Journal of Scheduling **7:** 85–91, 2004.

LOWER BOUNDS ON THE APPROXIMATION RATIOS OF LEADING HEURISTICS FOR THE SINGLE-MACHINE TOTAL TARDINESS PROBLEM

FEDERICO DELLA CROCE[1,*], ANDREA GROSSO[2], AND VANGELIS TH. PASCHOS[3]

[1] *D.A.I., Politecnico di Torino, Italy*
[2] *Dipartimento di Informatica, Università di Torino, Italy*
[3] *LAMSADE, Université Paris-Dauphine, France*

ABSTRACT

The weakly **NP**-hard single-machine total tardiness scheduling problem has been extensively studied in the last decades. Various heuristics have been proposed to efficiently solve in practice a problem for which a fully polynomial time approximation scheme exists (though with complexity $O(n^7/\varepsilon)$). In this note, we show that all known constructive heuristics for the problem, namely AU, MDD, PSK, WI, COVERT, NBR, present arbitrarily bad approximation ratios. The same behavior is shown by the decomposition heuristics DEC/EDD, DEC/MDD, DEC/PSK, and DEC/WI.

KEY WORDS: total tardiness, scheduling, approximation

1. INTRODUCTION

We consider the single-machine total tardiness $1\|\sum T_j$ problem where a jobset $N = \{1, 2, \ldots, n\}$ of n jobs must be scheduled on a single machine. For each job j, we define a processing time p_j and a due date d_j. The problem calls for arranging the jobset in a sequence $S = (1, 2, \ldots, n)$ so as to minimize $T(N, S) = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \max\{C_i - d_i, 0\}$ where $C_i = \sum_{j=1}^{i} p_j$.

The $1\|\sum T_j$ problem is **NP**-hard in the ordinary sense (Du and Leung, 1990). It has been extensively studied in the literature and many exact procedures have been proposed. The state-of-the-art exact method of Szwarc, Grosso, and Croce (2001) manages to solve problems with up to 500 jobs. A fully polynomial time approximation scheme was given in Lawler (1982), though with complexity $O(n^7/\varepsilon)$. Despite the presence of a fully polynomial time approximation scheme, various heuristic procedures were proposed (a nonexhaustive list of papers include Carroll (1965), Wilkerson and Irwin (1971), Baker and Bertrand (1982), Morton, Rachamadugu, and Vepsalainen (1984), Potts and Wassenhove (1991), Holsenback and Russell (1992), Panwalkar Smith and Koulamus (1993)). The purpose of this work is to analyze the approximation ratio of the most frequently applied heuristics for the $1\|\sum T_j$ problem. Given an algorithm A computing a feasible schedule S_A, for a jobset N of $1\|\sum T_j$, we denote by $T(N, S_A)$ the total tardiness of S_A and by $r(N, S_A)$ the approximation ratio $T(N, S_A)/T(N, S^*)$ where S^* indicates the optimal solution for $1\|\sum T_j$ on N. We will use r_A to indicate the worst value of $r(N, S_A)$ over all jobsets N.

*Correspondence to: Federico Della Croce. E-mail: frederico.dellacroce@polito.it

We show in this note that, quite surprisingly, all the constructive heuristics, the basic decomposition heuristic DEC/EDD, as well as the enhanced DEC/MDD, DEC/PSK, and DEC/WI decomposition heuristics perform arbitrarily badly since the lower bounds on their corresponding approximation ratios depend at least linearly on the problem size. The paper is organized as follows: in Section 2, the theoretical background for this problem is recalled; in Section 3, the constructive and decomposition heuristics are briefly presented and their approximation ratio is discussed; finally, Section 4 concludes the paper with final remarks.

2. THEORETICAL BACKGROUND

We make use of the following notation. Given the jobset $N = \{1, 2, \ldots, n\}$ let $(1, 2, \ldots, n)$ be an SPT sequence (where $i < j$ whenever $p_i = p_j$ and $d_i \leq d_j$). Let also $([1], [2], \ldots, [n])$ be an EDD sequence (where $[i] < [j]$ whenever $d_i = d_j$ and $p_i \leq p_j$). As the cost function is a regular performance measure, we know that in the optimal solution the jobs are processed with no interruption starting from time zero. Let $p(B) = \sum_{k \in B} p_k$. Let B_j and A_j be the sets of jobs that have been shown, at any time, to precede and follow job j in an optimal sequence. Correspondingly, let e_j and l_j be the earliest and latest completion times of job j in any sequence consistent with this partial ordering. Then, $e_j = p(B_j) + p_j$ and $l_j = p(N - A_j)$. The main known theoretical properties are the following.

Property 2.1. Consider two jobs i and j, $i < j$. Then, $i \to j$ if $d_i \leq max\{d_j, e_j\}$, else $j \to i$ if $d_i + p_i > l_j$ (Emmons, 1969).

Property 2.2. (Lawler, 1977) *Let job n in SPT correspond to job $[k]$ in EDD. Then, job n can be set only in position $h \geq k$ and the jobs preceding and following k are uniquely determined as $B_n = \{[1], [2], \ldots, [k-1], [k+1], \ldots, [h]\}$ and $A_n = \{[h+1], \ldots, [n]\}$.*

Property 2.3. (Lawler, 1977; Potts and Van Wassenhove, 1982; Szwarc, 1993) *Let $C_n(h) = \sum_{j=1}^{h} p_{[j]}$ be the completion time of job n when set in position $h \geq k$. Then, job n ($[k]$) cannot be set in such position if:*

(a) $C_n(h) \geq d_{[h+1]}, \quad h < n;$
(b) $C_n(h) < d_{[h]} + p_{[h]}, \quad h > k;$
(c) $C_n(h) \leq d_{[r]} + p_{[r]}, \quad \text{for some } r = k, \ldots, h-1.$

By exploiting Property 2.2, Lawler (1977) proposed a pseudo-polynomial dynamic programming algorithm running with complexity $O(n^4 \sum p_i)$. Also, by means of scaling techniques, he derived a fully polynomial time approximation scheme running with complexity $O(n^7/\varepsilon)$ (Lawler, 1982). Further recent improved dominance and decomposition results (Chang et al., 1995; Szwarc and Mukhopadhyay, 1996; Croce et al., 1998) are not mentioned here as they were not used in the considered heuristics.

3. APPROXIMATION RESULTS

The following lemma shows that the upper bound proposed in Lawler (1977) for the approximation ratio of the EDD sequence can be attained.

Lemma 3.1. $r_{\text{EDD}} \leq n$ *and this bound is tight.*

Proof. For the upper bound we recall here the proof of Lawler (1977). Consider a jobset N and an EDD sequence S_{EDD} on N. Denote by $T_{\max}(N, S_{\text{EDD}})$ the value of the maximum tardiness of S_{EDD} and by S^* an optimal solution of $1\|\sum T_j$ on N. Notice that that $T(N, S_{\text{EDD}}) = 0$ when $T(N, S^*) = 0$ and as the EDD rule minimizes the $1\|T_{\max}$ problem, we have $T_{\max}(N, S_{\text{EDD}}) \leq T_{\max}(N, S^*) \leq T(N, S^*)$. But then, $T(N, S_{\text{EDD}}) \leq nT_{\max}(N, S_{\text{EDD}}) \leq nT(N, S^*)$.

In order to prove the tightness of the ratio above, consider the following example (E_1): $N = \{1, 2, \ldots, n\}$, $p_1 = m$, $p_2, \ldots, p_n = 1$, $d_1 = 0$, $d_2, \ldots, d_n = \varepsilon$. The optimal sequence is $S^* = (2, \ldots, n, 1)$, with $T(N, S^*) = n(n+1)/2 + m - 1 - (n-1)\varepsilon$. The EDD rule produces sequence $S_{\text{EDD}} = (1, 2, \ldots, n)$, where $T_1 = m$, $T_i = m + i - 1 - \varepsilon$ for $i = 2, \ldots, n$. Thus, $T(N, S_{\text{EDD}}) = nm + n(n-1)/2 - (n-1)\varepsilon$. Hence, for m large enough and ε small enough, we have $r(N, S_{\text{EDD}}) \approx n$. ■

3.1. Constructive heuristics

This subsection deals with approximation ratios for constructive heuristics. Quick dispatching rules as well as simple greedy algorithms are grouped in this class. Below are indicated and briefly exposed the main constructive heuristics proposed for the $1\|\sum T_j$ problem. For conciseness, only one-shot procedures are fully described while, for the other procedures, the relevant references are indicated for details.

AU: at time t, schedule i before j if $u_i > u_j$, where $u_i = \exp[-\max\{d_i - t - p_i, 0\}/k\bar{p}]/p_i$ and $\bar{p} = \sum_{i=1}^{n} p_i/n$. This heuristic, specifically developed like COVERT for the more general weighted tardiness problem $1\|\sum w_j T_j$, does not take into account Property 2.1 (Morton, Rachamadugu, and Vepsalainen, 1984).

MDD: at time t, schedule i before j if $\max\{t + p_i, d_i\} < \max\{t + p_j, d_j\}$, or $\max\{t + p_i, d_i\} = \max\{t + p_j, d_j\}$ and $p_i < p_j$ (Baker and Bertrand, 1982).

PSK: start with an SPT sequence and scan the jobs in that order, searching for the best job to be placed in the first unscheduled position; once that position is filled, the next position is considered and the process is iterated until all jobs have been sequenced; we refer here to the description of the algorithm in (Panwalker, Smith, and Koulamas, 1993).

WI: (can be seen as a hybrid construction/local search heuristic) use adjacent job pairwise interchanges in the process of building the schedule; we refer here to the description of the algorithm in Wilkerson and Irwin (1971).

COVERT: given a partial sequence S, place one job at a time among the remaining unscheduled jobs according to the following priority index PI_j (E denotes the set of unscheduled jobs that have no unscheduled predecessors according to Property 2.1) (Carroll, 1965):

$$\text{PI}_j = \begin{cases} 1 & d_j \leq p(S) + p_j \\ \dfrac{p(S \cup E) - d_j}{p(E) - p_j} & p(S) + p_j < d_j < p(S \cup E) \\ 0 & p(S \cup E) \leq d_j; \end{cases}$$

the job selected is the one with largest PI_j/p_j ratio. This heuristic was designed for the more general $1\|\sum w_jT_j$ problem.

NBR: start with an EDD schedule and check whether a job should be relocated by means of a dominance rule based on Property 2.1 used in combination with the Net Benefit of job Relocation; we refer here to the description of the algorithm in Holsenback and Russell (1992).

Note that, $T(N, S_{\mathrm{PSK}}) = T(N, S_{\mathrm{WI}}) = T(N, S_{\mathrm{COVERT}}) = T(N, S_{\mathrm{NBR}}) = T(N, S_{\mathrm{EDD}}) = 0$ when $T(N, S^*) = 0$. As far as AU is concerned, the following proposition holds.

Proposition 3.1. $r_{\mathrm{AU}} \to \infty$.

Proof. Consider the following two-job example denoted by E_2: $N = \{1, 2\}$, $p_1 = 1$, $p_2 = \varepsilon$, $d_1 = 1$, $d_2 = 1 + \varepsilon$. Then, at $t = 0$: $u_1 = 1$, and $u_2 = \exp[-2/k(1+\varepsilon)]/\varepsilon > \exp(-2/k)/\varepsilon$. For any given value of the parameter k, setting $0 < \varepsilon < \exp(-2/k)$ yields $u_2 > 1$. Hence AU will schedule job 2 first and job 1 last: we get $S_{\mathrm{AU}} = (1, 2)$ and $T(N, S_{\mathrm{AU}}) = \varepsilon > 0$. The optimal sequence is $S^* = (1, 2)$ with $T(N, S^*) = 0$ and AU gives an infinite relative error. ■

Proposition 3.2. $r_{\mathrm{MDD}} = r_{\mathrm{PSK}} = r_{\mathrm{WI}} = r_{\mathrm{COVERT}} \geq n/2$.

Proof. Consider the following example denoted by E_3 in what follows: $N = \{1, 2, \ldots, n+1\}$, $p_1 = n$, $p_2, \ldots, p_{n+1} = 1$, $d_1 = n$, $d_2, \ldots, d_{n+1} = n + \varepsilon$.

The MDD rule selects at time $t = 0$ job 1 to be scheduled in first position. All the other (identical) jobs will then follow. Hence an EDD sequence $S = (1, \ldots, n+1)$ is generated where $T_1 = 0$, $T_i = i - 1 - \varepsilon$, for $i = 2, \ldots, n+1$. Thus, $T(N, S_{\mathrm{MDD}}) = n(n+1)/2 - n\varepsilon$. The optimal sequence is $S^* = (2, \ldots, n+1, 1)$, with $T(N, S^*) = n$. Hence, for ε small enough, $r(N, S_{\mathrm{MDD}}) \approx n/2$.

As pointed out in Alidaee and Gopalan (1997) and Cheng (1992), procedures PSK and WI are basically equivalent to the MDD rule for all those instances such as example E_3 where there are no couples of jobs i and j with $p_i \neq p_j$ or $d_i \neq d_j$ such that $\max\{t + p_i, d_i\} = \max\{t + p_j, d_j\}$. Indeed, both PSK and WI reach the same result as MDD in example E_3.

Finally, with respect to COVERT, notice that Property 2.1 implies $2 \to 3 \to \cdots \to n+1$, whereas job 1 is not involved in precedence relations. Also, notice that $\mathrm{PI}_1 = 1$. Consider the first stage, where $S = \emptyset$, $E = \{1, 2\}$. Then $p(S) = 0$, $p(E) = n + 1$, and $\mathrm{PI}_2 = (1 - \varepsilon)/n$. Hence job 1 is scheduled first, yielding the same sequence $S_{\mathrm{COVERT}} = (1, 2, \ldots, n+1)$, and this completes the proof of the proposition. ■

Proposition 3.3. $r_{\mathrm{NBR}} \geq n/6$.

Proof. Consider the following example denoted by E_4: set $n = 2m + 2$ and consider $N = \{1, 2, \ldots, 2m+2\}$, $p_1 = m$, $p_2 = 1$, $p_3 = \cdots = p_{m+1} = \varepsilon$, $p_{m+2} = \cdots = p_{2m+1} = 1$, $p_{2m+2} = 2\varepsilon$, $d_1 = m$, $d_2 = m + (\varepsilon/2)$, $d_i = m + (i-2)\varepsilon$, for $i = 3, \ldots, m+1$, $d_{m+2} = m + 1 + (m-1)\varepsilon$, $d_j = j + (m-1)\varepsilon$, for $j = m+3, \ldots, 2m$, $d_{2m+1} = d_{2m+2} = 2m + (m-1)\varepsilon$.

The NBR algorithm considers, at any stage, a sequence $\bar{S}$ and a set of jobs $i_1 < i_2 < \cdots < i_k$ in $\bar{S}$ such that $T_{i_k} < p_{i_k}$ and $p_{i_1} > p_{i_2} > \cdots > p_{i_k}$. A job is selected among $i_1, \ldots, i_{k-1}$ in order to be moved just after i_k, so that the decrease in tardiness is maximum (see Holsenback and Russell

(1992) for details). Notice that, for example E_4, NBR will execute one single stage, considering jobs 1, $2m+1$, $2m+2$, and moving job 1 to the last position with tardiness $(m+1)+(m+1)\varepsilon$. The sequence induced by NBR is then $S_{\mathrm{NBR}} = (2, 3, \ldots, 2m+2, 1)$ with $T(N, S_{\mathrm{NBR}}) = (m+1)+(m+1)\varepsilon$. The optimal sequence is $S^* = (2, 3, \ldots, m+1, 1, m+2, m+3, \ldots, 2m+2)$ with $T(N, S^*) = 3+(m+1)\varepsilon$. Recalling that $n = 2m+2$, we have $r(N, S_{\mathrm{NBR}}) = ((m+1)+(m+1)\varepsilon)/(3+2\varepsilon) \geq m/3 \approx n/6$. ■

3.2. *Decomposition heuristics*

Decomposition heuristics were proposed in Potts and Wassenhove (1991) and Koulamas (1994) to which we refer for details. Below are sketched the main decomposition heuristics applied to the $1\|\sum T_j$ problem.

DEC/EDD: exploits Properties 2.2 and 2.3; when more than one position k can be occupied by the largest processing time job, the EDD rule is used to solve the two subproblems generated for each value of k; the largest processing time job is then placed in the position inducing the best cost function value.

DEC/(MDD-PSK-WI): as above but applies MDD (or PSK or WI) instead of EDD.

Consider any decomposition heuristic *DEC/H*. Notice that $T(N, S_{\mathrm{DEC/H}}) = 0$ when $T(N, S^*) = 0$. The following proposition generalizes the result given in Yu (1996) for DEC/EDD.

Lemma 3.2. Let $r_H \leq f(n)$ where f is a strictly monotone-increasing function with $f(n) > 1$ and for any natural numbers p, q, $f(p)+f(q) \leq f(p+q)$. Then $r_{\mathrm{DEC/H}} \leq f(n-1)$.

Proof. Let $S_{\mathrm{DEC/H}}$ be the sequence produced by DEC/H. At the first stage, DEC/H will consider the job with largest processing time n, and will try all nondominated positions for it. Let $S^* = (S_1^*, n, S_2^*)$ be an optimal sequence, where S_1^* and S_2^* are optimal sequences for the job subsets N_1, N_2 given by the optimal decomposition. Let T_n be the (optimal) tardiness of job n in such a sequence. Then, by the position selection rule in the decomposition heuristics:

$$\begin{aligned} T(N, S_{\mathrm{DEC/H}}) &\leq T(N_1, S_{1,\mathrm{DEC/H}}) + T(N_2, S_{2,\mathrm{DEC/H}}) + T_n \\ &\leq f(|N_1|) \cdot T(N_1, S_1^*) + f(|N_2|) \cdot T(N_2, S_2^*) + T_n \\ &\leq f(|N_1|) + f(|N_2|) \cdot \left[T(N_1, S_1^*) + T(N_2, S_2^*)\right] + T_n \\ &\leq f(n-1) \cdot \left[T(N_1, S_1^*) + T(N_2, S_2^*)\right] + T_n \\ &\leq f(n-1) \cdot T(N, S^*). \end{aligned}$$

■

Remark 3.1. The above lemma provides $r_{\mathrm{DEC/EDD}} \leq n-1$ (derived in Yu (1996)) as a corollary, but it also implies for DEC/DEC/EDD (call it DEC^2/EDD) that $r_{\mathrm{DEC^2/EDD}} \leq n-2$ and, more generally, $r_{\mathrm{DEC^k/EDD}} \leq n-k$.

To the authors' knowledge no tight bounds are available for the decomposition heuristics. The following propositions provide lower bounds on the worst-case approximation ratio of the main decomposition heuristics.

Proposition 3.4. $r_{\text{DEC/EDD}} \geq n/2$.

Proof. Consider the following example denoted by E_5: set $n = 2m+1$ and consider $N = \{1, 2, \ldots, 2m+1\}$, $p_1 = m - \varepsilon$, $p_2 = \cdots = p_{m+1} = 1$, $p_{m+2} = \cdots = p_{2m+1} = \varepsilon$, $d_1 = m$, $d_i = m + i - 1$, for $i = 2, \ldots, m$, $d_{m+1} = 2m - 1$, $d_i = 2m - 1, + \varepsilon$, for $i = m+2, \ldots, 2m+1$.

We have $S_{\text{EDD}} = (1, \ldots, 2m+1)$. Job 1 is the largest processing time job. The completion time $C_1(r)$ for job 1 in position r is

$$C_1(r) = \begin{cases} m + r - 1 - \varepsilon & r = 1, \ \ldots, m+1 \\ 2m + (r - m - 2)\varepsilon & r = m+2, \ \ldots, 2m+1. \end{cases}$$

Positions $r = 2, \ldots, m+1$ are eliminated by $d_r + p_r > C_1(r)$; positions $r = m+2, \ldots, 2m$ are eliminated by $C_1(r) > d_{r+1}$. Job 1 can thus be placed in positions 1 and $2m+1$ inducing sequences $\mu = (1, \ldots, 2m+1)$ and $\theta = (2, \ldots, 2m+1, 1)$. We get $T(N, \mu) = (1 - \varepsilon) + \varepsilon m(m-1)/2 - (m-2)\varepsilon + (m+1)$ and $T(N, \theta) = m + (m-2)\varepsilon$. Hence, $T(N, \mu) > T(N, \theta)$ and position $2m+1$ is selected for job 1 with tardiness $m + (m-2)\varepsilon$. Then, sequence $(2, \ldots, 2m)$ is entirely early and the jobs can be scheduled as they are. So, $S_{\text{DEC/EDD}} = \theta$. Besides, the optimal sequence is $S^* = (1, m+2, \ldots, 2m+1, \ 2, \ldots, m+1)$ with $T(N, S^*) = 1 + m(m-1)\varepsilon$. Hence for ε small enough, $r(N, S_{\text{DEC/EDD}}) = [m + (m-2)\varepsilon)/(1 + m(m-1)\varepsilon] \approx m \approx n/2$. ■

Remark 3.2. Notice that, applying a slightly worse implementation of DEC/EDD where property 1(c) is neglected, makes, as shown in Yu (1996), the $n-1$ bound tight for this decomposition heuristic. Ironically, taking into account such a property reopens the gap since the tight example presented in Yu (1996) is no longer valid.

Proposition 3.5. $r_{\text{DEC/MDD}} = r_{\text{DEC/PSK}} = r_{\text{DEC/WI}} \geq n/3$.

Proof. Consider the following example E_6: set $n = 3m/2$ and assume $N = \{1, 2, \ldots, 3m/2\}$, with $p_1 = m^2$, $p_i = 2m$, for $i = 2, \ldots, m/2$, $p_j = 2$, for $j = (m = 2) + 1, \ldots, 3m/2$, $d_1 = m^2$, $d_i = m^2 + 2(i-1)m$, for $i = 2, \ldots, (m/2) - 1$, $d_{m/2} = 2m^2 - 2m - \varepsilon$, $d_j = 2m^2 - 2m + \varepsilon$, for $j = (m/2) + 1, \ldots, 3m/2$.

Job 1 has the largest processing time and can be placed in positions 1 and $3m/2$ only. If it is placed in position 1, its tardiness is 0 and the MDD rule induces sequence $(1, 2, \ldots, 3m/2)$ with value $m(m+1) - (m-1)\varepsilon = m^2 + m - (m-1)\varepsilon$. If, on the other hand, it is placed in the last position, its tardiness is m^2 but all the other jobs are early and the total tardiness remains m^2. Hence, for ε small enough, the last position is selected. In all, $S_{\text{DEC/MDD}} = (2, \ldots, 3m/2, 1)$ and $T(N, S_{\text{DEC/MDD}}) = m^2$. However, the optimal sequence for this example is $S^* = (1, 2, \ldots, (m/2) - 1, (m/2) + 1, \ldots, 3m/2, m/2)$. with value $2m + \varepsilon$. Recalling that $n = 3m/2$, we have $r(N, S_{\text{DEC/MDD}}) = m^2/(2m + \varepsilon) \approx m/2 = n/3$. Analogously to Proposition 2, we have that both DEC/PSK and DEC/WI reach the same result as DEC/MDD in example E_7, and this completes the proof of the proposition. ■

4. CONCLUSION

In this note we have discussed bounds for the approximation ratios of the leading heuristics for the $1\|\sum T_j$ problem. Though no tight bounds have been derived (except for the simple EDD rule), we have shown that there is no hope of constant approximation ratio for all the known constructive heuristics, nor for the decomposition ones. For the decomposition heuristics, even if the lower bounds obtained are slightly better than the corresponding bounds of the former category, their approximation ratios depend linearly on the problem size. Notice that, as far as examples E_5 and E_6 are concerned, the same bounds are obtained even if the improved elimination rules of Chang et al. (1995) and the double decomposition scheme of Della Croce et al. (1998) are embedded in the considered decomposition heuristics. Finally, we remark that in Yu (1996) an internal report in Chinese (that we could not manage to obtain) is cited where apparently a $n/2$ tight bound is proved for WI: should one take into account this result, then, by means of Lemma 3.2, a $(n-1)/2$ upper bound can be immediately derived for DEC/WI. Also, it should be possible to reach the same upper bound for DEC/MDD and DEC/PSK due to the results presented in Alidaee and Gopalan (1997) and Cheng (1992), respectively.

REFERENCES

Alidaee, B. and S. Gopalan, "A note on the equivalence of two heuristics to minimize total tardiness," *Eur. J. Oper. Res.*, **96**, 514–517 (1997).

Baker, K. R. and J. W. M. Bertrand, "A dynamic priority rule for scheduling against due dates," *J. Oper. Manage.*, **3**, 37–42 (1982).

Carroll, D. C., "Heuristic sequencing of single and multiple components," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1965.

Chang, S., Q. Lu, G. Tang, and W. Yu, "On decomposition of the total tardiness problem," *Oper. Res. Lett.*, **17**, 221–229 (1995).

Cheng, T. C., "A note on the equivalence of the Wilkerson–Irwin and modified due-date rules for the mean tardiness sequencing problem," *Comput. Ind. Eng.*, **22**, 63–66 (1992).

Della Croce, F., R. Tadei, P. Baracco, and A. Grosso, "A new decomposition approach for the single machine total tardiness scheduling problem," *J. Oper. Res. Soc.*, **49**, 1101–1106 (1998).

Du, J. and J. Y. T. Leung, "Minimizing total tardiness on one machine is NP-hard," *Math. Oper. Res.*, **15**, 483–495 (1990).

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

Holsenback, J. E. and R. M. Russell, "A heuristic algorithm for sequencing on one machine to minimize total tardiness," *J. Oper. Res. Soc.*, **43**, 53–62 (1992).

Koulamas, C. P., "The total tardiness problem: Review and extensions," *Oper. Res.*, **42**, 1025–1041 (1994).

Lawler, E. L., "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness," *Ann. Discrete Math.*, **1**, 331–342 (1977).

Lawler, E. L., "A fully polynomial approximation scheme for the total tardiness problem," *Oper. Res. Lett.*, **1**, 207–208 (1982).

Morton, T. E., R. M. Rachamadugu, and A. Vepsalainen, "Accurate myopic heuristics for tardiness scheduling," *GSIA Working Paper 36-83-84*, Carnegie Mellon University, PA, 1984.

Panwalkar, S. S., M. L. Smith, and C. P. Koulamas, "A heuristic for the single machine tardiness problem," *Eur. J. Oper. Res.*, **70**, 304–310 (1993).

Potts, C. N. and L. N. Van Wassenhove, "A decomposition algorithm for the single machine total tardiness problem," *Oper. Res. Lett.*, **5**, 177–181 (1982).

Potts, C. N. and L. N. Van Wassenhove, "Single machine tardiness sequencing heuristics," *IIE Trans.*, **23**, 93–108 (1991).

Szwarc, W., "Single machine total tardiness problem revisited," in Y. Ijiri (ed.), *Creative and Innovative Approaches to the Science of Management*, Quorum Books, Westport, CT, pp. 407–419, 1993.

Szwarc, W. and S. Mukhopadhyay, "Decomposition of the single machine total tardiness problem," *Oper. Res. Lett.*, **19**, 243–250 (1996).

Szwarc, W., A. Grosso, and F. Della Croce, "Algorithmic paradoxes of the single machine total tardiness problem," *J. Sched.*, **4**, 93–104 (2001).

Wilkerson, L. J. and J. D. Irwin, "An improved algorithm for scheduling independent tasks," *AIIE Trans.*, **3**, 239–245 (1971).

Yu, W., "The two-machines flow shop problem with delays and the one-machine total tardiness problem," Ph.D. dissertation, Eindhoven University of Technology, The Netherlands, 1996.