International Journal of Operations Research

International Journal of Operations Research Vol. 4, No. 1, 18−31 (2007)

Lower Bounds for Tardiness Minimization on a Single Machine with Family Setup Times

Imed Kacem*

Institut Charles Delaunay, LOSI, CNRS FRE 2848, Université de Technologie de Troyes, 12 rue Marie Curie, Troyes, France

Received January 2006; *Revised September 2006*; *Accepted October 2006*

Abstract—In this paper, we consider the scheduling of N jobs on a single machine with family setup times in order to minimize the total tardiness. The set of jobs is divided into F families. Between two jobs of the same family, we do not have to stop the machine. However, when switching from one family to another, a setup is required. Each family is characterized by a setup time independent of the sequence. We propose a set of approaches to compute lower bounds for the tardiness criterion. These approaches are analyzed and tested on a large set of numerical experiments in order to identify the dominant lower bounds.

Keywords—Scheduling, Lower bounds, Family setup times, Single machine

1. INTRODUCTION

Grounded in real industrial problems, this paper focuses on scheduling a set of jobs on a single machine which must undergo a setup period when switching processing jobs from a family to another. The aim is to minimize the total tardiness, given that the setup periods are independent of the sequence. This type of problem has been studied in the literature for different objective functions. Given the aim of our study, we provide a brief overview of previous works related to the minimization of total tardiness and/or to scheduling with setup times.

The particular case problem without setup times was intensively studied in the literature. The first remarkable work was presented by Emmons (1969). He proposed some efficient rules which can identify precedence relations between jobs in an optimal sequence. This result was then exploited to construct efficient decomposition approaches (see for example Potts and Van Wassenhove (1982), Della Croce et al. (1998), Chang et al. (1995)). Dynamic programming approaches were also studied by Lawler (1977) and Potts and Van Wassenhove (1987). Du and Leung (1990) studied the complexity of the problem and proposed a proof of its NP-Hardness (in the ordinary sense). Numerous heuristic approaches were proposed. For example, Baker (1999) studied a dynamic priority rule for minimizing tardiness. Koulamas (1997) considered the polynomially solvable tardiness problems and extends these results to the case of identical machines. An efficient branch-and-bound algorithm was also proposed by Szwarc et al. (1999) with efficient decomposition rules. A clever Lagrangian relaxation based-lower bound was described by Potts and Van Wassenhove (1985) for the weighted case. This lower bound was incorporated in an efficient branch-and-bound algorithm. Kondakci et al. (1994) proposed a fast lower bound. This bound is a special case of the one proposed by Chu (1992) for the tardiness problem with unequal release dates. Other extensions of the problem with release dates or/and weighted jobs were also studied (see for example Koulamas and Kyparisis (2001) and Akturk and Ozdemir (2001)). For more details, the reader can consult the state-of-the-art paper of Koulamas (1994).

There are few papers that considered the total tardiness criterion with setup times. The first paper was presented by Ragatz (1993). He proposed a branch and bound algorithm for the single machine problem with sequence-dependent setup times. Rubin and Ragatz (1995) proposed a genetic algorithm to solve the same problem. A comparison of four methods to solve the same problem was presented by Tan et al. (2000). Recently, Souissi et al. (2004) proposed a more efficient branch and bound algorithm and improved the results obtained by Ragatz. The problem considered in these references is more general than the one studied in this paper. However, they did not take the specificity of the family setup times into account. This is one of the reasons that motivate our study.

In our knowledge, there are few works in which the tardiness criterion with family setup times was studied. Schaller proposed branch-and-bound procedures to minimize the total tardiness for the case where the group technology assumption is used and for the case where such an assumption is removed (Schaller, 2006). Nakumara et al. (1978) considered the same problem under the group technology assumption. Baptiste and Jouglet (2001) proposed a pseudopolynomial algorithm to solve the serial batching machine problem. Hariri and Potts (1997) studied the maximum lateness and proposed a heuristic with a worst-case performance ratio of 5/3. They also proposed

* Corresponding author's email: imed.kacem@utt.fr

an effective branch-and-bound algorithm to solve problems with up to 50 jobs.

A large set of works was proposed in the literature to minimize other criteria. In the following paragraph, we provide a short description of some works related to this subject. For more details, we direct the reader to consult the excellent state-of-the-art papers by Allahverdi et al. (1999, 2006), Potts and Kovalyov (2000) and Liaee and Emmons (1977).

A number of works proposed different algorithms to solve single machine scheduling problems with setup times. In particular, the weighted flowtime objective was intensively studied. Ghosh (1994) considered the problem of minimizing the total completion time on a single machine and on identical parallel machines. Ahn and Hyun (1990) proposed an improved dynamic programming approach for multi-class job scheduling. Gupta (1988) studied mean flowtime minimization and proposed a heuristic solution. He showed its effectiveness empirically. Crauwels et al. (1997) proposed a local search method to minimize weighted flowtime. Other heuristics were also examined by Baker (1999) to solve the single machine problem under the lateness criterion. Efficient branch and bound algorithms and lower bounds were proposed by Dunstall et al. (2000) and by Crauwels et al. (1998). Dominance rules were also incorporated in these algorithms. Other extensions of these models and results were also successfully integrated to solve the identical machines problem (see for example the efficient approaches proposed by Azizoglu and Webster (2003) or by Dunstall and Wirth (2005)).

This paper is organized as follows. In Section 2, we formulate the problem and we propose a mixed linear model. In Section 3, we consider the problem without setup times and we present new lower bounds and some others from the literature. In Section 4, we propose two classes of lower bounds for the tardiness minimization with family setup times. The first class is distributive (i.e., the setup is split into pieces and distributed to the jobs of the corresponding family) and in the second class, the setup is not divided and it is considered independently of the processing times. Section 5 provides the description of the numerical experiments and reports the analysis of the results obtained in this work. Finally, we conclude the paper by some conclusions and perspectives.

2. MATHEMATICAL FORMULATION

In the problem studied in this paper, we have to schedule N jobs on a single machine. Each job i has a processing time p_i and a due date d_i. The set of jobs is partitioned into F families. Each job i belongs to a corresponding family $f(i)$. When switching from job i to job j, two cases are possible. If the two jobs belong to the same family (i.e. $f(i) = f(j)$), then, no setup is required between these jobs. In the second case, the families are different and a setup $s_{f(j)}$ is necessary before the execution of job j. The machine can execute only one job at a given time and preemption is not allowed. For a given sequence, a job i is tardy if its completion time C_i is greater than its due date. We aim to find the sequence for which the total tardiness of the set of jobs is minimal.

A mixed linear formulation can be associated to the problem $1\left|s_f\right|\sum T_i$. This formulation can be described in the following model:

Minimize $\sum_{1\le t\le N} T_t$
subject to:

$$C_t \ge C_{t-1} + \sum_{1\le i\le N} x_{i,t} p_i + \sum_{1\le i\le N} \delta_{i,t} s_{f(i)},\ \forall\ 1\le t\le N \quad (1)$$

$$\sum_{1\le i\le N} x_{i,t} = 1,\ \forall\ 1\le t\le N \quad (2)$$

$$\sum_{1\le t\le N} x_{i,t} = 1,\ \forall\ 1\le i\le N \quad (3)$$

$$T_t \ge C_t - \sum_{1\le i\le N} x_{i,t} d_i,\ \forall\ 1\le t\le N \quad (4)$$

$$\delta_{i,t} \ge x_{i,t} - \sum_{j/f(i)=f(j)} x_{j,t-1},\ \forall\ 1\le i\le N,\ \forall\ 2\le t\le N \quad (5)$$

$$\delta_{i,1} = x_{i,1},\ \forall\ 1\le i\le N \quad (6)$$

$$C_0 = 0,\ T_t \ge 0,\ C_t \ge 0,\ \forall\ 1\le t\le N \quad (7)$$

$$\delta_{i,t} \in \{0,1\},\ x_{i,t} \in \{0,1\},\ \forall\ 1\le i\le N,\ \forall\ 1\le t\le N \quad (8)$$

In this model, the objective is to minimize the sum of T_t (the tardiness of the job scheduled in the t^{th} position). $x_{i,t}$ is a binary variable equal to 1 if job i is scheduled in the t^{th} position and equal to 0 otherwise. $\delta_{i,t}$ is a binary variable equal to 1 if a setup of family $f(i)$ is scheduled in the t^{th} position and equal to 0 otherwise. Variable C_t represents the completion time of the job scheduled in the t^{th} position.

From this formulation, we can derive a lower bound by relaxing the binary constraints (8). The relaxed problem can be solved using a linear programming solver (CPLEX for example). The inconvenience of this technique consists in a large number of variables and constraints ($2N^2 + 2N$ variables and $3N^2 + 4N$ constraints). According to preliminary tests, the computational time required to solve the relaxed model increases rapidly as the size of the problems increases. This explains why this lower bound is not compared to the other lower bounds which are described in the remainder of this paper.

3. LOWER BOUNDS FOR THE TARDINESS MINIMIZATION WITHOUT SETUP TIMES

In this section, we present a set of lower bounds for the particular problem without setup times (noted $1\|\sum T_i$). These lower bounds will be used and/or generalized to the original problem with family setup times. We first present some practical properties and we deduce new lower bounds. Other bounds from the literature are proposed too.

Property 1. Let T_{opt} be the total tardiness of an optimal sequence for the problem $1\|\sum T_i$ and let α be a positive number, then the following inequality holds:

$$T_{opt}(p_1, p_2,..., p_N, d_1, d_2,..., d_i-\alpha,..., d_N)$$
$$\leq \alpha + T_{opt}(p_1, p_2,..., p_N, d_1, d_2,..., d_i,..., d_N) \quad (9)$$

Property 2. If jobs i and j verify $p_i \leq p_j$ and $d_i > d_j$ then, interchanging the two due dates between the two jobs does not increase the tardiness of the optimal solution. The proof of this property can easily be established by an interchange argument. Similar result was proposed by Jouglet (2002) for $1|r_i, pre|\sum T_i$.

Theorem 1 (Emmons, 1969). We assume that jobs are indexed in the non-decreasing order of their processing times. C_i^{SPT} is the completion time of the job scheduled in the i^{th} position in the SPT order, then the sequence is optimal if the following condition holds: $d_i \leq C_i^{SPT} + p_{i+1} - p_i \quad \forall i \leq N-1$.

Theorem 2 (Emmons, 1969). We assume that jobs are indexed in the non-decreasing order of their due dates. C_i^{EDD} is the completion time of the job scheduled in the i^{th} position in the EDD order, then the sequence is optimal if the following condition holds: $C_i^{EDD} \leq d_i + p_i \quad \forall i \leq N$.

3.1 Lower bound based on the SPT schedule (lb_1)

Based on Property 1 and Theorem 1, we can derive a new lower bound. For this purpose, we sort jobs in non-decreasing order of processing times. We schedule them in this order. If the completion time of job i does not correspond to the condition of Theorem 1 (i.e. $d_i > C_i^{SPT} + p_{i+1} - p_i$), then we set d_i' to $C_i^{SPT} + p_{i+1} - p_i$ and $\alpha_i = d_i - (C_i^{SPT} + p_{i+1} - p_i)$ in order to impose Emmons condition, otherwise $d_i' = d_i$ and $\alpha_i = 0$. The SPT schedule with the new due dates d_i' is optimal according to the theorem. A lower bound (lb_1) is obtained by computing the total tardiness of this sequence (with the modified due dates) and subtracting $\sum_i \alpha_i$ from the obtained result.

3.2 Lower bound based on the SPT schedule and the interchange of due dates (lb_2)

The principle of this lower bound is the same. We sort jobs in the non-decreasing order of processing times and we schedule them in this order. The difference consists in the fact that if the completion time of job i does not correspond to the condition of Theorem 1, then before setting d_i' to $C_i^{SPT} + p_{i+1} - p_i$, we seek a job j of the subset $\{i+1, i+2, ..., N\}$ such that $d_i > d_j$ and $d_j \leq C_i^{SPT} + p_{i+1} - p_i$. If such a job exists, then we interchange the due dates. If job j does not exist, then the interchange can be done with the smallest due date of the subset $\{i+1, i+2, ..., N\}$ and decreasing minimally the due date to meet Emmons condition. Property 1 is used to take the due date decrease into account to compute the lower bound.

Other versions can be derived from this lower bound. For example, we can shorten the processing times of jobs in the subset $\{1, 2, ..., i-1\}$ before decreasing the due dates. However, this type of modification has a strong bad effect on the obtained lower bound according to some preliminary tests. That is why, it will not be considered in our current study.

3.3 Lower bound based on a linear constrained formulation (lb_3)

This lower bound consists in solving a linear program. The objective function in such a linear program represents the total tardiness expressed in function of the starting times of jobs, their processing times and their due dates. The constraints are obtained on the starting times by associating some fictitious weights to jobs and computing the optimal value of the weighted flowtime associated using the SWPT rule. More explicitly, this lower bound can be described in Theorem 3.

Definition 1. Let $w = (w_1, w_2, ..., w_N)$ be a vector of positive numbers and p be the vector of processing times, i.e., $p = (p_1, p_2, ..., p_N)$. $WF(p, w)$ denotes the minimal weighted flowtime obtained by applying the SWPT rule (proposed by Smith) to the corresponding problem $1| |\sum w_i C_i$.

Property 3. Let $w = (w_1, w_2, ..., w_N)$ be a vector of positive numbers and $(t_i)_{1\leq i\leq N}$ the set starting times of jobs in a feasible schedule. Then, the following inequality holds: $\sum_{1\leq i\leq N} w_i(t_i + p_i) \geq WF(p, w)$.

Theorem 3. Let m be a positive integer and $(w^k)_{k\leq m}$ be a set of m vectors of positive numbers. Then, the solution of the following linear program is a lower bound for the total tardiness minimization:

Minimize $\sum_{1\leq i\leq N} \max\ (t_i + p_i - d_i, 0)$
subject to:

$$\sum_{1\leq i\leq N} w_i^k (t_i + p_i) \geq WF\ (p, w^k) \quad \forall 1 \leq k \leq m \quad (10)$$

Example 1. Let us consider an instance of 3 jobs such that $(p_1, p_2, p_3) = (3,\ 5,\ 8)$ and $(d_1, d_2, d_3) = (3,\ 16,\ 8)$. We take m = 3 and we choose $w^1 = (1,\ 2,\ 3)$, $w^2 = (1,\ 1,\ 1)$ and $w^3 = (1,\ 1,\ 2)$. By applying Theorem 3, we obtain the following linear program:

Minimize $(t_1 + \max(t_2 - 11,\ 0) + t_3)$
subject to:

$t_1 + 2t_2 + 3t_3 \geq 28$ (11)

$t_1 + t_2 + t_3 \geq 11$ (12)

$t_1 + t_2 + 2t_3 \geq 17$ (13)

$t_1,\ t_2,\ t_3 \geq 0$ (14)

The resolution of this linear program gives $lb_3 = 3$ which represents the optimal total tardiness for this example.

Remark 1. This principle can yield an efficient lower bound if the set of weights is judiciously chosen and it can also be easily extended to other problems. The lower bound can be computed easily using a linear programming solver (CPLEX solver for example), however, the number of constraints (equal to *m*) should be minimal in order to reduce the computational effort.

3.4 Szwarc et al.'s lower bound based on the EDD sequence (lb_4)

The principle of this lower bound consists in scheduling jobs according to the earliest due date sequence and in minimally increasing the due dates so that the condition in Theorem 2 will be verified (Szwarc et al., 1999). In others words, if $C_i^{EDD} > d_i + p_i$, then, we can set the due date to $C_i^{EDD} - p_i$.

3.5 Kondakci et al.'s lower bound (lb_5)

This bound is very fast and easy to implement. Initially, it was proposed by Chu (1992) for the problem $1|r_i|\sum_i T_i$ and used in the paper by Kondakci et al. (1994). The jobs are sorted in the non-increasing order of their processing times. C_i^{SPT} denotes the completion time of the job scheduled in the i^{th} position in the *SPT* order and $(d_i')_{1\leq i\leq N}$ represents the series obtained by sorting the due dates in non-decreasing order. The lower bound can be obtained by computing the following sum:

$$lb_5 = \sum_{1\leq i\leq N} \max\ (C_i^{SPT} - d_i',\ 0) \quad (15)$$

Note that this lower bound was improved by Della Croce et al. (1998) by dividing the set of jobs in two subsets before assigning due dates.

3.6 Souissi et al.'s lower bound (lb_6)

This lower bound is very fast and easy to implement too. It can improve the value obtained by the previous lower bound. The jobs are sorted in the non-decreasing order of their processing times. t_i^{SPT} denotes the starting time of the job scheduled in the i^{th} position in the *SPT* order and $(\Delta_i')_{1\leq i\leq N}$ represents the series obtained by sorting the slacks $(\Delta_i)_{1\leq i\leq N}$ of jobs (where $\Delta_i = d_i - p_i$) in non-decreasing order. The lower bound can be obtained by computing the following sum (Souissi et al., 2004):

$$lb_6 = \sum_{1\leq i\leq N} \max\ (t_i^{SPT} - \Delta_i',\ 0) \quad (16)$$

3.7 Potts and Van Wassenhove's lower bound based on a lagrangian relaxation (lb_7)

This bound is based on a clever relaxation in which the weighted tardiness problem can be transformed to a problem of weighted flowtime minimization. To compute this lower bound, we sort jobs in the non-decreasing order of their processing times and we schedule them in the *SPT* order. It is equal to the solution of the following linear problem:

Maximize $\sum_{1\leq i\leq N} \lambda_i\ (C_i^{SPT} - d_i)$
subject to:

$\lambda_i / p_i \geq \lambda_{i+1} / p_{i+1}\ \ \forall\ \ 1 \leq i \leq N-1$ (17)

$0 \leq \lambda_i \leq 1\ \ \forall\ \ 1 \leq i \leq N$ (18)

Potts and Van Wassenhove (1985) proposed a fast procedure to solve this linear problem. This bound is generalized to the problem with family setup times in this paper. The details are provided in the next section.

4. LOWER BOUNDS FOR THE TARDINESS MINIMIZATION WITH FAMILY SETUP TIMES

In this section, we present a set of lower bounds for the tardiness minimization problem with family setup times (denoted $1|s_f|\sum T_i$). The lower bounds presented in the previous section are used and/or generalized to this problem. Two classes of bounds can be distinguished: The distributive class and the non-distributive class.

4.1 Distributive lower bounds

The principle of distributive bounds was widely used by other researchers (Crauwels, 1998; Dunstall and Wirth, 2005; Azizoglu and Webster, 2003) to optimize other criteria. It is based on the transformation of the original problem in a modified problem without setup times. The transformation is established by splitting the setups into pieces and by adding these pieces to the processing times. Then a new instance without setup times is obtained. As a consequence, any lower bound for the problem without setup times can be applied to this instance with the modified processing times. The principle remains valid for any regular objective function (Dunstall et al., 2000).

Based on this principle, we can obtain a relaxation of any instance of the problem $1|s_f|\sum T_i$ in an instance of the problem $1||\sum T_i$ using the following transformation:

$$p_i' = p_i + \sum_{1 \le j \le F} \beta_{i,j} s_j \quad \forall \ 1 \le i \le N \tag{19}$$

such that:

$$\beta_{i,j} = 0 \ \text{ if } f(i) \ne j \ \ \forall \ 1 \le i \le N, \ \forall \ 1 \le j \le F \tag{20}$$

$$\beta_{i,j} \le 1 \ \text{ if } f(i) = j \ \ \forall \ 1 \le i \le N, \ \forall \ 1 \le j \le F \tag{21}$$

$$\sum_{1 \le i \le N} \beta_{i,j} = 1 \ \ \forall \ 1 \le j \le F \tag{22}$$

Therefore, we can apply the different lower bounds of the previous section (lb_1 to lb_7) to the new instance with the modified processing times to compute valid lower bounds for the tardiness minimization with setup times. These obtained bounds are respectively noted LB_1, LB_2, LB_3, LB_4, LB_5, LB_6 and LB_7. Note that LB_5 is one of the lower bounds incorporated in the branch-and-bound algorithm proposed by Schaller (2006).

4.2 Non-distributive lower bounds

Contrary to the distributive class, in the non-distributive lower bounds the setup is not divided and it is considered independently of the processing times. In this subsection, we present two new schemes to obtain valid lower bounds for any instance of the problem $1|s_f|\sum T_i$.

4.2.1 Lower bound based on a linear constrained formulation (*LB*₈)

In this paragraph, we generalize our lower bound lb_3 proposed in Section 3 and based on a linear constrained formulation. For this purpose, we will exploit the following idea that was shown by Mason and Anderson (1991) and used by Azizoglu and Webster (2003) for flowtime minimization. The idea consists in the fact that a problem with setup times can be separated into two independent problems. The first one is the scheduling of jobs. The second one is the scheduling of setups.

Let us consider a feasible schedule σ for the problem with setup times. Let $h_1, h_2, ..., h_F$ be the indexes of the family setups in the order of their appearance in schedule σ. It is obvious that these setups induce a time lag for each job belonging to family h_k ($1 \le k \le F$) at least equal to $\sum_{r=1}^{k} s_{h_r}$. Thereafter, the total time lag for all the jobs of family h_k is at least equal to $n_{h_k} \sum_{r=1}^{k} s_{h_r}$ where n_{h_k} is the cardinal of family h_k. In conclusion, the total time lag due to the setups is at least equal to $\tau(\sigma) = \sum_{1 \le k \le F} n_{h_k} \sum_{r=1}^{k} s_{h_r}$. Obtaining a lower bound on the time lag induced by the setups can be reduced to an instance of the problem $1||\sum w_i C_i$ by taking $s_1, s_2, ..., s_F$ as jobs to schedule and $n_1, n_2, ..., n_F$ as their corresponding weights. The result can be obtained by applying the SWPT rule. We note τ^* this lower bound on the time lag.

Based on this result, we can now generalize Theorem 3 to the case with family setup times. This result can be described in Theorem 4.

Theorem 4. Let m be a positive integer and $(w^k)_{k \le m}$ be a set of m vectors of positive numbers. Then, the solution of the following linear program is a lower bound for the total tardiness minimization with family setup times:

Minimize $\sum_{1 \le i \le N} \max\,(t_i + p_i + \tau_i - d_i,\ 0)$
subject to:

$$\sum_{1 \le i \le N} \tau_i \ge \tau^* \tag{23}$$

$$\tau_i \ge s_{f(i)} \ \ \forall \ 1 \le i \le N \tag{24}$$

$$\sum_{1 \le i \le N} w_i^k (t_i + p_i) \ge WF(p, w^k) \ \ \forall \ 1 \le k \le m \tag{25}$$

Remark 1 remains valid to this lower bound and the major difficulty to implement it is the determination of the fictitious weights.

4.2.2 Lower bound based on a Lagrangian relaxation (*LB*₉)

In this paragraph, we extend the excellent lower bound proposed by Potts and Van Wassenhove for the weighted tardiness minimization without setups. In this generalization, we integrate the setup times as dummy jobs. Indeed, we relax the original problem such that only one setup can be scheduled for each family. Therefore, the relaxed problem can be viewed as a problem of total tardiness minimization with chain precedence constraint.

This type of relaxation was exploited to build efficient lower bounds for the problem $1|s_f|\sum w_i C_i$ by Dunstall et al. (2000).

In the relaxed problem, noted (π), we remove the set of setups and we replace it by a set of dummy jobs $\{N+1, N+2, ..., N+F\}$ such that $\{s_1, s_2, ..., s_F\}$ is the set of respective processing times. Each job $(N+f)$ must be scheduled before jobs belonging to family f. The objective is to schedule jobs (initial jobs and dummy jobs) by respecting the precedence constraints to minimize the total tardiness of initial jobs (i.e. jobs belonging to $\{1, 2, ..., N\}$). This problem can be formulated, except the capacity constraint of the machine, as follows:

(π): Minimize $\sum_{1 \le i \le N} T_i$
subject to:

$$T_i \ge 0 \quad \forall \ 1 \le i \le N \tag{26}$$

$$T_i \ge C_i - d_i \quad \forall \ 1 \le i \le N \tag{27}$$

$$C_i \ge C_{N+f(i)} + p_i \quad \forall \ 1 \le i \le N \tag{28}$$

where C_i is the completion time of job i ($1 \le i \le N + F$).

In the generalized lower bound, we apply a Lagrangian relaxation which allows us to transform problem (π) to an instance of the problem $1||\sum w_i C_i$ (polynomially solvable by applying the SWPT rule). Such an instance can be described as follows:

(π'): Minimize $\sum_{1 \le i \le N+F} \mu_i C_i + L(\mu)$
subject to:

$$\mu_i = \begin{cases} \lambda_i^1 - \lambda_i^2 & \text{if } 1 \le i \le N \\ \sum_{j/f(j)=i-N} \lambda_j^2 & \text{if } N+1 \le i \le N+F \end{cases} \tag{29}$$

$$L(\mu) = \sum_{1 \le i \le N} (\lambda_i^2 p_i - \lambda_i^1 d_i) \tag{30}$$

$$\lambda_i^1 - \lambda_i^2 \ge 0 \quad \forall \;\; 1 \le i \le N \tag{31}$$

$$\lambda_i^1 \le 1 \quad \forall \;\; 1 \le i \le N \tag{32}$$

The completion times must respect the constraint of the machine capacity. According to the principle of Lagrangian relaxation, for any nonnegative multipliers, the optimal solution of problem (π') yields a valid lower bound for the total tardiness minimization with family setup times. The inconvenience of this technique is the difficulty to find the best multipliers maximizing the lower bound. To overcome this problem, Potts and Van Wassenhove proposed a clever approach in which they used a multiplier adjustment method. This method needs a heuristic method to schedule the jobs. The decision variables of the resulting problem are then reduced to the Lagrangian multipliers and the heuristic solution must be optimal for the chosen multipliers. Based on this result, our generalized lower bound can be described in the following Theorem.

Theorem 5. We assume that the $(N + F)$ jobs are sorted in non-decreasing order of their processing times. Let C_i^{SPT} be the completion time of the job scheduled in the i^{th} position according to the *SPT* order and $(\mu'_i)_{1 \le i \le N+F}$ be the series of the multipliers in the same order respecting the previous constraints (i.e. $\lambda_i^1 - \lambda_i^2 \ge 0, \; \forall \;\; 1 \le i \le N$ and $\lambda_i^1 \le 1, \; \forall \;\; 1 \le i \le N$). The solution of the following linear problem is a valid lower bound for the tardiness minimization with setup times:

Maximize $\sum_{1 \le i \le N+F} \mu'_i C_i^{SPT} + L(\mu)$
subject to:

$$\mu'_i / p_i \ge \mu'_{i+1} / p_{i+1}, \;\; \forall \;\; 1 \le i \le N + F - 1 \tag{33}$$

Remark 2. The principle of this lower bound remains valid for any heuristic. We have just to sort variables according to the jobs order given by the heuristic used to compute the lower bound.

5. NUMERICAL EXPERIMENTS

In this section, we provide the computational results used to evaluate the performance of the different methods presented above. The tests were carried out on a Pentium 4 PC in the Windows XP environment using the C language and CPLEX software to solve the linear programs. The following paragraphs describe our data generation methods, the results obtained and our analysis of these experiments.

5.1 Data generation

The experiments were carried out on six series. We tested instances for which the size is variable ($N \in \{15, 30, 50, 70, 90\}$). In each series, the instances were randomly generated according to a uniform distribution, such that $p_i \in [1, 100]$. For each series, we considered three levels for setup times: small ($s_f \in [0, 10]$), medium ($s_f \in [0, 50]$) and large ($s_f \in [0, 100]$). Jobs were assigned uniformly to the families. The due date were also randomly generated using a uniform distribution over $D(1 - r - T/2)$ and $D(1 - r + T/2)$, where:

- $D = \sum_{1 \le i \le N} p_i + \sum_{1 \le f \le F} s_f$
- T is the due date range and r is the tardiness factor.

The six series were generated as follows:

- 1st series: $r = 0.5$ and $T = 1.0$.
- 2nd series: $r = 0.6$ and $T = 0.8$.
- 3rd series: $r = 0.5$ and $T = 0.6$.
- 4th series: $r = 0.5$ and $T = 0.8$.
- 5th series: $r = 0.3$ and $T = 0.6$.
- 6th series: $r = 0.8$ and $T = 0.4$.

Variables F and N were chosen according to some groups of couples (N, F). In each group, we duplicated randomly 30 instances and computed the mean values of the lower bounds studied in this paper.

5.2 Results and analysis

The different results are summarized in Tables 1-7 and Figure 1. For each lower bound, the mean values obtained for each group are given (see Tables 1-6). The mean value of computation times obtained for all the tests for each lower bound is also given in Table 7 and Figure 1. From these results, we can make the following remarks:

a) Except the series 3, in which the level for setup times affects the behaviour of the dominant lower bounds, we can remark throughout the computational experiments that the performances of the different lower bounds depend mainly on the due date range and the tardiness factor.

b) Distributive lower bounds: The main advantages of these procedures are the computational effectiveness and the simplicity of their implementation (except LB_3 which needs to solve a linear program). We have just to apply simple and fast algorithms to the modified data (i.e. the modified processing times after including setup times). This approach leads generally to an interesting computation time (10^{-4} s for some lower bounds). For some generation data, all the lower bounds have relatively the same performance (series 6, for example). In this case, the choice of a distributive bound may be suitable since the computational effort is reduced. As an example, for series 6, the different lower bounds have almost the same performance (see Table 6). In this case, the choice of one of the fast lower bounds (LB_7) represents a good alternative to build an efficient branch-and-bound algorithm. However, the difficulty to obtain efficient distributive lower bounds consists in finding the best way of the distribution of setup times to the processing times. In this study, we have chosen to include uniformly a setup to the jobs of the corresponding family. This distribution scheme seems to be satisfactory. The paper by Schaller (2006) confirmed the same conclusion.

c) Lower bounds LB_1 and LB_2: These two lower bounds have almost the same performance and the same behaviour. The major advantage of these bounds is they can be obtained very quickly. We can remark that these bounds are relatively effective if the values and the dispersion of the due dates are small. This is the case of series 6 (see Table 6). The dispersion of the due dates has a bad effect on the performance of the two lower bounds. This is the case of the other series (see Tables 1-5). This behaviour can be explained by the fact that LB_1 and LB_2 are based on the SPT sequence and Theorem 1. Indeed, if the values and the dispersion of due dates are small, the conditions of Theorem 1 can be satisfied by the SPT sequence and the modification of these due dates (needed to compute these lower bounds) has not an important effect.

Table 1. Mean values for the lower bounds obtained for series 1

Group	(N, F)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
Small level for setup times										
1	N=15, F=2	98.0	95.3	356.5	367.7	173.0	123.7	376.5	356.7	**397.3**
2	N=15, F=3	42.6	42.0	401.6	370.9	150.7	100.0	422.9	388.4	**450.5**
3	N=15, F=4	61.0	60.7	365.0	357.9	177.7	123.8	380.0	339.2	**414.0**
4	N=30, F=2	0.0	0.0	892.7	781.7	111.2	45.5	913.5	895.4	**955.5**
5	N=30, F=5	0.0	0.0	742.3	723.7	111.8	53.0	746.3	693.1	**819.9**
6	N=30, F=10	0.0	0.0	842.1	729.2	109.0	53.9	842.9	698.3	**954.6**
7	N=50, F=2	0.0	0.0	1280.3	1264.4	133.7	67.0	1283.4	1282.5	**1348.4**
8	N=50, F=5	0.0	0.0	1618.4	1146.7	141.1	68.9	1634.0	1541.4	**1739.9**
9	N=50, F=10	0.0	0.0	1697.4	1314.2	137.7	67.5	1714.6	1462.5	**1957.9**
10	N=70, F=5	0.0	0.0	2644.3	1848.5	110.6	54.4	2681.5	2512.3	**2897.2**
11	N=70, F=10	0.0	0.0	4437.3	2392.1	152.5	56.4	4502.9	3962.3	**4991.8**
12	N=70, F=25	0.0	0.0	2756.0	1775.1	124.6	47.1	2807.5	1827.0	**3347.5**
13	N=90, F=5	0.0	0.0	4542.0	2496.9	172.7	96.2	4605.3	4358.6	**4881.9**
14	N=90, F=10	0.0	0.0	4566.8	2645.8	90.6	40.3	4669.7	4123.2	**5207.2**
15	N=90, F=25	0.0	0.0	4603.7	2323.0	116.3	46.4	4685.2	3545.0	**5464.9**
Medium level for setup times										
1	N=15, F=2	108.0	105.4	380.0	388.7	189.1	142.6	400.8	386.3	**521.1**
2	N=15, F=3	48.4	47.7	448.2	406.2	175.5	129.1	468.1	380.5	**643.7**
3	N=15, F=4	98.2	97.8	425.1	405.4	223.3	176.3	438.2	302.7	**652.4**
4	N=30, F=2	0.0	0.0	917.0	801.2	117.5	51.9	936.8	936.3	**1187.6**
5	N=30, F=5	0.0	0.0	789.0	776.2	124.8	69.8	797.0	535.5	**1344.8**
6	N=30, F=10	0.0	0.0	964.8	825.6	137.1	90.7	970.8	325.1	**1719.7**
7	N=50, F=2	0.0	0.0	1308.2	1288.5	138.7	71.2	1310.5	1328.9	**1765.0**
8	N=50, F=5	0.0	0.0	1696.6	1197.6	158.8	86.7	1718.5	1312.9	**2431.4**
9	N=50, F=10	0.0	0.0	1827.0	1420.2	160.3	90.7	1846.2	676.3	**3561.3**
10	N=70, F=5	0.0	0.0	2692.7	1897.3	116.8	60.6	2730.1	1939.7	**4099.1**
11	N=70, F=10	0.0	0.0	4629.1	2521.9	173.5	74.4	4726.2	2260.6	**7716.5**
12	N=70, F=25	0.0	0.0	3289.5	2087.7	155.6	75.9	3429.4	350.2	**7433.4**
13	N=90, F=5	0.0	0.0	4654.4	2557.3	184.0	105.1	4723.2	3628.9	**6505.1**
14	N=90, F=10	0.0	0.0	4736.1	2750.6	99.5	48.0	4869.8	2484.5	**8400.0**
15	N=90, F=25	0.0	0.0	5056.9	2560.2	148.7	73.5	5201.2	887.1	**10652.5**
Large level for setup times										
1	N=15, F=2	124.1	121.7	410.4	414.6	210.4	167.7	431.8	440.3	**698.4**
2	N=15, F=3	65.8	64.8	508.3	452.0	211.4	171.0	528.1	403.5	**902.4**
3	N=15, F=4	148.3	147.3	497.0	461.6	282.7	241.9	511.4	302.2	**949.4**
4	N=30, F=2	0.0	0.0	947.7	825.7	125.4	61.0	967.5	994.3	**1500.8**
5	N=30, F=5	3.2	3.0	848.2	842.9	142.5	91.3	860.4	443.9	**2050.5**
6	N=30, F=10	0.0	0.0	1109.6	943.9	176.7	143.0	1128.4	154.8	**2644.6**
7	N=50, F=2	0.0	0.0	1342.7	1317.3	144.7	77.3	1343.7	1397.9	**2387.2**
8	N=50, F=5	0.0	0.0	1789.7	1262.9	184.1	111.1	1821.5	1085.3	**3495.6**
9	N=50, F=10	0.0	0.0	1983.4	1552.0	189.6	119.7	2006.0	272.8	**5741.7**
10	N=70, F=5	0.0	0.0	2747.4	1956.6	124.7	69.2	2787.6	1390.6	**5755.4**
11	N=70, F=10	0.0	0.0	4827.5	2678.7	199.7	98.9	4973.2	1090.4	**11127.0**
12	N=70, F=25	0.0	0.0	3899.1	2475.5	191.3	110.5	4169.7	97.8	**12431.2**
13	N=90, F=5	0.0	0.0	4784.6	2630.0	198.8	118.0	4866.7	2995.6	**8746.4**
14	N=90, F=10	0.0	0.0	4929.4	2879.0	110.3	56.5	5108.7	1312.9	**12443.5**
15	N=90, F=25	0.0	0.0	5526.5	2859.6	183.1	105.0	5771.3	89.0	**18127.1**

d) Lower bounds LB_3 and LB_8: These lower bounds have relatively the same performance, behaviour and computational cost. Except series 5, they are robust according to the dispersion of data and have relatively a uniform performance (see the different results). To obtain an efficient performance for these lower bounds, we have to find a pertinent set of constraints by judiciously choosing the fictitious weights. In this work, we have limited the number of constraints to $m = N + 3$. The fictitious weights were chosen such that each job can occupy all the positions in the SWPT schedules associated to the fictitious weights (which needs N constraints). The three other constraints allow us to obtain the sequences SPT, EDD and MDD. Our choice is motivated by the need to obtain fast procedures and complementary constraints. Obviously, if we introduce some additional constraints, the performance may be enhanced, but, the computational effort will be more important too.

e) Lower bound LB_4: This lower bound is robust according to the data dispersion. Its major advantage is the computational efficiency (10^{-3} s almost). This lower bound is particularly effective if the values of the due dates are large. This is the case of all the instances in series 5. High values of due dates have a good effect on the performance of this lower bound. However, if the due dates are small, it is very difficult to meet the conditions of Theorem 2 and the relaxation needed to compute the lower bound

Table 2. Mean values for the lower bounds obtained for series 2

Group	(N, F)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
Small level for setup times										
1	N=15, F=2	321.3	318.9	910.4	578.7	633.2	509.8	966.6	908.9	**993.4**
2	N=15, F=3	222.1	224.6	875.9	556.2	559.5	420.9	930.2	853.5	**967.6**
3	N=15, F=4	278.1	278.1	914.5	593.8	632.2	508.0	974.8	873.4	**1019.5**
4	N=30, F=2	261.9	264.1	3057.0	1292.4	1562.3	1279.6	3243.6	3054.2	**3297.7**
5	N=30, F=5	280.3	281.8	2861.1	1273.6	1464.8	1211.5	3027.7	2754.8	**3149.9**
6	N=30, F=10	196.7	198.1	2892.6	1270.6	1452.6	1204.8	3075.2	2624.7	**3246.0**
7	N=50, F=2	189.7	190.0	7277.5	2250.0	3489.5	3046.4	7655.3	7268.5	**7763.7**
8	N=50, F=5	227.8	227.9	7039.0	2134.9	3416.2	3005.7	7383.5	6869.6	**7581.0**
9	N=50, F=10	110.9	111.2	7711.7	2247.6	3653.9	3205.7	8082.4	7255.1	**8463.1**
10	N=70, F=5	4.7	4.7	13935.6	3178.6	5895.9	5293.6	14672.7	13701.0	**14979.8**
11	N=70, F=10	372.4	372.7	16588.5	3395.4	7068.6	6414.7	17502.3	15967.1	**18085.1**
12	N=70, F=25	0.0	0.0	14196.4	3266.3	6141.2	5527.2	15135.0	12399.3	**16018.8**
13	N=90, F=5	292.5	292.6	23524.3	4136.7	9873.6	9059.8	24697.8	23226.3	**25097.6**
14	N=90, F=10	20.9	21.0	23570.4	4205.2	9573.2	8805.1	25066.9	22831.3	**25775.4**
15	N=90, F=25	0.0	0.0	22686.3	4148.6	9482.3	8716.3	24017.7	20414.4	**25343.8**
Medium level for setup times										
1	N=15, F=2	365.1	362.9	975.1	614.2	686.4	567.5	1034.1	970.8	**1189.8**
2	N=15, F=3	312.4	313.3	983.5	614.0	648.9	518.8	1045.7	862.0	**1270.6**
3	N=15, F=4	399.1	400.0	1058.1	673.4	760.3	647.7	1128.8	836.1	**1392.6**
4	N=30, F=2	318.3	320.3	3158.1	1326.2	1634.7	1351.4	3354.0	3146.7	**3688.9**
5	N=30, F=5	356.1	357.5	3109.9	1371.9	1648.4	1399.1	3316.0	2503.9	**4051.9**
6	N=30, F=10	444.5	446.9	3382.6	1451.0	1805.8	1570.2	3628.0	1887.4	**4665.9**
7	N=50, F=2	222.3	223.0	7461.0	2291.2	3600.5	3149.4	7853.2	7410.7	**8527.6**
8	N=50, F=5	336.1	336.6	7421.4	2220.4	3678.0	3265.0	7798.3	6446.8	**9029.0**
9	N=50, F=10	288.1	289.7	8474.8	2436.1	4195.0	3731.7	8973.1	5964.5	**11271.6**
10	N=70, F=5	29.5	29.4	14429.8	3269.2	6205.4	5585.0	15220.1	13071.2	**17079.2**
11	N=70, F=10	595.0	595.5	17681.9	3586.6	7813.6	7145.1	18776.7	14121.8	**22249.9**
12	N=70, F=25	677.9	680.1	16751.8	3756.1	7839.8	7172.6	18343.2	6976.7	**23217.1**
13	N=90, F=5	355.5	355.5	24192.7	4227.5	10299.1	9471.9	25463.0	22434.8	**27911.5**
14	N=90, F=10	98.0	98.0	24759.4	4380.5	10339.6	9546.2	26541.5	20500.0	**30811.4**
15	N=90, F=25	59.2	59.3	25749.9	4625.7	11440.6	10626.9	27776.5	13136.8	**35860.5**
Large level for setup times										
1	N=15, F=2	421.5	419.0	1058.0	658.7	755.5	642.4	1118.5	1059.9	**1441.9**
2	N=15, F=3	427.1	426.3	1119.7	684.8	766.2	645.6	1194.0	919.2	**1638.8**
3	N=15, F=4	557.6	553.5	1226.2	771.3	921.5	818.8	1317.3	863.4	**1803.7**
4	N=30, F=2	393.6	395.1	3283.3	1368.9	1727.2	1446.7	3491.2	3270.2	**4152.0**
5	N=30, F=5	457.2	457.7	3420.4	1494.9	1884.1	1640.1	3671.4	2259.3	**5101.4**
6	N=30, F=10	849.8	848.2	3962.3	1667.9	2291.3	2062.7	4300.0	1318.4	**6139.2**
7	N=50, F=2	273.8	275.0	7690.1	2342.8	3740.4	3283.9	8100.7	7605.8	**9460.5**
8	N=50, F=5	471.9	471.5	7877.4	2327.8	4015.4	3593.2	8320.9	5968.1	**10780.5**
9	N=50, F=10	628.3	631.5	9380.9	2676.8	4879.6	4407.2	10060.0	4633.3	**14422.9**
10	N=70, F=5	57.7	57.9	15010.5	3382.0	6595.5	5956.9	15903.5	12331.0	**19674.1**
11	N=70, F=10	1054.7	1056.8	18923.7	3826.9	8746.2	8069.8	20325.8	12029.1	**26931.3**
12	N=70, F=25	1912.0	1914.5	19631.0	4359.9	9878.5	9154.0	22115.4	2718.1	**30246.1**
13	N=90, F=5	431.3	431.6	24988.7	4339.7	10838.4	10001.3	26416.3	21498.9	**31368.7**
14	N=90, F=10	272.8	272.9	26093.5	4598.6	11307.1	10482.8	28343.2	17779.1	**36631.4**
15	N=90, F=25	501.7	502.5	29216.3	5218.4	13875.8	12998.1	32287.4	6523.4	**46955.9**

becomes important. This is the case of the instances of series 6. This behaviour can be explained by the fact that LB_4 is based on the EDD sequence and Theorem 2. Indeed, if the values of the due dates are small, the conditions of Theorem 2 are so difficult to be satisfied by the EDD sequence and the modification of these due dates needed to compute the lower bound has an important bad effect.

f) Lower bounds LB_5 and LB_6: The major advantage of these bounds is the computational efficiency. We can remark that these bounds are efficient in the case of series 3 (see Table 3). The dispersion of the due dates has a bad effect on the performance of the two lower bounds. This is the case of series 1 and 2 (see Tables 1 and 2). This behaviour can be explained by the fact that these lower bounds are based on the assignment of the due dates to the *SPT*-sequence completion times. Indeed, if the values and the dispersion of the due dates are small, the relaxation needed to compute these lower bounds has not an important effect and their performance may be satisfactory.

g) Lower bounds LB_7 and LB_9: Except series 5, they yield generally the best performances when comparing with the other lower bounds. In addition, they are robust and the dispersion of the due dates does not roughly affect their values. The performances of these lower bounds are comparable (see Tables 1-6). The complexity of the bound (LB_7) is very interesting using the procedure of Potts and

Table 3. Mean values for the lower bounds obtained for series 3

Group	(N, F)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
Small level for setup times										
1	N=15, F=2	25.5	24.2	310.0	391.1	**435.0**	306.2	319.7	308.5	334.8
2	N=15, F=3	11.5	11.5	337.1	388.8	**397.7**	253.6	348.2	320.3	368.7
3	N=15, F=4	0.0	0.0	321.3	383.6	**436.2**	310.1	330.7	291.1	355.0
4	N=30, F=2	0.0	0.0	698.2	777.8	**1155.6**	902.7	715.0	697.8	742.6
5	N=30, F=5	0.0	0.0	598.7	733.4	**1106.0**	869.8	604.6	530.7	663.4
6	N=30, F=10	0.0	0.0	650.5	771.7	**1095.1**	870.8	660.6	486.9	745.5
7	N=50, F=2	0.0	0.0	1014.1	1238.1	**2680.3**	2313.7	1018.6	1012.5	1069.3
8	N=50, F=5	0.0	0.0	1175.9	1247.9	**2638.5**	2267.0	1187.1	1090.4	1261.6
9	N=50, F=10	0.0	0.0	1267.3	1287.2	**2809.1**	2423.5	1278.2	992.2	1446.6
10	N=70, F=5	0.0	0.0	1957.5	1768.0	**4667.0**	4175.2	1974.0	1815.6	2111.7
11	N=70, F=10	0.0	0.0	3121.0	1965.0	**5279.0**	4733.8	3147.7	2640.7	3445.4
12	N=70, F=25	0.0	0.0	1940.0	1805.0	**4744.6**	4223.9	1970.2	1015.5	2323.5
13	N=90, F=5	0.0	0.0	3192.4	2328.9	**7676.3**	7004.1	3219.3	3009.4	3393.8
14	N=90, F=10	0.0	0.0	3222.5	2372.8	**7554.9**	6913.6	3278.4	2705.4	3615.9
15	N=90, F=25	0.0	0.0	3124.1	2268.3	**7520.0**	6862.6	3165.2	2018.8	3670.9
Medium level for setup times										
1	N=15, F=2	30.4	29.2	325.8	410.8	**468.1**	336.9	335.9	317.4	430.2
2	N=15, F=3	15.5	15.5	369.5	423.9	450.3	301.1	380.9	281.2	**506.7**
3	N=15, F=4	7.2	7.3	361.6	424.4	511.5	379.3	370.6	198.6	**529.6**
4	N=30, F=2	0.0	0.0	716.5	796.8	**1202.7**	945.2	733.8	715.8	908.8
5	N=30, F=5	0.0	0.0	633.0	780.1	**1223.2**	982.6	640.8	319.1	1027.2
6	N=30, F=10	0.0	0.0	733.1	874.5	**1310.8**	1072.0	748.6	96.7	1268.8
7	N=50, F=2	0.0	0.0	1037.4	1259.3	**2754.9**	2382.0	1042.0	1028.2	1360.2
8	N=50, F=5	0.0	0.0	1226.7	1297.5	**2809.3**	2433.4	1243.0	799.5	1740.7
9	N=50, F=10	0.0	0.0	1352.3	1387.3	**3165.0**	2763.8	1368.4	203.3	2503.8
10	N=70, F=5	0.0	0.0	1994.1	1813.9	**4878.6**	4375.8	2010.1	1186.8	2881.6
11	N=70, F=10	0.0	0.0	3253.1	2065.3	**5765.5**	5202.5	3297.6	904.2	5192.6
12	N=70, F=25	0.0	0.0	2282.7	2073.8	**5845.3**	5276.8	2361.7	23.9	5030.7
13	N=90, F=5	0.0	0.0	3267.2	2378.8	**7961.4**	7277.1	3295.8	2205.8	4430.1
14	N=90, F=10	0.0	0.0	3335.1	2468.7	**8076.9**	7416.4	3403.9	1060.5	5605.8
15	N=90, F=25	0.0	0.0	3417.1	2507.2	**8909.7**	8209.7	3500.2	0.3	7023.9
Large level for setup times										
1	N=15, F=2	37.0	35.9	347.8	436.0	510.4	376.8	358.5	332.4	**561.1**
2	N=15, F=3	20.0	20.0	411.0	469.1	517.6	362.5	423.1	246.6	**694.0**
3	N=15, F=4	27.2	27.3	409.1	476.6	605.6	468.1	419.4	136.3	**747.1**
4	N=30, F=2	0.0	0.0	737.5	820.2	**1260.7**	998.7	755.5	738.0	1142.7
5	N=30, F=5	0.0	0.0	679.9	840.2	1371.3	1125.5	689.7	190.0	**1514.0**
6	N=30, F=10	0.0	0.0	835.5	1000.6	1579.3	1320.0	858.9	1.5	**1906.8**
7	N=50, F=2	0.0	0.0	1064.0	1286.3	**2851.2**	2471.5	1068.7	1046.4	**1766.5**
8	N=50, F=5	0.0	0.0	1295.2	1358.7	3028.2	2644.2	1315.9	543.5	**2436.5**
9	N=50, F=10	0.0	0.0	1453.9	1512.6	3604.0	3181.6	1476.4	5.0	**3933.7**
10	N=70, F=5	0.0	0.0	2036.0	1870.4	5145.7	4632.7	2054.0	579.4	3949.2
11	N=70, F=10	0.0	0.0	3387.4	2189.4	6361.4	5776.9	3451.4	220.3	**7331.4**
12	N=70, F=25	0.0	0.0	2679.2	2408.2	7109.0	6479.6	2830.8	0.0	**8162.0**
13	N=90, F=5	0.0	0.0	3357.1	2441.4	**8321.8**	7625.0	3392.1	1545.7	**5871.1**
14	N=90, F=10	0.0	0.0	3453.4	2586.0	**8716.6**	8036.6	3546.5	293.7	**8119.2**
15	N=90, F=25	0.0	0.0	3721.3	2816.7	10562.6	9811.3	3861.6	0.0	**11788.3**

Van Wassenhove (1985). The non-distributive one uses the power of the precedence constraints used to improve the Lagrangian relaxation. The latter lower bound outperforms generally the other ones. Note also that the use of the multipliers adjustment method had a positive effect on the computational cost of LB_9. In the latter bound, we used the *SPT* heuristic and we computed the multipliers by maximizing the lower bound value. The principle remains valid for any other heuristic. LB_9 is usually greater than LB_7, but the latter bound is more computationally effective than LB_9 (10^{-4} s for LB_7 against 10^{-2} s almost LB_9).

h) According to Figure 1, some of the lower bounds are very computationally efficient (this is the case of LB_1, LB_2, LB_4, LB_5, LB_6 and LB_7). One of the possible exploitations of the lower bounds would be their incorporation into a branch-and-bound procedure. It may be that other lower bounds, that perform better, would not computationally effective in a branch and bound algorithm if the variation in term of performance is not significant. Based on this analysis, Table 8 presents for each series a set of suitable and/or potential lower bounds which can be used in a branch-and-bound procedure.

Table 4. Mean values for the lower bounds obtained for series 4

Group	(N, F)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
					Small level for setup times					
1	N=15, F=2	64.1	62.2	339.3	**399.5**	279.3	181.8	355.5	338.0	373.5
2	N=15, F=3	29.1	29.1	380.4	396.5	252.0	140.2	398.3	365.4	**422.5**
3	N=15, F=4	22.6	22.4	353.0	395.1	282.2	184.3	368.3	323.5	**397.9**
4	N=30, F=2	0.0	0.0	798.7	800.4	513.6	331.9	826.7	798.3	**861.2**
5	N=30, F=5	0.0	0.0	673.4	731.0	491.5	323.5	685.8	613.3	**754.0**
6	N=30, F=10	0.0	0.0	747.5	784.4	483.7	323.8	764.2	589.9	**857.5**
7	N=50, F=2	0.0	0.0	1147.7	**1219.8**	1062.4	816.7	1157.1	1146.4	1213.4
8	N=50, F=5	0.0	0.0	1399.1	1225.7	1063.4	804.5	1418.5	1319.0	**1507.3**
9	N=50, F=10	0.0	0.0	1476.1	1272.7	1128.0	849.8	1495.5	1229.1	**1711.9**
10	N=70, F=5	0.0	0.0	2318.8	1830.3	1676.0	1364.5	2349.8	2183.8	**2530.4**
11	N=70, F=10	0.0	0.0	3798.9	2106.6	1977.2	1614.5	3849.7	3320.8	**4243.5**
12	N=70, F=25	0.0	0.0	2344.1	1794.8	1668.3	1342.8	2396.3	1412.1	**2848.9**
13	N=90, F=5	0.0	0.0	3885.5	2361.5	2703.1	2252.7	3932.3	3701.8	**4153.2**
14	N=90, F=10	0.0	0.0	3888.7	2552.2	2515.9	2101.1	3992.2	3403.8	**4428.3**
15	N=90, F=25	0.0	0.0	3861.7	2324.9	2574.5	2148.5	3934.2	2771.4	**4580.3**
					Medium level for setup times					
1	N=15, F=2	72.2	70.2	358.4	419.7	301.9	201.3	375.7	352.1	**485.0**
2	N=15, F=3	33.7	33.4	417.6	430.3	286.5	170.4	436.1	333.5	**589.1**
3	N=15, F=4	49.2	49.5	400.5	438.6	338.6	233.9	416.7	247.0	**603.7**
4	N=30, F=2	0.0	0.0	817.4	818.7	536.6	349.2	846.1	816.8	**1058.1**
5	N=30, F=5	0.0	0.0	712.6	780.3	546.8	371.0	728.7	422.3	**1204.7**
6	N=30, F=10	0.0	0.0	847.2	888.6	580.9	406.6	872.0	188.5	**1509.2**
7	N=50, F=2	0.0	0.0	1171.6	1241.5	1094.1	842.8	1181.3	1164.6	**1583.2**
8	N=50, F=5	0.0	0.0	1461.6	1274.3	1143.5	877.1	1488.5	1056.5	**2099.6**
9	N=50, F=10	0.0	0.0	1581.0	1370.5	1286.5	991.0	1607.8	412.9	**3044.5**
10	N=70, F=5	0.0	0.0	2361.4	1873.9	1754.0	1436.0	2391.2	1567.4	**3520.3**
11	N=70, F=10	0.0	0.0	3955.0	2211.1	2173.5	1789.4	4035.4	1554.8	**6480.4**
12	N=70, F=25	0.0	0.0	2769.4	2096.0	2043.2	1680.7	2899.1	130.3	**6252.2**
13	N=90, F=5	0.0	0.0	3981.1	2412.4	2814.4	2354.1	4032.4	2927.7	**5487.6**
14	N=90, F=10	0.0	0.0	4026.1	2655.7	2702.7	2274.6	4147.5	1731.6	**7024.8**
15	N=90, F=25	0.0	0.0	4229.0	2575.5	3074.5	2614.0	4361.2	308.0	**8852.6**
					Large level for setup times					
1	N=15, F=2	81.7	79.7	383.2	445.4	331.4	226.8	401.7	374.7	**640.7**
2	N=15, F=3	39.3	38.8	464.8	474.4	331.5	210.4	484.7	314.2	**812.7**
3	N=15, F=4	84.3	84.0	455.5	492.7	407.9	295.3	474.8	201.8	**863.9**
4	N=30, F=2	0.0	0.0	843.1	842.2	565.6	372.2	872.8	844.6	**1339.2**
5	N=30, F=5	0.0	0.0	763.0	841.6	617.1	434.0	783.0	277.7	**1805.8**
6	N=30, F=10	0.0	0.0	965.8	1021.3	705.0	513.3	1003.1	26.7	**2290.9**
7	N=50, F=2	0.0	0.0	1201.7	1269.9	1134.4	878.3	1211.8	1187.3	**2097.9**
8	N=50, F=5	0.0	0.0	1537.1	1336.3	1244.8	968.6	1570.7	800.8	**2985.6**
9	N=50, F=10	0.0	0.0	1705.5	1495.9	1479.9	1165.7	1742.4	61.5	**4855.9**
10	N=70, F=5	0.0	0.0	2406.7	1929.8	1851.5	1525.2	2438.1	935.2	**4875.9**
11	N=70, F=10	0.0	0.0	4124.7	2342.5	2414.8	2003.7	4238.8	544.4	**9265.2**
12	N=70, F=25	0.0	0.0	3257.6	2459.6	2491.0	2076.9	3500.0	0.0	**10318.9**
13	N=90, F=5	0.0	0.0	4082.2	2473.7	2950.2	2479.6	4145.3	2248.1	**7322.5**
14	N=90, F=10	0.0	0.0	4180.6	2784.4	2925.2	2481.8	4336.8	689.2	**10298.8**
15	N=90, F=25	0.0	0.0	4607.5	2885.1	3664.3	3153.5	4825.0	0.0	**14979.5**

Table 5. Mean values for the lower bounds obtained for series 5

Group	(*N*, *F*)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
Small level for setup times										
1	*N*=15, *F*=2	0.0	0.0	0.0	**70.0**	37.1	12.0	0.0	0.0	0.0
2	*N*=15, *F*=3	0.0	0.0	0.0	**90.6**	42.0	11.8	0.0	0.0	0.3
3	*N*=15, *F*=4	0.0	0.0	0.0	**66.1**	49.0	19.3	0.0	0.0	0.0
4	*N*=30, *F*=2	0.0	0.0	0.0	**90.8**	31.5	4.4	0.0	0.0	0.0
5	*N*=30, *F*=5	0.0	0.0	0.0	**92.8**	30.7	3.2	0.0	0.0	0.0
6	*N*=30, *F*=10	0.0	0.0	0.0	**67.9**	35.5	5.4	0.0	0.0	0.0
7	*N*=50, *F*=2	0.0	0.0	0.0	**89.8**	40.7	9.6	0.0	0.0	0.0
8	*N*=50, *F*=5	0.0	0.0	0.0	**130.1**	39.8	6.0	0.0	0.0	0.0
9	*N*=50, *F*=10	0.0	0.0	0.0	**106.4**	41.5	8.6	0.0	0.0	0.0
10	*N*=70, *F*=5	0.0	0.0	0.0	**68.3**	36.3	11.5	0.0	0.0	0.0
11	*N*=70, *F*=10	0.0	0.0	0.0	**143.0**	40.2	3.4	0.0	0.0	0.0
12	*N*=70, *F*=25	0.0	0.0	0.0	**101.0**	38.0	5.5	0.0	0.0	0.0
13	*N*=90, *F*=5	0.0	0.0	0.0	**117.4**	30.5	3.9	0.0	0.0	0.0
14	*N*=90, *F*=10	0.0	0.0	0.0	**115.3**	34.1	6.0	0.0	0.0	0.0
15	*N*=90, *F*=25	0.0	0.0	0.0	**104.5**	35.1	4.9	0.0	0.0	0.0
Medium level for setup times										
1	*N*=15, *F*=2	0.0	0.0	0.0	**73.0**	39.2	13.0	0.0	0.0	0.0
2	*N*=15, *F*=3	0.0	0.0	0.0	**97.2**	46.6	14.2	0.0	0.0	1.4
3	*N*=15, *F*=4	0.0	0.0	0.0	**73.0**	56.4	24.8	0.0	0.0	0.0
4	*N*=30, *F*=2	0.0	0.0	0.0	**91.8**	32.9	5.0	0.0	0.0	0.0
5	*N*=30, *F*=5	0.0	0.0	0.0	**96.2**	33.2	3.9	0.0	0.0	0.0
6	*N*=30, *F*=10	0.0	0.0	0.0	**74.2**	40.7	8.0	0.0	0.0	0.0
7	*N*=50, *F*=2	0.0	0.0	0.0	**91.2**	41.8	10.2	0.0	0.0	0.0
8	*N*=50, *F*=5	0.0	0.0	0.0	**133.3**	41.5	6.9	0.0	0.0	0.0
9	*N*=50, *F*=10	0.0	0.0	0.0	**113.4**	45.7	10.5	0.0	0.0	0.0
10	*N*=70, *F*=5	0.0	0.0	0.0	**68.6**	37.4	12.2	0.0	0.0	0.0
11	*N*=70, *F*=10	0.0	0.0	0.0	**147.1**	42.9	3.9	0.0	0.0	0.0
12	*N*=70, *F*=25	0.0	0.0	0.0	**115.0**	43.8	6.8	0.0	0.0	0.0
13	*N*=90, *F*=5	0.0	0.0	0.0	**118.4**	31.7	4.2	0.0	0.0	0.0
14	*N*=90, *F*=10	0.0	0.0	0.0	**116.3**	35.4	6.3	0.0	0.0	0.0
15	*N*=90, *F*=25	0.0	0.0	0.0	**109.9**	38.9	5.6	0.0	0.0	0.0
Large level for setup times										
1	*N*=15, *F*=2	0.0	0.0	0.0	**76.8**	42.4	14.6	0.0	0.0	0.0
2	*N*=15, *F*=3	0.0	0.0	0.0	**104.9**	52.2	16.9	0.0	0.0	2.7
3	*N*=15, *F*=4	0.0	0.0	0.0	**82.0**	65.8	31.7	0.0	0.0	1.8
4	*N*=30, *F*=2	0.0	0.0	0.0	**92.8**	34.5	5.5	0.0	0.0	0.0
5	*N*=30, *F*=5	0.0	0.0	0.0	**99.6**	36.2	4.9	0.0	0.0	0.0
6	*N*=30, *F*=10	0.0	0.0	0.0	**82.7**	46.5	10.8	0.0	0.0	0.0
7	*N*=50, *F*=2	0.0	0.0	0.0	**92.9**	43.2	10.9	0.0	0.0	0.0
8	*N*=50, *F*=5	0.0	0.0	0.0	**138.5**	43.9	8.4	0.0	0.0	0.0
9	*N*=50, *F*=10	0.0	0.0	0.0	**122.7**	50.1	12.8	0.0	0.0	0.0
10	*N*=70, *F*=5	0.0	0.0	0.0	**69.9**	39.0	13.0	0.0	0.0	0.0
11	*N*=70, *F*=10	0.0	0.0	0.0	**152.3**	45.9	4.4	0.0	0.0	0.0
12	*N*=70, *F*=25	0.0	0.0	0.0	**130.6**	50.4	7.7	0.0	0.0	0.0
13	*N*=90, *F*=5	0.0	0.0	0.0	**119.9**	33.0	4.7	0.0	0.0	0.0
14	*N*=90, *F*=10	0.0	0.0	0.0	**118.0**	37.3	6.8	0.0	0.0	0.0
15	*N*=90, *F*=25	0.0	0.0	0.0	**119.0**	43.5	6.6	0.0	0.0	0.0

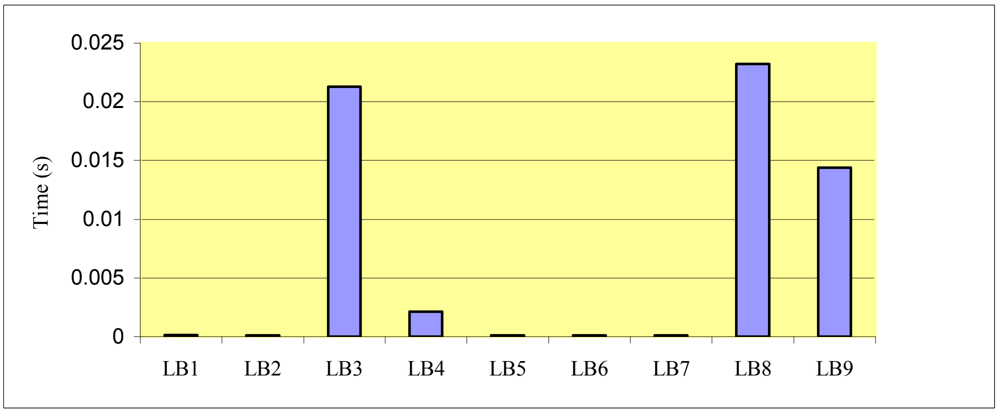

Figure 1. Mean values for the computation times.

6. CONCLUSION

In this paper, we consider the scheduling problem for a single machine with family setup times to minimize the total tardiness. We proposed a set of approaches to build lower bounds for the tardiness criterion. These lower bounds were analyzed and tested on a large set of numerical experiments. Two classes were proposed. The first class is distributive (i.e., the setup is split into pieces and distributed to the jobs of the corresponding family) and in the second class, the setup is not divided and it is considered independently of the processing times. A certain number of efficient lower bounds were identified (according to the data distribution). In our future work, we hope to build efficient branch and bound algorithms based on these results and to improve the effectiveness of these lower bounds by using some new dominance rules.

Table 6. Mean values for the lower bounds obtained for series 6

Group	(N, F)	LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
					Small level for setup times					
1	N=15, F=2	2152.9	2148.6	2251.7	703.1	2258.8	2137.7	2460.8	2253.9	**2490.6**
2	N=15, F=3	1896.2	1894.1	2052.2	669.9	2049.6	1904.1	2276.3	2039.0	**2319.3**
3	N=15, F=4	2127.5	2125.2	2222.5	711.2	2246.4	2118.5	2459.4	2194.1	**2510.9**
4	N=30, F=2	7643.8	7637.4	8147.5	1483.6	8031.7	7766.3	9190.8	8157.9	**9255.3**
5	N=30, F=5	7478.4	7473.4	7943.5	1478.7	7810.6	7555.4	8974.2	7856.8	**9112.3**
6	N=30, F=10	7565.6	7559.9	7979.6	1476.6	7959.7	7703.6	9114.9	7746.1	**9320.0**
7	N=50, F=2	19523.9	19518.0	20764.5	2481.6	20513.6	20037.7	23924.1	20772.5	**24050.5**
8	N=50, F=5	18448.0	18444.3	20051.0	2450.1	19880.4	19350.6	23327.4	19858.7	**23560.9**
9	N=50, F=10	19838.4	19832.6	21211.1	2524.2	20952.0	20448.1	24403.6	20770.8	**24841.6**
10	N=70, F=5	35875.3	35869.7	39377.2	3476.2	38266.3	37495.2	45554.9	39120.7	**45910.0**
11	N=70, F=10	39385.0	39377.2	42204.2	3577.4	41293.1	40567.6	48849.5	41566.9	**49511.2**
12	N=70, F=25	37948.8	37940.5	39869.2	3537.2	39823.6	39150.0	46885.7	38151.3	**47866.0**
13	N=90, F=5	58526.8	58523.4	64265.7	4473.5	62770.2	61744.0	74854.5	63934.4	**75312.2**
14	N=90, F=10	59540.8	59537.4	64860.9	4504.8	63188.2	62128.4	75761.7	64054.0	**76570.8**
15	N=90, F=25	58936.2	58930.2	64059.8	4507.8	62730.7	61714.2	75115.7	61606.9	**76726.9**
					Medium level for setup times					
1	N=15, F=2	2355.0	2349.2	2429.5	745.4	2436.4	2333.0	2643.5	2472.8	**2831.6**
2	N=15, F=3	2225.7	2221.3	2339.2	737.7	2343.6	2220.6	2573.4	2316.1	**2840.1**
3	N=15, F=4	2553.3	2544.9	2613.1	798.5	2632.4	2532.2	2852.7	2539.0	**3160.6**
4	N=30, F=2	7998.0	7991.1	8451.7	1521.8	8349.1	8094.1	9513.7	8528.6	**9899.9**
5	N=30, F=5	8442.4	8433.0	8729.5	1582.2	8680.3	8468.9	9839.3	8434.4	**10696.1**
6	N=30, F=10	9436.7	9424.5	9656.1	1679.3	9692.8	9509.1	10833.3	8653.6	**11876.2**
7	N=50, F=2	20208.9	20202.4	21316.1	2525.3	21121.8	20660.5	24535.9	21422.3	**25315.3**
8	N=50, F=5	19901.5	19895.0	21171.0	2545.0	21152.6	20656.7	24638.7	20484.5	**26054.8**
9	N=50, F=10	22901.3	22893.2	23789.5	2726.7	23692.4	23269.8	27172.7	21997.0	**29729.9**
10	N=70, F=5	37931.9	37924.7	40914.1	3573.8	40047.9	39311.0	47370.7	39603.7	**49523.0**
11	N=70, F=10	43575.4	43566.7	45608.1	3778.0	45077.4	44425.1	52658.2	42754.9	**56598.1**
12	N=70, F=25	48252.7	48239.0	49144.8	4053.9	49408.3	48912.7	56547.0	42448.7	**59281.0**
13	N=90, F=5	61212.3	61207.8	66152.1	4571.1	65067.5	64054.5	77219.4	64450.1	**80019.0**
14	N=90, F=10	64577.4	64569.9	68410.5	4692.6	67634.5	66653.6	80312.3	64300.6	**85223.8**
15	N=90, F=25	72454.3	72443.7	74332.5	5017.8	74847.4	74006.1	87361.5	65035.5	**94714.9**
					Large level for setup times					
1	N=15, F=2	2605.5	2597.9	2660.1	796.5	2666.2	2579.8	2874.9	2798.1	**3207.0**
2	N=15, F=3	2632.2	2622.2	2715.7	823.8	2718.4	2617.4	2949.3	2733.1	**3386.2**
3	N=15, F=4	3060.9	3051.5	3119.7	905.3	3112.0	3033.0	3338.5	3026.5	**3831.3**
4	N=30, F=2	8433.5	8425.8	8832.0	1569.8	8745.7	8510.1	9915.2	9026.7	**10548.0**
5	N=30, F=5	9613.7	9600.9	9782.2	1710.0	9774.2	9611.2	10913.4	9453.9	**12405.0**
6	N=30, F=10	11625.7	11611.4	11958.8	1931.0	11815.9	11667.3	12939.9	10157.7	**14479.6**
7	N=50, F=2	21055.6	21048.3	22014.8	2580.1	21884.3	21449.5	25296.8	22437.0	**26542.5**
8	N=50, F=5	21703.2	21695.8	22610.8	2663.9	22773.4	22316.9	26288.7	21764.6	**28667.9**
9	N=50, F=10	26576.0	26563.2	27224.9	2979.0	27102.8	26777.2	30593.9	24308.3	**34674.6**
10	N=70, F=5	40474.7	40467.0	42802.2	3694.9	42305.5	41618.6	49647.3	40906.2	**53365.3**
11	N=70, F=10	48652.3	48639.1	50052.9	4028.5	49812.7	49247.1	57373.4	45599.9	**63611.0**
12	N=70, F=25	59979.0	59961.6	62528.3	4695.2	60836.8	60420.6	68189.6	49535.7	**70604.0**
13	N=90, F=5	64529.3	64524.7	68539.8	4693.5	67949.7	66969.6	80174.1	65980.5	**84790.1**
14	N=90, F=10	70685.9	70675.0	72820.6	4926.7	73211.1	72320.1	85959.5	66579.5	**93848.1**
15	N=90, F=25	88300.6	88286.2	90658.5	5653.3	89921.9	89215.6	102412.9	73976.7	**112189.1**

Table 7. Mean values for the computation times (in seconds)

LB_1	LB_2	LB_3	LB_4	LB_5	LB_6	LB_7	LB_8	LB_9
0,00013086	0,00011561	0,02127658	0,00213457	0,00011375	0,00010037	0,00012082	0,02322788	0,01439257

Table 8. Suitable and/or potential lower bounds

Series	Non dominated lower bounds
1	LB_7, LB_9
2	LB_7, LB_9
3	LB_5, LB_9
4	LB_4, LB_7, LB_9
5	LB_4
6	LB_7, LB_9

ACKNOWLEDGMENTS

This study was partially funded by "Conseil Général de Champagne-Ardenne" and "Université de Technologie de Troyes".

REFERENCES

1. Ahn, B.H. and Hyun, J.H. (1990). Single facility multi-class job scheduling. *Computers and Operations Research*, 17: 265-272.
2. Akturk, M.S. and Ozdemir, D. (2001). A new dominance rule to minimize total weighted tardiness with unequal release dates. *European Journal of Operational Research*, 135: 394-412.
3. Allahverdi, A., Gupta, J.N.D., and Aldowaisan, T.A. (1999). A review of scheduling research involving setup considerations. *Omega*, 27: 219-239.
4. Allahverdi, A., Ng, C.T., Cheng, T.C.E., and Kovalyov, M.Y. (2007). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* (to appear).
5. Azizoglu, M. and Webster, S. (2003). Scheduling parallel machines to minimize weighted flowtime with family setup times. *International Journal of Production Research*, 41: 1199-1215.
6. Baker, K.R. (1999). Heuristic procedures for scheduling job families with setup and due dates. *Naval Research Logistics*, 46: 978-991.
7. Baker, K.R. (1982). A dynamic priority rule for scheduling against due dates. *Journal of Operations Management*, 3: 37-42.
8. Baptiste, Ph. and Jouglet, A. (2001). On minimizing total tardiness in a serial batching problem. *Rairo Operations Research*, 35: 107-115.
9. Chang, S., Lu, Q., and Yu, W. (1995). On decomposition of the total tardiness problem. *Operations Research Letters*, 17: 221-229.
10. Chu, C. (1992). A branch and bound algorithm to minimize the total tardiness with different release dates. *Naval Research Logistics*, 39: 265-283.
11. Crauwels, H.A.J., Hariri, A.M.A., Potts, C.N., and Van Wassenhove, L.N.V. (1998). Branch and bound algorithms for single machine scheduling with batch setup times to minimize total weighted completion time. *Annals of Operations Research*, 83: 59-76.
12. Crauwels, H.A.J., Potts, C.N., and Van Wassenhove, L.N.V. (1997). Local search heuristics for single machine scheduling with batch setup times to minimize total weighted completion time. *Annals of Operations Research*, 70: 261-279.
13. Della Croce, F., Tadei, R., Baracco, P., and Grosso, A. (1998). A new decomposition approach for the single machine total tardiness scheduling problem. *Journal of the Operational Research Society*, 49: 1101-1106.
14. Du, J. and Leung, J.Y.T. (1990). Minimizing total tardiness on one machine is NP-Hard. *Mathematics of Operations Research*, 15: 483-495.
15. Dunstall, S. and Wirth, A. (2005). A comparison of branch and bound algorithm for a family scheduling problem with identical parallel machines. *European Journal of Operational Research*, 167: 283-296.
16. Dunstall, S., Wirth, A., and Baker, K. (2000). Lower bounds and algorithms for flowtime minimization on a single machine with setup times. *Journal of Scheduling*, 3: 51-69.
17. Emmons, H. (1969). One-machine sequencing to minimize certain functions of job tardiness. *Operations Research*, 17: 701-715.
18. Ghosh, J.B. (1994). Batch scheduling to minimize total completion time. *Operations Research Letters*, 16: 271-275.
19. Gupta, J.N.D. (1988). Single facility scheduling with multiple job classes. *European Journal of Operational Research*, 8: 42-45.
20. Hariri, A.M.A. and Potts, C.N. (1997). Single machine scheduling with batch setup times to minimize maximum lateness. *Annals of Operations Research*, 70: 75-92.
21. Jouglet, A. (2002). *Ordonnancer Une Seule Machine Pour Minimiser Les Coûts*, PhD Thesis of the University of Technolgy of Compiègne, France.
22. Kondakci, S., Kirca, O., and Azizoglu, M. (1994). An efficient algorithm for the single machine tardiness problem. *International Journal of Production Economics*, 36: 213-219.
23. Koulamas, C. (1994). The total tardiness problem: Review and extensions. *Operations Research*, 42: 1025-1041.
24. Koulamas, C. (1997). Polynomially solvable total tardiness problems: Review and extensions. *Omega*, 25: 235-239.
25. Koulamas, C. and Kyparisis, G. (2001). Single machine scheduling with release times, deadlines and tardiness objectives. *European Journal of Operational Research*, 133: 447-453.
26. Lawler, E. (1977). A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1: 331-342.
27. Liaee, M.M. and Emmons, H. (1977). Scheduling families of jobs with setup times. *International Journal of Production Economics*, 51: 165-176.
28. Mason, A.J. and Anderson, E.J. (1991). Minimizing flow time on a single machine with job classes and setup times. *Naval Research Logistics*, 38: 333-350.
29. Nakumara, N., Yoshida, T., and Hitomi, K. (1978). Group production scheduling for total tardiness: (Part I). *AIIE Transactions*, 10: 157-162.

30. Potts, C. and Kovalyov, M.Y. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120: 228-249.
31. Potts, C. and Van Wassenhove, L.N.V. (1982). A decomposition algorithm for the single machine total tardiness problem. *Operations Research Letters*, 1: 177-191.
32. Potts, C. and Van Wassenhove, L.N.V. (1985). A branch and bound algorithm for the total weighted tardiness problem. *Operations Research*, 33: 363-377.
33. Potts, C. and Van Wassenhove, L.N.V. (1987). Dynamic programming and decomposition approaches for the single machine total tardiness problem. *European Journal of Operational Research*, 32: 405-414.
34. Ragatz, G.L. (1993). A branch-and-bound method for minimum tardiness sequencing on a single processor with sequence dependent setup times. *Proceedings of the 24th Annual Decision Sciences Institute*, November, 1993.
35. Rubin, P.A. and Ragatz, G.L. (1995). Scheduling in a sequence dependent setup environment with genetic search. *Computers and Operations Research*, 85-99.
36. Schaller, J. (2006). Scheduling on a single machine with family setups to minimize total tardiness. *International Journal of Production Economics* (in press).
37. Souissi, A., Kacem, I., and Chu, C. (2004). New lower-bounds for minimizing total tardiness on a single machine with sequence-dependent setup times. *Proceedings of the Project Management and Scheduling '04*, April, Nancy, France, pp. 26-28.
38. Tan, K.C., Narasimhan, R., Rubin, P.A., and Ragatz, G.L. (2000). A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times. *Omega*, 28: 313-326.
39. Szwarc, W., Della Croce, F., Tadei, R., and Grosso, A. (1999). Solution of the single machine total tardiness problem. *Journal of Scheduling*, 2: 55-71.