800.53	–	10	51 341.00	1800.64	–	10	20 050.90	540.08	0.00	3
3	45 757.80	1260.25	0.00	7	88.40	1.55	0.00	0	1.00	0.00	0.00	0
4	14 635.20	1801.31	–	10	4716.60	901.85	0.00	5	1133.00	541.60	0.00	3
5	1872.10	379.62	0.00	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	2249.50	1445.45	0.00	8	1.00	0.00	0.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	357.40	730.77	0.00	4	39.20	180.37	0.00	1	1.00	0.00	0.00	0
11	228.00	181.47	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 16. Performance of the BAB for $P|r_j|\sum T_j$ with $n = 50$.

$n = 50$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	26 246.40	1801.39	–	10	29 596.10	1800.81	–	10	30 078.80	1801.49	–	10
2	29 492.30	1801.54	–	10	35 278.00	1800.82	–	10	21 230.70	1080.56	0.00	6
3	22 864.80	1260.57	0.00	7	30 346.80	540.08	0.00	3	1.00	0.00	0.00	0
4	7303.40	1804.14	–	10	1.00	0.00	0.00	0	1.00	0.00	0.00	0
5	2598.10	540.27	0.00	3	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	774.90	935.37	6.67	4	1.00	0.00	0.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	169.90	500.94	12.50	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	49.70	184.18	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 17. Percentage of solved problem for the problem $P|r_j|\sum T_j$.

n	10			15			20			25		
m	2	3	4	2	3	4	2	3	4	2	3	4
Solved problems	100	100	100	96.8	92.4	95.2	88	88	94.8	81.6	90.8	94

n	30			40			50		
m	2	3	4	2	3	4	2	3	4
Solved problems	80.8	89.6	94.4	79.2	89.6	93.6	81.2	90.8	93.6

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 18. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 10$.

$n = 10$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	1069.00	0.02	7.24	0	9279.40	0.20	20.78	0	60 629.80	1.44	33.26	0
2	6763.50	0.11	43.82	0	15 480.90	0.29	0.00	0	1.00	0.00	0.00	0
3	290.30	0.01	27.91	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
4	2068.70	0.03	88.85	0	2332.80	0.05	60.00	0	3049.40	0.08	10.00	0
5	896.40	0.02	78.11	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	636.40	0.01	70.00	0	966.00	0.03	40.00	0	1.00	0.00	0.00	0
8	280.40	0.01	10.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	541.60	0.01	30.00	0	2983.80	0.05	20.00	0	1.00	0.00	0.00	0
11	340.20	0.01	20.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 19. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 15$.

$n = 15$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	9245.20	0.34	3.78	0	19 7861.30	7.65	7.93	0	1 303 250.40	57.31	16.72	0
2	158 589.60	4.43	22.02	0	24 683 289.40	501.55	68.77	2	2 056 065.60	62.39	10.00	0
3	14 269 655.70	238.77	70.76	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
4	23 789.10	0.72	89.28	0	537 125.70	10.69	70.00	0	757 543.50	27.48	20.00	0
5	10 5473.50	1.95	56.55	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	8601.60	0.25	76.35	0	87 909.60	2.77	40.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	9952.20	0.26	68.40	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	20.70	0.00	10.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 20. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 20$.

$n = 20$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	107 741.90	5.75	2.28	0	2 749 833.80	175.58	4.96	0	22 783 601.60	1534.29	7.46	8
2	4 927 664.90	231.74	15.65	0	32 041 345.80	1219.70	63.77	6	3 945 412.60	180.00	0.00	1
3	26 568 730.60	814.20	45.71	4	2 935 313.40	180.00	0.00	1	1.00	0.00	0.00	0
4	1 124 611.70	46.27	80.43	0	7 624 014.40	264.90	77.78	1	11 148 261.30	360.00	0.00	2
5	4 244 826.80	125.50	30.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	31 045.00	2.16	68.37	0	50 697.10	4.31	10.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	69 932.20	2.48	48.22	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 21. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 25$.

$n = 25$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	1 131 337.20	68.22	1.41	0	16 824 218.60	1533.86	3.90	7	21 502 052.30	1800.02	–	10
2	16 141 179.60	1444.06	12.69	7	21 221 369.50	1720.49	13.43	9	3 300 856.20	360.00	0.00	2
3	14 626 604.00	722.68	16.67	4	1.00	0.00	0.00	0	1.00	0.00	0.00	0
4	5 452 748.50	330.26	80.81	1	18 360 401.90	957.91	59.27	5	8 509 888.50	399.85	25.00	2
5	7 399 806.10	180.00	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	831 422.80	44.80	86.98	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	135 239.00	13.15	47.35	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 22. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 30$.

$n = 30$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	10 248 492.30	914.24	1.20	3	13 904 428.40	1659.32	2.00	8	9 904 774.40	1800.02	–	10
2	11 577 740.50	1502.87	5.91	8	12 588 941.70	1800.02	–	10	4 340 458.30	900.01	0.00	5
3	9 634 430.50	1080.01	0.00	6	1 020 291.90	180.00	0.00	1	1.00	0.00	0.00	0
4	16 344 455.50	1338.35	84.62	5	15 891 698.90	900.01	0.00	5	1.00	0.00	0.00	0
5	5 684 548.80	393.95	10.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	20 138 140.80	1120.26	76.17	5	2527.00	0.93	10.00	0	1.00	0.00	0.00	0
8	1521.30	0.39	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	7 706 450.40	542.36	84.56	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 23. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 40$.

$n = 40$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	15 198 104.50	1800.02	–	10	5 987 535.80	1800.02	–	10	4 282 411.20	1800.02	–	10
2	5 900 499.90	1800.02	–	10	6 120 558.20	1800.02	–	10	3 166 532.60	1080.01	0.00	6
3	7 063 749.90	1669.27	100.00	9	383 472.00	180.00	0.00	1	1.00	0.00	0.00	0
4	9 728 259.30	1620.02	91.39	9	7 123 206.30	1068.69	20.00	5	2 007 170.40	180.00	0.00	1
5	5 704 909.70	360.06	0.00	2	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	6 087 544.60	583.73	42.86	3	547 577.80	180.00	0.00	1	1.00	0.00	0.00	0
8	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	9 982 081.20	720.20	0.00	4	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

TABLE 24. Performance of the BAB for $P|r_j|\sum w_jT_j$ $n = 50$.

$n = 50$	$m = 2$				$m = 3$				$m = 4$			
Groups	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS	Nodes	Time	Gap	NS
1	8 588 834.70	1800.02	–	10	2 264 079.30	1800.02	–	10	2 250 650.90	1800.02	–	10
2	3 270 285.90	1800.02	–	10	2 958 550.20	1800.02	–	10	1 555 676.10	1260.02	0.00	7
3	1 953 241.10	1440.02	0.00	8	208 704.80	180.00	0.00	1	1.00	0.00	0.00	0
4	6 118 179.50	1800.02	–	10	5 664 468.00	531.92	12.50	2	1.00	0.00	0.00	0
5	3 845 140.70	900.01	0.00	5	1.00	0.00	0.00	0	1.00	0.00	0.00	0
6	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
7	13 665 269.40	1243.60	47.12	6	272 861.00	180.00	0.00	1	1.00	0.00	0.00	0
8	4759.60	5.58	10.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
9	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
10	457 626.30	185.53	0.00	1	1.00	0.00	0.00	0	1.00	0.00	0.00	0
11	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0
12	1.00	0.00	0.00	0	1.00	0.00	0.00	0	1.00	0.00	0.00	0

of jobs are performed in time which gives a small lower bound or equal to zero in much cases. Therefore, the branch-and-bound algorithm cannot solve the problem in 30 min.

However, for large values of α and β, the problem is very easy to solve and we obtain the optimal solution at the root. We can notice also that when the number of machines increases, the problem difficulty decreases even for the large number of jobs.

The columns "Gap" of Tables 10–16 prove that the effectiveness of the lower bound LB_{split} since the average gap between the lower bound and the optimal solution is equal to zero for many instances.

In summary, the branch-and-bound algorithm presented is efficient to solve 93.6% of problems with 50 jobs and 4 machines (see Tab. 17). Unfortunately, the proposed algorithm could not solve the hardest problems with large size.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

TABLE 25. Percentage of solved problem for the problem $P|r_j|\sum w_jT_j$.

n	10			15			20			25		
m	2	3	4	2	3	4	2	3	4	2	3	4
Solved problems	100	100	100	99.6	99.2	100	98.4	96.8	95.6	94.8	91.6	94.4

n	30			40			50		
m	2	3	4	2	3	4	2	3	4
Solved problems	88.4	90.4	94	81.2	93.2	93.2	80	90.4	93.2

5.3.2. Branch-and-bound algorithm for $P_m|r_j|\sum w_jT_j$

In this section we present the computational results of the branch-and-bound algorithm for the problem $P_m|r_j|\sum w_jT_j$. This algorithm uses the lower bound $LB5$ and the genetic algorithm presented previously.

The tables presented are analogous to Tables given for the problem $P_m|r_j|\sum T_j$.

From Tables 18–24, we notice that the hardness to solve the problem depends on the two parameters α and β like for the problem (P): $P_m|r_j|\sum T_j$. We can observe also that the hard problems to solve are the same as for the problem (P). Indeed, the groups 1 and 2 ($\alpha = 0$ and $\beta = 0.05, 0.25$) are the most difficult to solve and for a great number of jobs, the majority of instances are not solved in 30 min.

However, for the rest of problem groups, the branch-and-bound algorithm can solve the most of instances within a reasonable amount of time and from the gap between the optimal solution (when it is founded) and the lower bound used we can see the effectiveness of $LB5$ since it gives the optimal solution in many cases.

The proposed algorithm can solve 98.4% of instances for problem with 20 jobs and 2 machines and 93.2% of the 250 instances for problem with 50 jobs and 4 machines.

Table 25 gives the percentage of solved problem above the 250 instances generated for each combination of n and m. The results obtained prove the effectiveness of our algorithm.

6. CONCLUSION

In this paper, we studied the identical parallel-machine scheduling problem to minimize the total (weighted) tardiness with release dates. We provided some lower bounds for the $P|r_j|\sum T_j$ and $P|r_j|\sum w_jT_j$ scheduling problems based on different principles. Some of these lower bounds are efficient and allowed us to construct a branch-and-bound method. Such a method incorporates constructive rules and a genetic algorithm. The computational result show that the proposed procedure is efficient and it can solve problems of a large size.

The extension of our lower bounds to shop scheduling problems is an interesting perspective. Moreover, we aim to study other techniques such as the branch-and-price approach.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press

References

[1] B. Alidaee and D. Rosa, Scheduling parallel machines to minimize total weighted and unweighted tardness. *Comput. Oper. Res.* **24** (1979) 75–788.

[2] E.J. Anderson and J.C. Nyirenda, Two new rules to minimize tardiness in a job shop. *Int. J. Prod. Res.* **28** (1990) 2277–2292.

[3] M. Azizoglu and O. Kirka, Tardiness minimization on parallel machines. *Int. J. Prod. Econ.* **55** (1998) 163–168.

[4] P. Baptiste and C. Le Pape, Scheduling a single machine to minimize a regular objective function under setup constraints. *Discrete Optim.* **2** (2005) 83–99.

[5] P. Baptiste, C. Carlier and J. Antoine, A branch, bound procedure to minimze total tardiness on one machine with arbitrary release dates. *Eur. J. Oper. Res.* **158** (2004) 595–608.

[6] P. Baptiste, A. Jouglet and D. Savourey, Lower bounds for parallel machine scheduling problems. *Int. J. Oper. Res.* **3** (2008) 643–664.

[7] S.J. Chen and L. Lin, Reducing total tardiness cost in manufacturing cell scheduling by a multi-factor priority rule. *Int. J. Prod. Res.* **37** (1999) 2939–2956.

[8] C. Chu, A branch and bound algorithm to minimise total flow time with unequal release dates. *Nav. Res. Logist.* **39** (1992) 859–875.

[9] C. Chu, A branch and bound algorithm to minimize total tardiness with release dates. *Nav. Res. Logist.* **39** (1992) 265–283.

[10] S.E. Elmaghraby and S.H. Park, Scheduling jobs on a number of identical machines. *AIIE Trans.* **6** (1974) 1–13.

[11] H. Emmons, One machine sequencing to minimize certain functions of job tardiness. *Oper. Res.* **17** (1969) 701–715.

[12] C. Koulamas, Polynomially solvable total tardiness problem: review and extensions. *Int. J. Manage. Sci.* **25** (1997) 235–239.

[13] Y.H. Lee and M. Pinedo, Scheduling jobs on parallel machines with sequence-dependent times. *Eur. J. Oper. Res.* **100** (1997) 464–474.

[14] G. Mosheiov and D. Oron, A note on the SPT heuristic for solving scheduling problems with generalized due dates. *Comput. Oper. Res.* **31** (2004) 645–655.

[15] R. Nessah and C. Chu, An efficient branch and bound algorithm for the problem $Pm|r_j|\sum C_j$. Submitted manuscript.

[16] C.N. Potts and L.N. Van Wassenhove, A branch and bound algorithm for the total weighted tardiness problem. *Oper. Res.* **33** (1985) 363–377.

[17] A.H.G. Rinnooy Kan, *Machine sequencing problem: classification, complexity and computaion.* Nijhoff, The Hague (1976).

[18] S.-O. Shim and Y.-D. Kim, Scheduling on parallel identical machines to minimize total tardiness. *Eur. J. Oper. Res.* **177** (2007) 135–146.

[19] H.D. Sherali and O. Ulular, Conjugate gradient methods using quasi-Newton updates with inexact line searches. *J. Math. Anal. Appl.* **150** (1990) 359–377.

[20] N. Souayah, I. Kacem, M. Haouari and C. Chu Scheduling on parallel identical machines to minimise the total weighted tardiness. *Int. J. Adv. Oper. Manage.* **1** (2009) 30–69.

[21] L.-H. Su and C.-J. Chen, Minimizing totla tardiness on a single machine with unequal release dates. *Eur. J. Oper. Res.* **186** (2008) 496–503.

[22] Z.J. Tian, C.T. Ng and T.C.E. Cheng, On the single machine total tardiness problem. *Eur. J. Oper. Res.* **165** (2005) 843–846.

[23] F. Yalaoui and C. Chu, Parallel machines scheduling to minimize total tardiness. *Int. J. Prod. Econ.* **76** (2002) 265–279.

[24] F. Yalaoui and C. Chu, New exact method to solve the $Pm|r_j|\sum C_j$ schedule problem. *Int. J. Prod. Econ.* **100** (2005) 168–179.

https://doi.org/10.1051/ro/2012010 Published online by Cambridge University Press