Engineering Optimization
Vol. 00, No. 00, September 2011, 1–13

Heuristic methods for the single machine scheduling problem with different ready times and a common due date

Ernesto G. Birgin[a] and Débora P. Ronconi[b]*

[a]Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090 São Paulo, SP - Brazil; [b]Department of Production Engineering, EP-USP, University of São Paulo, Av. Prof. Almeida Prado, 128, Cidade Universitária, 05508-900, São Paulo SP, Brazil

(*Received 9 June 2011; final version received 8 September 2011*)

The single machine scheduling problem with a common due date and non-identical ready times for the jobs is examined in this work. Performance is measured by the minimization of the weighted sum of earliness and tardiness penalties of the jobs. Since this problem is NP-hard, the application of constructive heuristics that exploit specific characteristics of the problem to improve their performance is investigated. The proposed approaches are examined through a computational comparative study on a set of 280 benchmark test problems with up to 1000 jobs.

Keywords: Scheduling; single machine; earliness and tardiness; heuristic methods.

1. Introduction

In the seventies, a new manufacturing philosophy named *Just in Time* (JIT) became worldwide, being the automobiles manufacturer Toyota one of its precursors (Womack *et al.* 1990). According to (Arnold 1998), JIT consists in the elimination of all waste and the continuous improvement of productivity. As one of the scheduling goals is to make the best use of manufacturing resources, the scheduling activity has an important responsibility in the improvement of the system performance when executed according to the JIT philosophy. Based on the premise of elimination of waste, the scheduling must balance the activities' execution trying to complete them near their due date, not before neither later. Illustrating the inconvenience of early activities, it is mentioned in (Bagchi *et al.* 1987) the case in which products completed before their due dates use the space reserved for stock and, consequently, cause a lack of space, as well as interfere in the cash flow due to the goods waiting for their delivery dates. Both situations

*Corresponding author. Email: dronconi@usp.br

ISSN: 0305-215X print/ISSN 1029-0273 online

DOI: 10.1080/0305215X.YYYY.CATSid
http://www.informaworld.com

may indicate an imbalance in the allocation and usage of resources and, therefore, an inefficient productive chain. The consequences of the tardy delivery of a product, like the loss of reputation among clients, the cost of lost sales and even financial costs related to contractual penalties, are analysed in (Davis and Kanet 1993). Summing up, costs involved in both situations, earliness and tardiness, stress the system inefficiency with respect to the scheduling of activities. Probably based on this motivation, many authors have considered the scheduling problem aiming to minimize earliness and tardiness in the delivery of goods. Comprehensive surveys on the common due date assignment and scheduling problems can be found in (Baker and Scudder 1990, Gordon *et al.* 2002).

Examples of the particular case in which several jobs share a common due date are given in (Sidney 1977, Kanet 1981). The production of perishable goods is approached in (Sidney 1977). The problem consists in schedule the production of items A, B and C. A is a perishable chemical product that must be combined with B to produce C. If A is produced before B being ready, it will deteriorate. If A is produced after B, the delay on producing C will have an associated additional cost. Another practical example is given in (Kanet 1981) and consists in the production of components for assembling a final product. In this case, the common due date is the date for starting to assemble the final product. Another simple example appears in a scenario in which a client orders several items that must be delivered altogether. Early jobs will have a stock cost while tardy jobs may have contractual penalties to be paid to the client. An extra constraint of the described scenario may be that each job might have its own ready time, related to the availability of the raw material needed to process the job. When jobs have non-identical ready times, the insertion of idle time in the scheduling may be advantageous (Ronconi and Powell 2010). See (Kanet and Sridharan 2000) for a literature review about scheduling with inserted idle time.

The problem with a common due date and common penalties α and β for earliness and tardiness, respectively, was tackled, for example, in (Panwalkar *et al.* 1982, Bagchi *et al.* 1987, Emmons 1987, Mondal and Sen 2001). However, none of them consider a different ready time for each job, i.e. all jobs are available since the beginning. Previous works that take into account different arrival times include (Nandkeolyar *et al.* 1993, Sridharan and Zhou 1996, Bank and Werner 2001, Cheng *et al.* 2002). In (Sridharan and Zhou 1996, Bank and Werner 2001) different penalties for each order are assumed. An unrelated parallel machine environment is considered in (Bank and Werner 2001), while (Sridharan and Zhou 1996) addresses the single machine with non-identical due dates. In (Nandkeolyar *et al.* 1993) a single machine with common due date is considered, but a common penalty for earliness and tardiness of each order is used, while (Cheng *et al.* 2002) addresses this problem with different penalties for earliness and tardiness for all jobs, but the due date has to be determined. The single machine problem with different penalties is also considered in (Valente and Alves 2007), where it is assumed that artificial idle times cannot be included in the schedule.

In the present work the problem of scheduling a set of jobs with non-identical ready times and a known restrictive common due date in a single machine environment is tackled. Earliness and tardiness have different penalties, that are common to all the jobs. The approached problem is NP-hard (Valente and Alves 2003). The heuristic method for the individual-due-date case proposed in (Sridharan and Zhou 1996) is evaluated to solve the problem with a common due date. Several improvements based on properties of the common-due-date case are proposed and analysed. The rest of the paper is organized as follows. Section 2 describes the problem and the proposed method. A numerical evaluation is presented in Section 3. Section 4 gives some concluding remarks.

Notation. Let $[\cdot]_+ = \max\{\cdot, 0\}$.

2. Description of the problem and proposed method

The problem consists of n production orders with different processing times p_i, $i = 1\dots,n$, on the single machine, and different ready times r_i, $i = 1,\dots,n$. There is also a common due date d and common positive penalties α and β for each unit of earliness and tardiness, respectively. It is considered that all data is deterministic and known in advance. Preemption is not allowed. The goal is to find a schedule that minimizes the total weighted sum of earliness and tardiness.

A mixed-integer linear programming (MILP) model for the problem follows. See (Ronconi and Birgin 2011) for an evaluation of MILP models for flowshop scheduling problems minimizing the total earliness and tardiness. Variables of the formulation are: $x_{ij} \in \{0,1\}, i = 1,\dots,n-1, j = i+1,\dots,n$, $E_i \geq 0, i = 1,\dots,n$, $T_i \geq 0, i = 1,\dots,n$, and $s_i, i = 1,\dots,n$. In the model, $x_{ij} = 1$ if job i precedes job j in the sequence (not necessarily immediately before it), $x_{ij} = 0$ otherwise. E_i, T_i, and s_i stand for earliness, tardiness, and start time of job i, respectively.

$$\text{Minimize } \sum_{i=1}^{n} \alpha E_i + \beta T_i \tag{1}$$

$$\text{subject to } E_i \geq d - s_i - p_i, \qquad i = 1,\dots,n, \tag{2}$$

$$T_i \geq s_i + p_i - d, \qquad i = 1,\dots,n, \tag{3}$$

$$s_i + p_i \leq s_j + M(1 - x_{ij}), \qquad i = 1,\dots,n-1, j = i+1,\dots,n, \tag{4}$$

$$s_j + p_j \leq s_i + M x_{ij}, \qquad i = 1,\dots,n-1, j = i+1,\dots,n, \tag{5}$$

$$s_i \geq r_i, \qquad i = 1,\dots,n. \tag{6}$$

In the formulation above, the objective function (1) is the weighted sum of earliness and tardiness of the jobs. Constraints (2) and (3), together with the non-negativity constraints $E_i \geq 0$ and $T_i \geq 0$, give the earliness and tardiness of each job, i.e. $E_i = \max\{0, d - (s_i + p_i)\}$ and $T_i = \max\{0, (s_i + p_i) - d\}$. Constraints (4) and (5) say that each job must be completed before a job that follows it starts to be processed. In the constraints, M is a large positive number. Constraint (6) indicates that a job cannot start to be processed before its ready time.

The rest of the present section is organized as follows. Subsection 2.1 describes the heuristic method proposed in (Sridharan and Zhou 1996) that considers individual due dates for each job. Subsections 2.2 and 2.3 present modifications to this method that greatly improve its performance when applied to the common due date situation, as suggested by the numerical experiments. The improvement proposed in Subsection 2.2 is based on the existence of a polynomial time algorithm to compute the optimal schedule for a given sequence (Sakuraba *et al.* 2009); while the improvement suggested in Subsection 2.3 uses the so-called "V-shaped shedule" property (Hoogeveen and Vandevelde 1991, Biskup and Feldmann 2001) that applies to the common due date case with ready times $r_i = 0\ \forall i$.

2.1. *Heuristic method DT-ET*

In the present subsection, the heuristic method proposed in (Sridharan and Zhou 1996), called DT-ET (Decision Theory for Earliness and Tardiness), is described. While the method considers individual due dates for each job, a particularization for a common due date version is described below. The study presented in (Nandkeolyar *et al.* 1993) motivated the reanalysis of the DT-ET heuristic. In their study, the authors provide a comparison among twelve different heuristic methods. The evaluation is based on a modular view of the analysed strategies. The three considered modules were: (i) *Main Theme* (main rule used to define the next job to be dispatched), (ii) *Front End* (look-ahead vision considering job arriving after the decision time in order to create a future scenario), and (iii) *Balancing Routine* (the possibility of including idle time in the schedule). The numerical experiments in (Nandkeolyar *et al.* 1993) suggest that methods including modules of type (ii) and (iii) as for the DT-ET introduced in (Sridharan and Zhou 1996) present the best performances.

According to (Sridharan and Zhou 1996), the basic idea of the DT-ET heuristic may be summarized as follows. When a resource completes processing a job, two questions are raised: (a) which job should be processed next, and (b) when to start its processing. To answer the first question, the jobs currently in the queue plus some arriving jobs are considered. To answer the second question, the future impact of choosing a job upon all the remaining jobs is analysed. The DT-ET heuristic is now described in details.

The input data for subroutine DT-ET is: the common earliness and tardiness penalties α and β, the common due date d, the number of jobs n and, for each job $i \in \{1, \ldots, n\}$, its processing time p_i and release date r_i. On output, DT-ET returns a sequence π of the given jobs, the completion time C_j of the j-th job, namely π_j, for $j = 1, \ldots, n$, and the objective function value f associated with π and C. Algorithm 1 shows the pseudo code of DT-ET. A detailed description of each step follows.

Algorithm 1: Pseudo code of the heuristic method DT-ET introduced in Sridharan and Zhou (1996).

Input: $\alpha, \beta, d, n \in \mathbb{Z}$, $p, r \in \mathbb{Z}^n$.
Output: Sequence π, completion time $C \in \mathbb{Z}^n$ and objective function value $f \in \mathbb{Z}$.
DT-ET$(\alpha, \beta, d, n, p, r)$
begin
    $f \leftarrow 0$
    for $i \leftarrow 1$ **to** n **do**
        Set current time t_0
        Build set $S(t_0)$ of candidate jobs to be sequenced at position i
        foreach $j \in S(t_0)$ **do**
            Consider sequencing job j at position i and find a feasible completion time $\widehat{C}_j$
            Estimate the total cost κ_j associated with this choice, considering jobs in $S(t_0) \setminus \{j\}$
        Compute $j_{\min} = \arg\min_{j \in S(t_0)} \{\kappa_j\}$
        $\pi_i \leftarrow j_{\min}$ and $C_i \leftarrow \widehat{C}_{j_{\min}}$
        $f \leftarrow f + \alpha\, [d - C_i]_+ + \beta\, [C_i - d]_+$
    return π, C and f
end

To simplify the explanations, let F_i and N_i be the sets of fixed and non-fixed jobs at the beginning of iteration i, respectively. Obviously, $F_1 = \emptyset$, $N_1 = \{1, \ldots, n\}$ and (F_i, N_i) is a partition of $\{1, \ldots, n\}$ for $i = 1, \ldots, n$.

The current time t_0 at line 4 is computed as:

$$t_0 = \begin{cases} r_{\min}, & \text{if } i = 1, \\ \max\{r_{\min}, C_{i-1}\}, & \text{otherwise}, \end{cases} \tag{7}$$

where $r_{\min}$ is the smallest release date among the non-fixed jobs, i.e. $r_{\min} = \min_{k \in N_i}\{r_k\}$.

Set $S(t_0)$ at line 5 is composed by jobs that will be available before the common due date d, jobs that are available at time t_0 and jobs that will be available soon. Specifically, $S(t_0)$ is computed as

$$S(t_0) = U \cup V, \tag{8}$$

where

$$U = \{k \in N_i \mid r_k \leq t_0\}$$

and

$$V = \{k \in N_i \setminus U \mid r_k \leq \max\{t_0 + \max_{j \in U}\{p_j\}\}, d\}.$$

The loop on lines 6–8 considers the possibility of sequencing each job $j \in S(t_0)$ at position i. For each j, two quantities are computed: (a) a feasible completion time $\widehat{C}_j$ (line 7); and (b) an estimate κ_j of the total cost, considering all the other jobs in $S(t_0)$, associated with the impact of sequencing job j at position i with completion time $\widehat{C}_j$ (line 8).

Let

$$\widehat{C}_j^{\text{earliest}} = \max\{t_0, r_j\} + p_j, \tag{9}$$

be the earliest possible completion time for job j. If $\widehat{C}_j^{\text{earliest}} \geq d$, there is nothing better to be done for job j other than setting $\widehat{C}_j = \widehat{C}_j^{\text{earliest}}$. On the other hand, if $\widehat{C}_j^{\text{earliest}} < d$, any value $\widehat{C}_j \in [\widehat{C}_j^{\text{earliest}}, d]$, with an associated penalty given by $\alpha(d - \widehat{C}_j)$, would be possible; the optimal option for $\widehat{C}_j$, considering only job j, being $\widehat{C}_j = d$. Therefore, ignoring the impact on the remaining jobs, the optimal choice $\widehat{C}_j^*$ for $\widehat{C}_j$ is given by

$$\widehat{C}_j^* = \max\{\widehat{C}_j^{\text{earliest}}, d\}. \tag{10}$$

On the other hand, according to (Sridharan and Zhou 1996), aiming to reduce the impact on the remaining jobs in $S(t_0)$, $\widehat{C}_j$ may be chosen as

$$\widehat{C}_j^{\text{shifted}} = \max\{\widehat{C}_j^{\text{earliest}}, \widehat{C}_j^* - (\bar{C} - d)\}, \tag{11}$$

where

$$\begin{array}{ll} \bar{C} = \widehat{C}_j^* + \bar{R} + (P - \bar{P})/2 + \bar{P}, & \\ \bar{P} = P/(|S_{(t_0)}| - 1), & P = \sum_{k \in S(t_0) \setminus \{j\}} p_k, \\ \bar{R} = R/(|S_{(t_0)}| - 1), & R = \sum_{k \in S(t_0) \setminus \{j\}} \max\{r_k, t_0\}, \end{array}$$

and $\bar{P}$, $\bar{R}$ and $\bar{C}$ stand for average processing time, release time and completion time of the remaining jobs in $S(t_0)$. The decision between $\widehat{C}_j^*$ and $\widehat{C}_j^{\text{shifted}}$ depends on the earliness and tardiness penalties and is given by

$$\widehat{C}_j = \begin{cases} \widehat{C}_j^*, & \text{if } \alpha \geq \beta, \\ \widehat{C}_j^{\text{shifted}}, & \text{otherwise.} \end{cases} \tag{12}$$

According to (Sridharan and Zhou 1996), the total cost associated to the possibility of sequencing job j at position i with completion time $\widehat{C}_j$ is given by

$$\kappa_j = \sum_{k \in S(t_0)} \alpha \, [d - \widehat{C}_k]_+ + \beta \, [\widehat{C}_k - d]_+, \tag{13}$$

where

$$\widehat{C}_k = \max\{d, r_k + p_k, \widehat{C}_j + (P - p_k)/2 + p_k\}, \; \forall \, k \in S(t_0) \setminus \{j\}. \tag{14}$$

Note that, as $\widehat{C}_k \geq d$, $\forall k \neq j$, it is assumed in (14) that all the remaining jobs in $S(t_0)$ will be on time or late.

Finally, on line 9, the job $j \in S(t_0)$ with the smallest associated total cost κ_j is chosen to be sequenced at position i with completion time $C_i = \widehat{C}_j$ (line 10). The value of the objective function is updated at line 11.

Algorithm 1 with the definitions (7), (8), (12) and (13) complete the description of the method proposed in (Sridharan and Zhou 1996), named DT-ET. It is easy to see that this method can be implemented with time complexity $O(n^2)$, if common expressions are used to compute κ_j, $\forall \, j \in S(t_0)$, in the inner loop on lines 6–8.

2.2. *First improvement: using the* TIMING *algorithm*

For the common due date case with different release dates, there exists a polynomial time algorithm to compute the optimal schedule for a given sequence (Sakuraba *et al.* 2009). It means that the output sequence of the method proposed in (Sridharan and Zhou 1996) can be used as an input to the method proposed in (Sakuraba *et al.* 2009) to obtain an optimal schedule. The timing algorithm introduced in (Sakuraba *et al.* 2009) provides the optimal schedule for a given jobs sequence π, i.e. it provides the optimal completion time for each operation. The method computes the completion times C_j of the j-th job, namely π_j, for all j, that minimizes the objective function

$$f(C) = \sum_{k=1}^{n} \alpha[d - C_k]_+ + \beta[C_k - d]_+. \tag{15}$$

The calculation is based on the fact that function $f(\cdot)$ in (15) is a convex piecewise linear function (see (Sakuraba *et al.* 2009) for details). The method was developed for the case in which each job i has its own earliness and tardiness penalties α_i and β_i, respectively. For completeness, Algorithm 2 reproduces the timing algorithm for the case of interest of the present work: $\alpha_i = \alpha$ and $\beta_i = \beta$ for all i. The timing algorithm returns an index k and a time W such that jobs $\pi_1, \ldots, \pi_{k-1}$ must be processed without idle time and with π_{k-1} being completed at time W. The remaining jobs $\pi_k, \ldots, \pi_n$ must be processed starting at time W and as early as possible. Time complexity of the timing algorithm is $O(n)$.

Algorithm 2: Pseudo code of the timing algorithm introduced in Sakuraba *et al.* (2009) to compute the optimal schedule of a given sequence.

Input: $\alpha, \beta, d, n \in \mathbb{Z}$, $p, r \in \mathbb{Z}^n$, sequence π.
Output: $W, k \in \mathbb{Z}$ such that jobs $\pi_1, \ldots, \pi_{k-1}$ must be scheduled ending at W without idle time between them, while jobs $\pi_k, \ldots, \pi_n$ must be scheduled at their earliest possible time after W.

Timing$(\alpha, \beta, d, n, p, r, \pi)$

begin
    $\hat{\alpha} \leftarrow 0$, $\hat{\beta} \leftarrow 0$
    $S \leftarrow d$, $W \leftarrow d$, $\Delta \leftarrow d$
    $k \leftarrow 1$
    $h \leftarrow 1$
    while $k \leq n$, $\Delta > 0$, $r_{\pi_k} < W$ **do**
        $\hat{\beta} \leftarrow \hat{\beta} + \beta$
        $W \leftarrow W + p_{\pi_k}$
        $\Delta \leftarrow \min\{\Delta, W - p_{\pi_k} - r_{\pi_k}\}$
        while $\Delta > 0$, $\hat{\beta} \geq \hat{\alpha}$ **do**
            if $\Delta > p_{\pi_h}$ **then**
                $\hat{\alpha} \leftarrow \hat{\alpha} + \alpha$, $\hat{\beta} \leftarrow \hat{\beta} - \beta$
                $S \leftarrow S - p_{\pi_h}$, $W \leftarrow W - p_{\pi_h}$, $\Delta \leftarrow \Delta - p_{\pi_h}$
                $h \leftarrow h + 1$
            else
                $S \leftarrow S - \Delta$, $W \leftarrow W - \Delta$, $\Delta \leftarrow 0$
        $k \leftarrow k + 1$
    return W and k
end

Algorithm 3 shows the algorithm named DT-ET+Timing that calls subroutine DT-ET to compute a sequence by the method introduced in (Sridharan and Zhou 1996) and then, with the help of subroutine Timing and preserving the sequence π given by DT-ET, computes the optimal schedule.

2.3. *Further improvements based on problem properties*

The method presented in (Sridharan and Zhou 1996) was originally developed for the case in which each job i has its own due date d_i. Of course, it can be applied to the common due date case too. However, for the latter case, some properties of the problem allow the improvement of the estimation of the completion time $\widehat{C}_j$ and its associated total cost κ_j for all $j \in S(t_0)$ (in the inner loop on lines 6–8 of Algorithm 1).

Consider iteration i of DT-ET subroutine described in Algorithm 1. For each job $j \in S(t_0)$, the method computes a plausible completion time $\widehat{C}_j$ and estimates its associated total cost κ_j. If the earliest possible completion time $\widehat{C}_j^{\text{earliest}}$ (9) for job j, sequenced at position i, is greater than or equal to the common due date d, then there is nothing to be decided. Otherwise, any value of $\widehat{C}_j \in [\widehat{C}_j^{\text{earliest}}, d]$ would be possible. In fact, following (Sridharan and Zhou 1996), the possibilities are reduced to select one between $\widehat{C}_j^*$ (10) and $\widehat{C}_j^{\text{shifted}}$ (11). The decision made in (Sridharan and Zhou 1996) between those values depends on the comparison between penalties α and β, as stated in (12).

Algorithm 3: Pseudo code of the first improvement to the DT-ET method.

Input: $\alpha, \beta, d, n \in \mathbb{Z}$, $p, r \in \mathbb{Z}^n$.
Output: Sequence π, completion time $C \in \mathbb{Z}^n$ and objective function value $f \in \mathbb{Z}$.
DT-ET+TIMING$(\alpha, \beta, d, n, p, r)$
begin
  $(\pi, C, f) \leftarrow$ DT-ET$(\alpha, \beta, d, n, p, r)$
  $(W, k) \leftarrow$ TIMING$(\alpha, \beta, d, n, p, r, \pi)$
  if $k \geq 2$ **then**
    $C_{k-1} \leftarrow W$
    for $i \leftarrow k-2$ **to** 1 **do**
      $C_i \leftarrow C_{i+1} - p_{\pi_{i+1}}$
  if $k \leq n$ **then**
    $C_k \leftarrow \max\{W, r_{\pi_k} + p_{\pi_k}\}$
    for $i \leftarrow k+1$ **to** n **do**
      $C_i \leftarrow \max\{C_{i-1}, r_{\pi_i}\} + p_{\pi_i}$
  $f \leftarrow 0$
  for $i \leftarrow 1$ **to** n **do**
    $f \leftarrow f + \alpha[d - C_i]_+ + \beta[C_i - d]_+$
  return π, C and f
end

For the particular case of a common due date, as in the problem being tackled in the present work, the impact of shifting the i-th job to the left can be better estimated. In the present work, it is estimated considering only the jobs in $S(t_0)$. On one hand, by setting $\widehat{C}_j = \widehat{C}_j^* = d$ (recall that it is being assumed $\widehat{C}_j^{\text{earliest}} < d$, i.e. $\widehat{C}_j^* = d$), job j, sequenced at position i, will be completed on the due date (paying no penalties) and all the remaining jobs in $S(t_0)$ will be delayed, adding an individual average tardiness of $\beta(\bar{C} - d)$ to the objective function, i.e. adding a total average tardiness of

$$\beta\ (\bar{C} - d)\ (|S(t_0)| - 1). \tag{16}$$

On the other hand, by setting $\widehat{C}_j = \widehat{C}_j^{\text{shifted}}$, and assuming $\widehat{C}_j^* - (\bar{C} - d) \geq \widehat{C}_j^{\text{earliest}}$, job j will pay an earliness penalty given by

$$\alpha\ (d - \widehat{C}_j^{\text{shifted}}) = \alpha\ (\bar{C} - d), \tag{17}$$

while the remaining jobs in $S(t_0)$ will be completed, on average, on the due date, paying no penalties (that is clearly a simplification of the real situation). Therefore, the natural criterion to select one between $\widehat{C}_j^*$ and $\widehat{C}_j^{\text{shifted}}$ is to compare (16) with (17), which can be further reduced to comparing α with $\beta\ (|S_{t_0}| - 1)$. Summing up, the modification consists in replacing the computation of $\widehat{C}_j$ in (12) by

$$\widehat{C}_j = \begin{cases} \widehat{C}_j^*, & \text{if } \alpha \geq \beta\ (|S_{t_0}| - 1), \\ \widehat{C}_j^{\text{shifted}}, & \text{otherwise.} \end{cases} \tag{18}$$

The second improvement is related to the estimation of the total cost associated with sequencing job j at position i, given by κ_j. Note that the estimation of the total cost (13) used in (Sridharan and Zhou 1996) is based on average values related to the remaining jobs in $S(t_0)$, while neither a plausible schedule nor a sequence are assumed for the jobs. The improvement on this estimation introduced in the present work is based on a property of the tackled problem for the particular case $r_i = 0, \ \forall i$. The property (Hoogeveen and Vandevelde 1991, Biskup and Feldmann 2001), known as "V-shaped schedule", can be stated as follows:

There exists an optimal solution such that the schedule is V-Shaped, that is, jobs that are completed at or before the due date are sequenced in a non-increasing order of their processing time p_i, while jobs that are started at or after the due date are sequenced in a non-decreasing order of their processing time p_i.

When considering job $j \in S(t_0)$ to be sequenced at position i with completion time $\widehat{C}_j$ computed as suggested in (18), the estimation of the total cost κ_j proceeds as follows:

- **(a)** Let $X \subset S(t_0)$ be the set of jobs in $S(t_0)$ with processing time less than or equal to p_j, and let $Y = S(t_0) \setminus X$. Let $D \leftarrow \widehat{C}_j$ and initialize $\kappa_j \leftarrow \alpha[d-D]_+ + \beta[D-d]_+$.
- **(b)** While $X \neq \emptyset$ and $D < d$, let $k \in X$ be the job with largest processing time p_k. Ignoring the release date r_k, set $D \leftarrow D + p_k$, and, pretending that job k can be scheduled to be completed on D, set $\kappa_j \leftarrow \kappa_j + \alpha[d - D]_+ + \beta[D - d]_+$. Let $X \leftarrow X \setminus \{k\}$.
- **(c)** Let $Z = Y \cup X$. While $Z \neq \emptyset$, let $k \in Z$ be the job with shortest processing time p_k. Ignoring the release date r_k, set $D \leftarrow D + p_k$, and, pretending that job k can be scheduled to be completed on D, set $\kappa_j \leftarrow \kappa_j + \alpha[d - D]_+ + \beta[D - d]_+$. Let $Z \leftarrow Z \setminus \{k\}$.

Steps (a)–(c) can be executed in $O(|S(t_0)|)$ if jobs in $S(t_0)$ are ordered by their processing time, as shown in Algorithm 4. Each ordered set $S(t_0)$ constructed on line 5 of Algorithm 1 can be computed in $O(n)$ provided that the original set of n jobs is sorted once. Therefore, Algorithm 1 with the two modifications proposed in the present subsection can be implemented with time complexity $O(n^2)$. Since the TIMING algorithm, depicted in Algorithm 2, has time complexity $O(n)$, Algorithm 3 with the modifications proposed in the present subsection has time complexity $O(n^2)$ too, as the one presented in (Sridharan and Zhou 1996).

3. Numerical experiments

In order to evaluate the methods proposed in the present work, instances with different number of jobs and different scenarios for the due date were considered. For each $n \in \{10, 20, 50, 100, 200, \ 500, 1000\}$ and $scen \in \{1, 2, 3, 4\}$, ten different instances were considered. It means that the set of test instances consists of 280 different test instances.

Processing times of the jobs where taken from (Biskup and Feldmann 2001), where problems with the same dimensions were considered. Earliness and tardiness penalties were also taken from (Biskup and Feldmann 2001). Since in (Biskup and Feldmann 2001) each job j has its own penalties α_j and β_j, it was arbitrarily considered $\alpha = \alpha_1$ and $\beta = \beta_1$. In scenario i, for $i = 1, \dots, 4$, the common due date d is given by

$$d = 0.2i \sum_{k=1}^{n} p_k.$$

It means that scenarios $1, \dots, 4$ range from the one with the more restrictive due date to the one with the loosest, still restrictive, due date. For each instance, release dates are

Algorithm 4: Pseudo code of the estimated total cost computation.

Input: $\alpha, \beta, d, n, k, \widehat{C} \in \mathbb{Z}$, $p \in \mathbb{Z}^n$ and set $S(t_0) = \{w_1, \ldots, w_{|S(t_0)|}\}$ with $w_1 \leq \cdots \leq w_{|S(t_0)|}$.
Output: Estimated total cost $\kappa \in \mathbb{Z}$ for the k-th job in $S(t_0)$ scheduled to be completed at $\widehat{C}$.

EstimatedTotalCost$(\alpha, \beta, d, n, k, \widehat{C}, p)$

begin
  Set $X \leftarrow \{w_1, \ldots, w_{k-1}\}$ and $Y \leftarrow \{w_{k+1}, \ldots, w_{|S(t_0)|}\}$.
  Set $D \leftarrow \widehat{C}$ and $\kappa \leftarrow \alpha[d - D]_+ + \beta[D - d]_+$.
  $i \leftarrow k + 1$
  while $i \leq |S(t_0)|$ *and* $D < d$ **do**
    $D \leftarrow D + p_i$
    $\kappa \leftarrow \kappa + \alpha[d - D]_+ + \beta[D - d]_+$
  $ilim \leftarrow i$
  for $i \leftarrow |S(t_0)|$ **to** $ilim$ **do**
    $D \leftarrow D + p_i$
    $\kappa \leftarrow \kappa + \alpha[d - D]_+ + \beta[D - d]_+$
  for $i \leftarrow k$ **to** 1 **do**
    $D \leftarrow D + p_i$
    $\kappa \leftarrow \kappa + \alpha[d - D]_+ + \beta[D - d]_+$
  return κ
end

computed as $\lceil r \rceil$, where r is a uniformly distributed random variable in $[0, \frac{1}{2}\sum_{k=1}^{n} p_k]$. The whole set of instances is available at *http://www.ime.usp.br/∼egbirgin/* for further comparisons and benchmarking.

When comparing a method A against a method B, the PD (percentage difference) measurement given by

$$PD = 100(f_A - f_B)/f_A\%,$$

was used, where f_A is the objective function value found by method A and f_B is the objective function value found by method B. As ten different instances for each scenario and problem dimension were considered, the *average* PD, that can be interpreted as the average percentage of relative improvement of method B over method A was analyzed. Therefore, positive average PD values mean that, in average, method B found better solutions than method A.

Method DT-ET proposed in (Sridharan and Zhou 1996) corresponds to Algorithm 1 with the definitions (7), (8), (12) and (13). Method DT-ET computes a sequence and a schedule for a given set of jobs. However, given a sequence, the optimal scheduling can be found by the method proposed in (Sakuraba *et al.* 2009). Therefore, a first experiment evaluates the improvement provided by replacing the schedule given by method DT-ET with the optimal one computed using the timing method proposed in (Sakuraba *et al.* 2009). This combination corresponds to Algorithm 3, with the definitions (7), (8), (12) and (13), and is called Method DT-ET+Timing. Table 1 shows the average PD values when comparing Method DT-ET against Method DT-ET+Timing. The figures in the table show that replacing the schedule given by Method DT-ET by the one computed using the timing algorithm provides a reasonable improvement.

A second set of experiments aims to evaluate the modifications to Method DT-ET proposed on Subsections 2.2 and 2.3. The proposed method is given by Algorithm 3 with definitions (7), (8) and (18) and using Algorithm 4 to compute the total cost estimation κ_j. This method is called Method DT-ET-CDD (DT-ET for the common due

Table 1.: Comparison of Method DT-ET against Method DT-ET+Timing.

Scen	10	20	50	100	200	500	1,000	Average
1	0.31	0.20	0.15	0.15	0.09	0.14	0.13	0.17
2	2.35	2.57	1.29	0.65	0.60	0.69	0.63	1.25
3	5.56	7.10	6.82	9.00	6.26	6.43	7.18	6.91
4	9.81	7.32	6.57	10.14	6.85	6.34	8.11	7.88
Average	4.51	4.30	3.71	4.98	3.45	3.40	4.01	

(Columns 10–1,000: Number of jobs)

Table 2.: Comparison of Method DT-ET against Method DT-ET-CDD.

Scen	10	20	50	100	200	500	1,000	Average
1	1.52	1.84	7.89	7.72	7.67	5.72	6.47	5.55
2	8.22	11.62	12.62	16.07	12.28	7.79	8.44	11.01
3	17.48	21.94	21.70	28.64	26.63	21.69	23.87	23.14
4	23.19	29.04	33.06	38.21	37.62	33.06	36.78	32.99
Average	12.60	16.11	18.82	22.66	21.05	17.06	18.89	

Table 3.: Comparison of Method DT-ET-CDD−Timing (DT-ET-CDD *minus* Timing) and Method DT-ET-CDD.

Scen	10	20	50	100	200	500	1,000	Average
1	0.08	0.13	0.19	0.10	0.06	0.08	0.09	0.11
2	0.36	0.17	0.22	0.54	0.47	0.63	0.77	0.45
3	1.31	0.96	0.57	1.29	0.83	1.04	1.22	1.03
4	1.51	0.88	1.01	0.99	1.03	1.92	1.13	1.21
Average	0.81	0.54	0.50	0.73	0.60	0.92	0.80	

date situation) from now on. Table 2 shows the comparison of Method DT-ET against Method DT-ET-CDD. The figures in the table show that, although Method DT-ET is applicable to the considered problem, the modifications based on characteristics specific to the common due date case make Method DT-ET-CDD a preferable choice.

Looking at Tables 1 and 2, it could be concluded that, for example, for $n = 10$, approximately one third of the 12.60% improvement of Method DT-ET-CDD over Method DT-ET is provided by the usage of the timing method to find the optimal schedule for the computed sequence. However, it is worth noting that this is not true. The improvement of 4.51% when using the timing method is given over the solution found by Method DT-ET, as shown in Table 1, but similar improvements are not observed when applying the timing method to the sequences constructed by Method DT-ET-CDD. Table 3 shows the comparison between Method DT-ET-CDD−Timing (DT-ET-CDD *minus* Timing) and DT-ET-CDD. The results on Table 3 (small improvements given by the timing method when applied to the sequence given by Method DT-ET-CDD−Timing) suggest that, for the computed sequence, the schedule given by Method DT-ET-CDD−Timing seems to be closer to the optimal schedule than the schedule provided by Method DT-ET.

4. Concluding remarks

An heuristic method for the single machine scheduling problem with a common due date and non-identical ready times for the jobs was developed in this work. The introduced method exploits specific characteristics of the problem being tackled and the numerical experiments show empirical evidences of its efficiency and robustness. The situation in which each job has its own earliness and tardiness penalty, as well as the multiple machine case, will be subjects of future research.

The common due date situation is a particular case of the individual due date situation and, therefore, any method developed for the later case can be applied to the former one. However, considering properties of the problem being tackled on the development of a method can greatly improve its performance, as shown in the present work. In some sense, this claim favours the development of specialized methods in detriment of methodologies that, with few or no modifications at all, can be applied to a wide range of different problems. While this methods may have the positive characteristic of delivering a "reasonable" solution with a low development effort, their performances could be far more competitive if the underlying heuristics fully explore characteristics and properties of the problem being solved.

Acknowledgements

This work was supported by PRONEX-CNPq/FAPERJ (E-26/171.1510/2006-APQ1), FAPESP (Grants 2006/53768-0, 2006/03496-3, 2009/10241-0, and 2010/10133-0), CNPq (308000/2009-9 and 304484/2007-5).

References

Arnold, J.R.T., 1998. *Introduction to Materials Management*. Upper Saddle River, NJ: Prentice Hall.

Bagchi, U., Chang, Y.L., and Sullivan, R.S., 1987. Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties and a common due date. *Naval Research Logistics*, 34 (5), 739–751.

Baker, K.R. and Scudder, G.D., 1990. Sequencing with earliness and tardiness penalties - A review. *Operations Research*, 38 (1), 22–36.

Bank, J. and Werner, F., 2001. Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. *Mathematical and Computer Modelling*, 33 (4-5), 363–383.

Biskup, D. and Feldmann, M., 2001. Benchmarks for scheduling on a single machine against restrictive and unrestrictive common due dates. *Computers & Operations Research*, 28 (8), 787–801.

Cheng, T.C.E., Chen, Z.L., and Shakhlevich, N.V., 2002. Common due date assignment and scheduling with ready times. *Computers & Operations Research*, 29 (14), 1957–1967.

Davis, J.S. and Kanet, J.J., 1993. Single-machine scheduling with early and tardy completion costs. *Naval Research Logistics*, 40 (1), 85–101.

Emmons, H., 1987. Scheduling to a common due date on parallel uniform processors. *Naval Research Logistics*, 34 (6), 803–810.

Gordon, V., Proth, J.M., and Chu, C.B., 2002. A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139 (1), 1–25.

Hoogeveen, J.A. and Vandevelde, S.L., 1991. Scheduling around a small common due date. *European Journal of Operational Research*, 55 (2), 237–242.

Kanet, J.J., 1981. Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics*, 28 (4), 643–651.

Kanet, J.J. and Sridharan, V., 2000. Scheduling with inserted idle time: Problem taxonomy and literature review. *Operations Research*, 48 (1), 99–110.

Mondal, S.A. and Sen, A.K., 2001. Single machine weighted earliness-tardiness penalty problem with a common due date. *Computers & Operations Research*, 28 (7), 649–669.

Nandkeolyar, U., Ahmed, M.U., and Sundararaghavan, P.S., 1993. Dynamic single-machine-weighted absolute deviation problem - Predictive heuristics and evaluation. *International Journal of Production Research*, 31 (6), 1453–1466.

Panwalkar, S.S., Smith, M.L., and Seidmann, A., 1982. Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research*, 30 (2), 391–399.

Ronconi, D.P. and Birgin, E.G., 2011. Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. *In*: Ríos-Mercado, R. Z. and Ríos-Solís, Y. A., ed. *Just-in-Time Systems*. Springer Optimization and Its Applications, Vol. 60, Pardalos, P. M. and Du, Ding-Zhu (Series eds.), Springer, 99–114.

Ronconi, D.P. and Powell, W.B., 2010. Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming. *Journal of Scheduling*, 13 (6), 597–607.

Sakuraba, C.S., Ronconi, D.P., and Sourd, F., 2009. Scheduling in a two-machine flowshop for the minimization of the mean absolute deviation from a common due date. *Computers & Operations Research*, 36 (1), 60–72.

Sidney, J.B., 1977. Optimal single-machine scheduling with earliness and tardiness penalties. *Operations Research*, 25 (1), 62–69.

Sridharan, V. and Zhou, Z., 1996. A decision theory based scheduling procedure for single-machine weighted earliness and tardiness problems. *European Journal of Operational Research*, 94 (2), 292–301.

Valente, J.M.S. and Alves, R.A.F.S., 2003. Efficient polynomial algorithms for special cases of weighted early/tardy scheduling with release dates and a common due date. *Pesquisa Operacional*, 23, 443–456.

Valente, J.M.S. and Alves, R.A.F.S., 2007. Heuristics for the early/tardy scheduling problem with release dates. *International Journal of Production Economics*, 106 (1), 261–274.

Womack, J., Jones, D., and Roos, D., 1990. *The Machine that Changed the World*. 1st ed. New York, NY: Rawson Associates.