Journal of the Operations Research
Society of Japan
Vol. 39, No. 3, September 1996

MINIMIZING TOTAL TARDINESS FOR SINGLE MACHINE SEQUENCING

Tsung-Chyan Lai Yuh-Kwo Kuo
National Taiwan University

(Received July 18, 1994; Final January 18, 1996)

Abstract We consider the single machine sequencing problem in which each job has a processing time and a due date. The objective is to find a sequence of n jobs which minimizes the sum of the tardiness of each job. We present an $O(n \log n)$ MDD (Modified Due Date) rule which satisfies local optimality and show that the MDD rule has a worst-case performance ratio of $\frac{n}{2}$. The MDD rule is superior to other known $O(n \log n)$ heuristics in the sense of worst-case performance.

1. Introduction

We consider the single machine sequencing problem in which each of the n jobs i has a processing time p_i on the machine and a due date d_i. The objective is to minimize the total tardiness of the n jobs. The tardiness T_i of job i is defined as $\max\{0,\ C_i - d_i\}$, where C_i is the completion time of job i. The problem is denoted by $1//\sum T_i$ (Lawler et al. [8]). The complexity of the problem had remained unknown for over one decade until recently Du and Leung [3] showed that the problem is NP-hard.

Branch and bound optimal algorithms [4, 6, 12, 14, 16] have been proposed by taking advantage of dominace rules [5] and tight lower bounds. Dynamic programming algorithms [1, 9, 13, 15, 17] are also available. Lawler [9] presented an $O(n^4 \sum p_i)$ or $O(n^5 \max_i\{p_i\})$ pseudopolynomial dynamic programming algorithm based on a decomposition theorem that the job, say k, with the largest processing time in a sequence in nondecreasing order of due dates can be placed only after the k-th position in an optimal sequence. The enumerative algorithm of Potts and Wassenhove [13] also uses the decomposition theorem to partition the problem into subproblems, each of which is solved by the dynamic programming of Schrage and Baker [15] when storage capacity permits. Based on the pseudopolynomial dynamic programming algorithm and data scaling, Lawler [10] further presented an $O(\frac{n^7}{\epsilon})$ fully polynomial approximation scheme, where ϵ is the allowable deviation.

In this paper, we present an $O(n \log n)$ MDD (Modified Due Date) rule which satisfies local optimality. 'Local optimality' means the optimality of two adjacent jobs in a given sequence, assuming that the other jobs are fixed. We show that the MDD rule has a worst-case performance ratio of $\frac{n}{2}$. The MDD rule is superior to other known $O(n \log n)$ heuristics in the sense of worst-case performance. This paper is organized as follows. We derive, in section 2, the worst-case performance ratio for the MDD rule and conclude with section 3.

2. Analysis of the MDD Rule

In this section we present our main result on deriving the worst-case performance ratio of the MDD rule. Given a problem instance I, we let $T^H(I)$ represent the total tardiness generated by heuristic H and $T^*(I)$ denote the minimum total tardiness. A worst-case performance ratio associated with heuristic H, denoted by η^H, can be defined as follows:

$$\eta^H \equiv \sup_I \{\frac{T^H(I)}{T^*(I)}\} \qquad \text{for all problem instances } I.$$

Let $\sigma^H(I)$ denote the sequence generated by heuristic H and $\sigma^*(I)$ denote the optimal sequence corresponding to a problem instance I. Also, we let $T_j^H(I)$ denote the tardiness of job j in $\sigma^H(I)$ and $T_j^*(I)$ the tardiness of job j in $\sigma^*(I)$. In what follows, we will suppress the problem instance I and use T^H, T^*, σ^H, σ^*, T_j^H and T_j^* whenever no confusion arises.

It is well-known that the SPT (Shortest Processing Time) rule, which sequences the jobs in nondecreasing order of p_i, generates an optimal sequence if all the due dates are equal or if none of the n jobs can meet their due dates. The EDD (Earliest Due Date) rule, which sequences the jobs in nondecreasing order of d_i, generates an optimal sequence if all the processing times are equal or if at most one job is tardy in the sequence generated by the EDD rule.

If the criterion is minimization of maximum tardiness, the EDD rule (Jackson [7]) generates an optimal sequence. If the criterion is minimization of the number of tardy jobs, Moore's algorithm [11, 18] generates an optimal sequence. Moore's algorithm adopts the sequence generated by the EDD rule as the initial sequence. In each iteration, Moore's algorithm scans sequentially from the first position until a tardy job is found, and discards the job with the largest processing time among the scanned jobs. The process is repeated until no tardy jobs have been found. The final sequence is formed by appending all the discarded jobs in an arbitrary order to the undiscarded group in nondecreasing order of due dates. It is easy to show that the EDD rule has a worst-case performance ratio of n_1, where n_1 is the number of tardy jobs in σ^{EDD}, and Moore's algorithm (denoted by M) and the SPT rule are both arbitrarily bad.

We now present the MDD rule, which can be viewed as a modified EDD rule. At time t, we use the index $I_i(t) = (t + p_i - d_i)^+ + d_i$ to select the next job for processing among the unsequenced jobs, where a small index value has a high priority and where $(x)^+ = \max\{0, x\}$. We note that the MDD rule can also be viewed as a modified SPT rule since at any time t, $I_i(t)$ is equivalent to $I_i'(t) = p_i + (d_i - t - p_i)^+$, where $I_i(t) = I_i'(t) + t$. We note that $I_i(t)$ is also equivalent to $I_i''(t) = \max\{t + p_i, d_i\}$. As we will show in Theorem 1, the MDD rule satisfies local optimality. 'Local optimality' means the optimality of two adjacent jobs in a given schedule assuming that the jobs preceding and following them are fixed. Consider two jobs i and j and let $T_{hk}(t)$ $(h,\ k \in \{i,\ j\},\ h \neq k)$ denote the sum of the tardinesses of jobs h and k if job h precedes k. When no confusion arises, we will use T_{hk} and I_h instead of $T_{hk}(t)$ and $I_h(t)$.

We note that the MDD rule can be implemented in $O(n \log n)$ in a similar manner as the priority rule $I_j(t) = 2(r_j - t)^+ + p_j$ proposed by Chu [2] for the single machine sequencing problem $1/r_j/\sum F_j$ using the notations by Lawler et al. [8]. Chu [2] showed that the priority rule $I_j(t) = 2(r_j - t)^+ + p_j$ satisfies local optimality and showed that the priority rule has a worst-case performance ratio somewhere between $\frac{n+1}{3}$ and $\frac{n+1}{2}$. However, an exact bound was not derived. The following theorem shows that the MDD rule satisfies local optimality. We note that the MDD rule reduces to the EDD rule in case of equal processing times and reduces to the SPT rule in case of equal due dates. Since $I_i''(t) = I_i(t)$, we will also consider $I_i''(t)$ in what follows.

Theorem 1. At any time t, $T_{ij} \le T_{ji}$ if $I_i'' \le I_j''$.

Proof. First we note that

$$T_{ij} = I_i'' + \max\{t+p_i+p_j,\ d_j\} - d_i - d_j \quad \text{and} \quad T_{ji} = I_j'' + \max\{t+p_i+p_j,\ d_i\} - d_i - d_j.$$

First consider the case that $t+p_i+p_j > d_j$. Since $I_i'' \le I_j''$ and since $\max\{t+p_i+p_j,\ d_j\} \le \max\{t+p_i+p_j,\ d_i\}$ in this case, we clearly have $T_{ij} \le T_{ji}$. We then consider the case that $t+p_i+p_j \le d_j$. Since $\max\{t+p_i+p_j,\ d_j\} = d_j = \max\{t+p_j,\ d_j\} = I_j''$ in this case and since it is clear that $I_i'' \le \max\{t+p_i+p_j,\ d_i\}$, we have $T_{ij} \le T_{ji}$. ◇

Before we derive the worst-case performance ratio for the MDD rule, we first develop some properties associated with the MDD rule. For the derivation of η^{MDD}, if ties occur, we assume that the MDD rule selects the job with the larger processing time between two nontardy jobs, selects the job with the smaller due date between two tardy jobs, and selects the tardy job between a tardy job and a nontardy job; otherwise, selects the job with a smaller subscript. Whenever there is a choice in an optimal sequence, we sequence the job first with a smaller subscript.

In the proofs of Lemmas 1(1), 1(2) and 1(3), we apply a perturbation technique, where we perturb an instance $I = (p_1,\ d_1,\ \ldots,\ p_n,\ d_n)$ to construct another instance I' at least as bad as I. Three types of perturbations are used in the following proofs. In type I perturbation, we increase a due date d_h an infinitely small amount ϵ. In type II perturbation, we decrease a due date d_h an infinitely small amount ϵ. In type III perturbation, we both increase a processing time p_h and decrease another processing time p_j the same infinitely small amount ϵ. Whenever we apply the perturbation technique, we assume that we consider the optimal sequence which remains unchanged after the perturbation in case there are alternative optimal sequences. We also assume that ϵ is chosen such that both σ^{MDD} and σ^* remain unchanged after the perturbation. Another technique in constructing I' from I, where I' is at least as bad as I, is deleting job n if job n is nontardy or aggregating several consecutive nontardy jobs into a single nontardy job.

In what follows, we assume that, by renumbering the jobs, $\sigma^{MDD}(I) = (1\ 2\ \ldots\ n)$.

Lemma 1. When deriving the worst-case performance ratio for the MDD rule, without loss of generality, given an instance $I = (p_1,\ d_1,\ \ldots,\ p_n,\ d_n)$, we can assume that

(1) the first job is nontardy and the other jobs are tardy in σ^{MDD};

(2) $d_i = d$ for all i;

(3) $p_1 = d_1$.

Proof. (1) We first show that we only need to consider an instance $I = (p_1, d_1, \ldots, p_n, d_n)$ in which $p_i \le d_i$ for all i. Therefore, $T_1^{MDD} = 0$. Suppose that $p_i > d_i$, where i is the smallest, in σ^{MDD}. In constructing I' from I, we apply type I perturbation on job i.

Suppose that $T_h^{MDD} = 0$, where h is the largest and $2 \le h \le n$. Since we can delete job n from I to obtain I' if $h = n$, we can assume that $2 \le h < n$. If all the first h jobs are nontardy, then we can construct I' from I by aggregating all the first h jobs into a single job with a due date of d_h and a processing time of $\sum_{j=1}^{h} p_j$. Thus, we can assume that job h', h' being the largest and $h' < h$, is tardy. We clearly can assume that $h' = h - 1$.

Let job k_1 (k_2) denote the job having identical processing time and due date with job $h-1$ ($h+1$), where k_1 (k_2) is the smallest with $k_1 \le h-1$ ($k_2 \le h+1$). Also, let job l_1

(l_2) denote the job having identical processing time and due date with job $h-1$ $(h+1)$, where l_1 (l_2) is the largest with $l_1 \geq h-1$ $(l_2 \geq h+1)$.

Consider the case that job $h+1$ is tardy in $\sigma^\star$. We note that in this case job l_2 is tardy in σ^{MDD} due to $l_2 \geq h+1$ and job l_2 is also tardy in $\sigma^\star$ by our construction of $\sigma^\star$. In constructing I' from I, we apply type I perturbation on job l_2.

Consider the case that job $h+1$ completes earlier than d_{h+1} in $\sigma^\star$. We note that in this case job k_2 is strictly nontardy in $\sigma^\star$. In constructing I' from I, we apply type II perturbation on job k_2.

Consider the case that job $h+1$ completes at d_{h+1} in $\sigma^\star$. If job $h-1$ precedes $h+1$ in $\sigma^\star$, we know that job $h-1$ is strictly nontardy and thus job k_1 is strictly nontardy in $\sigma^\star$. In constructing I' from I, we apply type II perturbation on job k_1. If job $h+1$ precedes $h-1$ in $\sigma^\star$, we know that job $h-1$ is tardy and thus job l_1 is tardy in $\sigma^\star$. In constructing I' from I, we apply type I perturbation on job l_1. This completes our proof.

(2) By Lemma 1(1), we can assume that the first job is nontardy and the other jobs are tardy. Suppose that $d_1 = d_2 = \cdots = d_h < d_{h+1}$. We consider three cases: (i) job h is tardy in $\sigma^\star$, (ii) job h completes earlier than d_h in $\sigma^\star$, and (iii) job h completes at d_h in $\sigma^\star$. In case (iii) we further consider two cases: job $h+1$ precedes h in $\sigma^\star$ or the other way around. The proof is similar to that of Lemma 1(1).

(3) By Lemma 1(2), we can assume that $d_i = d$ for all i. Suppose that $p_1 < d$. If $p_1 \leq p_2$, then σ^{MDD} is an optimal sequence since by Lemmas 1(1) and 1(2) and by the definition of the MDD rule, we have $p_2 \leq p_3 \leq \cdots \leq p_n$. Consider the case that $p_1 > p_2$. In this case we know that job 2 precedes job 1 in the optimal sequence. In constructing instance I' from I, we apply type III perturbation by both decreasing p_2 and increasing p_1 the same infinitely small amount ϵ. ◊

We now derive the worst-case performance ratio for the MDD rule.

Theorem 2. $\eta^{MDD} = \frac{n}{2}$.

Proof. By Lemmas 1(1), 2(2) and 2(3), we can assume that $C_1^{MDD} = d$ and $C_i^{MDD} > d$ for $i \geq 2$, where $C_i^{MDD} = \sum_{j=1}^{i} p_j$. By the definition of the MDD rule, we have $p_2 \leq p_3 \leq \cdots \leq p_n$. Clearly, we have

$$T^\star \geq \sum_{j=2}^{n} p_j$$

and

$$T^{MDD} = \sum_{j=2}^{n}\sum_{i=2}^{j} p_i = \sum_{j=2}^{n}(n-j+1)p_j.$$

Thus, we have

$$\frac{T^{MDD}}{T^\star} \leq \frac{\sum_{j=2}^{n}(n-j+1)p_j}{\sum_{j=2}^{n} p_j} \leq \frac{n}{2}.$$

Note that the second inequality holds since $0 < p_2 \leq p_3 \leq \cdots \leq p_n$. Let

$$g \equiv \sup_{0<p_2\leq p_3\leq\cdots\leq p_n} \{f(p_2, p_3, \cdots, p_n)\},$$

where

$$f(p_2, p_3, \cdots, p_n) \equiv \frac{\sum_{j=2}^{n}(n-j+1)p_j}{\sum_{j=2}^{n} p_j}.$$

It is easy to verify by examining the partial derivatives of f that f is nondecreasing in p_i if f is nondecreasing in p_j, where $2 \le i < j \le n$. On the other hand, it is easy to verify that f is nonincreasing in p_j if f is nonincreasing in p_i, where $2 \le i < j \le n$. Also, it is easy to verify that f is nondecreasing in p_2 and nonincreasing in p_n. Hence, we can assume that $0 < p_2 = p_3 = \cdots = p_n$ in deriving g and therefore we have

$$g = \frac{\sum_{j=2}^{n}(n-j+1)}{(n-1)} = \frac{\sum_{j=1}^{n-1} j}{(n-1)} = \frac{n}{2}.$$

Hence, the second inequality holds.

To show that the bound is tight, consider the example with $n = K+1$ and $d_i = d$ for all i: $d = K$, $p_1 = K$, $p_i = 1$, $i = 2, 3, \ldots, K+1$. Then, $\sigma^{MDD} = (1\ 2\ \ldots\ n)$ with $T^{MDD} = \frac{K(K+1)}{2}$ while $\sigma^* = (2\ 3\ \ldots\ n\ 1)$ with $T^* = K$. Hence, we have $\frac{T^{MDD}}{T^*} = \frac{K(K+1)}{2K} = \frac{n}{2}$. ◇

3. Conclusion

The $O(n \log n)$ MDD rule presented in this paper has a worst-case performance ratio of $\frac{n}{2}$ and is superior to other known $O(n \log n)$ heuristics (e.g. the EDD rule, the SPT rule, Moore's algorithm) in the sense of worst-case performance. It seems nontrivial to extend the MDD rule, which satisfies local optimality, to the total weighted tardiness problem. One avenue of further research is to seek such a priority rule for the total weighted tardiness problem. Another avenue of further research is to refine the MDD rule to achieve a constant (independent of n) worst-case performance ratio. Such a heuristic can be used to improve the $O(\frac{n^7}{\epsilon})$ fully polynomial approximation scheme by Lawler [10] to one with $O(\frac{n^6}{\epsilon})$.

Acknowledgements

The authors would like to thank anonymous referees and the associate editor for their suggestions and comments. This research was supported by National Science Council of Taiwan under NSC 82-0301-H002-095.

Reference

[1] Baker, K.R. and L.E. Schrage, "Finding an optimal sequence by dynamic programming: An extension to precedence-related tasks", *Operations Research* **26**, 111-120 (1978).

[2] Chu, C., "Efficient heuristics to minimize total flow time with release dates", *Operations Research Letters* **12**, 321-330 (1992).

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

[4] Elmaghraby, S.E., "The one-machine sequencing problem with delay costs", *Journal of Industrial Engineering* **19**, 105-108 (1968).

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

[6] Fisher, M.L., "A dual algorithm for the one-machine scheduling problem", *Mathematical Programming* **11**, 229-251 (1976).

[7] Jackson, J.R., "Scheduling a production line to minimize maximum tardiness",

Research Report 43, Management Science Research Project, UCLA, 1955.

[8] Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, and D. B. Shmoys, "Sequencing and scheduling: algorithms and complexity", in: S.C. Graves et al. (Eds.), *Handbooks in Operations Research and Management Science, Vol. 4*, 445-522 (1993).

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

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

[11] Moore, J.M., "An n-job, one machine sequencing algorithm for minimizing the number of late Jobs", *Management Science* **15**, 102-109 (1968).

[12] Picard, J.C. and M. Queyranne, "The time-dependent traveling salesman problem and applications to the tardiness problem in one-machine scheduling", *Operations Research* **26**, 88-110 (1978).

[13] Potts, C.N. and L.N.V. Wassenhove, "A decomposition algorithm for the single machine total tardiness problem", *Operations Research Letters* **1**, 177-181 (1982).

[14] Rinnooy Kan, A.H.G, B.J. Lageweg, and J.K. Lenstra, "Minimizing total costs in one-machine scheduling", *Operations Research* **23**, 908-927 (1975).

[15] Schrage, L. and K.R. Baker, "Dynamic programming solutions of sequencing problem with precedence constraints", *Operations Research* **26**, 444-449 (1978).

[16] Shwimer, J., "On the n-job, one-machine sequence-independent scheduling problem with tardiness penalties: A branch and bound solution", *Management Science* **18**, B 301-313 (1972).

[17] Srinivasan, V., "A hybrid algorithm for the one-machine sequencing problem to minimize total tardiness", *Naval Res. Logist. Quart.* **18**, 317-327 (1971).

[18] Sturm, L.B.J.M., "A simple optimality proof of Moore's sequencing algorithm", *Management Science* **17**, 116-118 (1970).

Tsung-Chyan LAI: Department of Industrial and
Business Management, College of Management,
National Taiwan University, 50, Lane 144,
Keelung Road Section 4, Taipei 106, TAIWAN