UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS UMR 7024

CAHIER DU LAMSADE

193

Avril 2002

Lower bounds on the approximation ratios of leading heuristics for the single machine total tardiness problem

F. Della Croce, A. Grosso, V. Th. Paschos

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Table of Contents

Des bornes inférieures pour le rapport d'approximation pour les principales heuristiques du problème du retard total d'ordonnancement sur une seule machine (*single machine total scheduling tardiness problem*)

Résumé

Le problème du retard total d'ordonnancement sur une seule machine (*single machine total scheduling tardiness problem*) est faiblement **NP**-hard ; il a été largement étudié durant les dernières décennies. Plusieurs méthodes heuristiques ont été proposées qui résolvent assez bien ce problème en pratique. Par aileurs, il peut être aussi résolu par un schéma d'approximation complétement polynomial (avec complexité d'$O(n^7/\epsilon)$, $\forall\epsilon > 0$). Dans cette note, nous démontrons que toutes les heuristiques constructives connues comme MDD, PSK, WI, COVERT, AU, NBR ont un rapport d'approximation au pire des cas arbitrairement mauvais. Le même comportement est observé par les heuristique de décomposition DEC/EDD, DEC/MDD, DEC/PSK et DEC/WI.

Mots-clé : ordonnancement, retard total, algorithme d'approximation, rapport d'approximation, problème **NP**-complet.

Lower bounds on the approximation ratios of leading heuristics for the single machine total tardiness problem

Abstract

The weakly **NP**-hard single machine total scheduling tardiness 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 MDD, PSK, WI, COVERT, AU, NBR, present arbitrarily bad approximation ratio. Same behavior is shown by the decomposition heuristics DEC/EDD, DEC/MDD, DEC/PSK and DEC/WI.

Keywords: scheduling, total tardiness, approximation algorithm, approximation ratio, **NP**-complete problem.

1 Introduction

We consider the one-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 $S^* = (1, 2, \ldots, n)$ such that $T(N, S^*) = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \max\{C_i - d_i, 0\}$, where $C_i = \sum_{i=1}^{n} p_i$ is minimum.

The $1||\sum T_j$ problem is **NP**-hard in the ordinary sense ([1]). It has been extensively studied in the literature and many exact procedures were proposed. The state-of-the-art exact method of [2] manages to solve problems with up to 500 jobs. A fully polynomial time approximation scheme was given in [3], though with huge complexity $O(n^7/\varepsilon)$. Despite the presence of a fully polynomial time approximation scheme, various heuristic procedures were proposed (a non exhaustive list of papers include [4, 5, 6, 7, 8, 9, 10]). The purpose of this work is to analyze the approximation ratio of the most applied heuristics for the $1||\sum T_j$ problem. Given an algorithm A computing a feasible schedule S, for a jobset N of $1||\sum T_j$, we denote by $T_{\mathrm{A}}(N, S)$ the total tardiness of S and by $r_{\mathrm{A}}(N, S)$ the approximation ratio $T_{\mathrm{A}}(N, S)/T(N, S^*)$ where S^* indicates the optimal solution for $1||\sum T_j$ on N. We will use r_{A} to indicate the worst value of $r_{\mathrm{A}}(N, S)$ over all jobsets N.

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 bad 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 $d_i \leqslant d_j$). Let also $([1], [2], \ldots, [n])$ be an EDD sequence (where $[i] < [j]$ whenever $d_i = d_j$ and $p_i \leqslant p_j$). 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 this sequence. Then, $e_j = p(B_j) + p_j$ and $l_j = p(N - A_j)$. The main known theoretical properties are the following.

Property 1. *([11]) Consider two jobs i and j, $i < j$. Then, $i \to j$ if $d_i \leqslant \max\{d_j, e_j\}$, else $j \to i$ if $d_i + p_i > l_j$.*

Property 2. *([12]) Let job n in SPT corresponds to job $[k]$ in EDD. Then, job n can be set only in position $h \geqslant k$ 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 3. *([12, 13, 14]) Let $C_n(h) = \sum_{j=1}^{h} p_{[j]}$. Then, job n ($[k]$) cannot be set in position $h \geqslant k$ if:*

$C_n(h) \geqslant d_{[h+1]}$, *$h < n$;*

$C_n(h) < d_{[h]} + p_{[h]}$, *$h > k$;*

$C_n(h) \leqslant d_{[r]} + p_{[r]}$, *for some $r = k, \ldots, h-1$.*

By exploiting Property 2, Lawler proposed in [12] a pseudo-polynomial dynamic programming algorithm running with complexity $O(n^4 \sum p_i)$. Also, by means of scaling techniques, he derived in [3] a fully polynomial time approximation scheme running with complexity $O(n^7/\varepsilon)$. Further recent improved dominance and decomposition results ([15, 16, 17]) are not mentioned here as they were not used in the considered heuristics.

3 Approximation results

The following lemma provides an upper bound for the approximation ratio of the EDD sequence. It limits it to n and shows that it can be attained.

Lemma 1. *$r_{\text{EDD}} \leqslant n$ and this bound is tight.*

Proof. 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. As the EDD rule minimizes the $1||T_{\max}$ problem, we have $T_{\max}(N, S_{\text{EDD}}) \leqslant T_{\max}(N, S^*) \leqslant T(N, S^*)$. But then, $T(N, S_{\text{EDD}}) \leqslant nT_{\max}(N, S_{\text{EDD}}) \leqslant nT(N, S^*)$, and the upper bound claimed is proved.

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 $r_{\text{EDD}}(N, S_{\text{EDD}}) \approx n$. □

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 sake of conciseness, only one-shot procedures are fully described while, for the other procedures, the relevant references are indicated for details.

MDD ([4]): 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 $p_i < p_j$.

PSK: starts with a SPT sequence and scans in that order the jobs 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 [8].

WI: can be seen as a hybrid construction/local search heuristic because it uses adjacent job pairwise interchanges in the process of building the schedule; we refer here to the description of the algorithm in [10].

COVERT ([5]): given a partial sequence S, places 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 1):

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

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

AU ([7]): 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 1.

NBR: starts with an EDD schedule and checks whether a job should be relocated by means of a dominance rule based on Property 1 used in combination with the Net Benefit of job Relocation; we refer here to the description of the algorithm in [6].

Proposition 1. $r_{\mathrm{MDD}} = r_{\mathrm{PSK}} = r_{\mathrm{WI}} = r_{\mathrm{COVERT}} \geqslant n/2$.

Proof. Consider the following example denoted by E_2 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, $T_{\mathrm{MDD}}(N, S) = n(n+1)/2 - n\varepsilon$. The optimal sequence is $S^* = (2, \ldots, n+1, 1)$, with $T(N, S^*) = n$. Hence, for ε small enough, $r_{\mathrm{MDD}}(N, S) \approx n/2$.

As pointed out in [18] and [19], procedures PSK and WI are basically equivalent to the MDD rule for all those instances such as example E_2 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_2.

Finally, with respect to COVERT, notice that Property 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 2. $r_{\mathrm{AU}} \geqslant n^k$ *for any constant* $k \geqslant 1$.

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

Notice that $\bar{p} = (n + n\varepsilon)/n \approx 1$, for large n. We have $S^* = (2, \ldots, n, n+1, 1)$ and $T(N, S^*) = n\varepsilon$. Applying AU, at time $t = 0$ we get: $u_1 = 1/n$ and, for $i = 2, \ldots, n$, $u_i = (1/\varepsilon)(\exp[-(n-1-\varepsilon)/(k\bar{p})])$. For $\varepsilon = n^{-k}$ and n large enough, we have $u_i < u_1$ and job 1 is scheduled first. The other (identical) jobs are sequenced next. The resulting sequence is $S = (1, 2, \ldots, n)$, with $T(N, S) = n - n\varepsilon$, thus giving $r_{\mathrm{AU}}(N, S_{\mathrm{AU}}) \approx 1/\varepsilon = n^k$. □

Proposition 3. $r_{\mathrm{NBR}} \geqslant n/6$.

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 [6] 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 $r_{\text{NBR}}(N, S_{\text{NBR}}) = ((m+1) + (m+1)\varepsilon)/(3+2\varepsilon) \geqslant m/3 \approx n/6$. □

3.2 Decomposition heuristics

Decomposition heuristics were proposed in [9, 20] 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 and 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.

Proposition 4. $r_{\text{DEC/EDD}} \geqslant n/2$.

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, $r_{\text{DEC/EDD}}(N, S_{\text{DEC/EDD}}) = (m + (m-2)\varepsilon)/(1 + m(m-1)\varepsilon) \approx m \approx n/2$. □

Proposition 5. $r_{\text{DEC/MDD}} = r_{\text{DEC/PSK}} = r_{\text{DEC/WI}} \geqslant n/3$.

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 $r_{\text{DEC/MDD}}(N, S) = m^2/(2m+\varepsilon) \approx m/2 = n/3$. Analogously to Proposition 1, 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 Conclusions

In this note we have provided lower bounds for the approximation ratios of the leading heuristics for the $1||\sum T_j$ problem. Though no upper bounds have been simultaneously derived, we have shown that there is no hope of constant approximation ratio for all known constructive heuristics, nor for the decomposition ones. For this latter category, 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 [15] and the double decomposition scheme of [16] are embedded in the considered decomposition heuristics.

References

[1] J. Du and J. Y. T. Leung (1990), "Minimizing total tardiness on one machine is NP–hard", *Mathematics of Operations Research* 15, 483–495.

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

[3] E. L. Lawler (1982), "A fully polynomial approximation scheme for the total tardiness problem", *Operations Research Letters* 1, 207–208.

[4] K. R. Baker and J. W. M. Bertrand (1982), "A dynamic priority rule for scheduling against due dates", *Journal of Operations Management* 3, 37–42.

[5] D. C. Carroll (1965), "Heuristic sequencing of single and multiple components", Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

[6] J. E. Holsenback and R. M. Russell (1992), "A heuristic algorithm for sequencing on one machine to minimize total tardiness", *Journal of the Operational Research Society* 43, 53–62.

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

[8] S. S. Panwalkar, M. L. Smith and C. P. Koulamas (1993), "A heuristic for the single machine tardiness problem", *European Journal of Operational Research* 70, 304–310.

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

[10] L. J. Wilkerson and J.D. Irwin (1971), "An improved algorithm for scheduling independent tasks", *AIIE Transactions*é3, 239–245.

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

[12] E. L. Lawler (1977), "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness", *Annals of Discrete Mathematics* 1, 331–342.

[13] C. N. Potts and L. N. Van Wassenhove (1982), "A decomposition algorithm for the single machine total tardiness problem", *Operations Research Letters* 5, 177–181.

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

[15] S. Chang, Q. Lu, G. Tang and W. Yu (1995), “On decomposition of the total tardiness problem”, *Operations Research Letters* 17, 221–229.

[16] F. Della Croce, R. Tadei, P. Baracco and A. Grosso (1998), “A new decomposition approach for the single machine total tardiness scheduling problem”, *Journal of the Operational Research Society* 49, 1101–1106.

[17] W. Szwarc and S. Mukhopadhyay (1996), “Decomposition of the single machine total tardiness problem”, *Operations research Letters* 19, 243–250.

[18] B. Alidaee and S. Gopalan (1997), “A note on the equivalence of two heuristics to minimize total tardiness”, *European Journal of Operational Research* 96, 514–517.

[19] T. C. Cheng (1992), “A note on the equivalence of the Wilkerson-Irwin and Modified Due-Date rules for the mean tardiness sequencing problem”, *Computers and Industrial Engineering* 22, 63–66.

[20] C. P. Koulamas (1994), “The total tardiness problem: review and extensions”, *Operations Research* 42, 1025–1041.