Department of Social Systems and Management

Discussion Paper Series

No. 1198

A Note on FPTAS for Single Machine Weighted Tardiness Problem with a Common Due Date

by

Takeru Kabasawa, Kentaro Murata, Maiko Shigeno

March 2008

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

A note on FPTAS for single machine weighted tardiness problem with a common due date [*]

Takeru Kabasawa, Kentaro Murata, Maiko Shigeno [†]

keywords: Approximate algorithms; Scheduling problem;

1 Introduction

Let us consider n non-preemptable jobs to be scheduled on a single machine. Each job j has a processing time p_j and weight w_j. The machine can handle no more than one job at a time, and it is continuously available from time zero onwards only. All jobs have a common and unrestrictive due date d. For a schedule S, let C_j^S be the completion time of job j. A job j is *early* if $C_j^S \leq d$, and it is *tardy* otherwise. The *tardiness* of a tardy job is defined as $C_j^S - d$. The aim of this problem is to find a schedule S that minimizes the total weighted tardiness:

$$T(S) = \sum_{j=1}^{n} w_j (C_j^S - d)^+,$$

where $(a)^+$ stands for $\max\{a, 0\}$. It is assumed that all p_j and w_j are positive integers and that $P = \sum_{i=1}^{n} p_j$ and $W = \sum_{i=1}^{n} w_j$.

It is known that this problem is NP-hard [7]. An $\mathrm{O}(n^2 d)$ pseudopolynomial dynamic programming algorithm was developed for solving this problem by Lawler–Moore [4]. Some approximation algorithms have been investigated. Let S^* be an optimal schedule. For a worst-case ratio bound α, an *α-approximation algorithm* finds a schedule S such that $T(S) \leq \alpha T(S^*)$. When α is given by $1+\varepsilon$, an α-approximation algorithm is called a *fully polynomial-time approximation scheme* (FPTAS) if its running time is bounded by a polynomial with respect to the length of the problem input and $1/\varepsilon$. Fathi–Nuttle [1] developed a 2-approximation algorithm with a running time of $\mathrm{O}(n^2)$. Recently, the first FPTAS was proposed by Kellerer–Strusevich [2].

[*] This research was partially supported by a Scientific Grant-in-Aid from the Ministry of Education, Science, Sports and Culture of Japan.

[†] Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan.

It is obtained by converting an especially designed dynamic programming algorithm and runs in $\mathrm{O}((n^6/\varepsilon^3)\log n \log W)$ time. This leaves an open question an FPTAS with a running time that is only polynomial in n and $1/\varepsilon$. This paper gives a positive answer to this question. We shall show two slight modifications of Kellerer–Strusevich's FPTAS. The first one runs in $\mathrm{O}((n^5/\varepsilon^3)\log PW \log P \log W)$ time, and the second one runs in $\mathrm{O}((n^7/\varepsilon^3)\log n \log(n/\varepsilon))$ time.

The remainder of this paper is as follows: Section 2 describes some properties of the problem and Kellerer–Strusevich's FPTAS. Sections 3 and 4 propose our modified versions.

2 Previous Work

Without loss of generality, we assume that $0 < \varepsilon < 1$. If $\varepsilon \geq 1$, then a 2-approximation algortihm can be taken as a $(1+\varepsilon)$-approximation algorithm.

Job j on a schedule is said to be *straddling* if it is scheduled as the first tardy job. The subsequent tardy jobs are said to be *late*. We can see that there is an optimal schedule where (i) the first job starts at time zero, (ii) jobs are processed without intermediate idle time, and (iii) late jobs are ordered by non-increasing Smith's ratio p_j/w_j [6], while early jobs are processed in any order. In Kellerer–Strusevich's FPTAS [2], each job in turn is set as the straddling job, and a schedule almost minimizing the total weighted tardiness is determined. The one that has the minimum total weighted tardiness $T(S)$ among found ones is then output. When a straddling job is fixed, the problem is reduced to simply apportioning jobs into early and late, which can be represented by 0-1 variables:

$$x_j = \begin{cases} 1 & \text{if job } j \text{ is late,} \\ 0 & \text{if job } j \text{ is early.} \end{cases}$$

Suppose that job s is set as the straddling job. We index the remaining jobs in accordance with Smith's order so that

$$\frac{p_1}{w_1} \leq \frac{p_2}{w_2} \leq \cdots \leq \frac{p_{n'}}{w_{n'}},$$

where $n' = n-1$. From a 0-1 vector $x = (x_1, x_2, \ldots, x_{n'})$, we make a corresponding schedule $S(x)$ whose processing order is given as follows: first, jobs j with $x_j = 0$ are processed according to Smith's order; second, job s is processed; finally, jobs j with $x_j = 1$ are processed according to Smith's order. From conditions (i) and (ii), we assume that the first job in $S(x)$ starts at time zero and there are no idle times in $S(x)$. Note that job s is straddling in $S(x)$ if and only if $\sum_{j=1}^{n'} p_j(1-x_j) \leq d$ and

$\sum_{j=1}^{n'} p_j(1-x_j) + p_s > d$. Let

$$(\mathrm{SP}^s) \quad \text{minimize}\{Z^s(x) \mid \sum_{j=1}^{n'} p_j(1-x_j) \le d, x \in \{0,1\}^{n'}\},$$

where $Z^s(x) = \sum_{j=1}^{n'} w_j \left(\sum_{i=1}^{j} p_i x_i + (\sum_{i=1}^{n'} p_i(1-x_i) + p_s - d)^+\right) x_j + w_s(\sum_{i=1}^{n'} p_i(1-x_i) + p_s - d)^+$. If a feasible solution x of (SP^s) satisfies $\sum_{j=1}^{n'} p_j(1-x_j)+p_s > d$, the completion time of late job j is given by $\sum_{i=1}^{j} p_i x_i + (\sum_{i=1}^{n'} p_i(1-x_i) + p_s - d)^+$. Thus, $Z^s(x)$ is equal to the total weighted tardiness of the corresponding schedule, $T(S(x))$. Otherwise, i.e., if $\sum_{j=1}^{n'} p_j(1-x_j) + p_s < d$ holds, $Z^s(x)$ is the weighted tardiness of a schedule with the same processing order as $S(x)$ and with idle time between job s and the first late job. Therefore, we can evaluate $T(S(x)) \le Z^s(x)$. Moreover, in this case, since s is not the straddling job in $S(x)$, this schedule is also a feasible solution for another straddling job.

For any $x \in \{0,1\}^{n'}$, define

$$P_j^E(x) = \sum_{i=1}^{j} p_i(1-x_i), \quad P_j^T(x) = \sum_{i=1}^{j} p_i x_i,$$

$$W_j(x) = \sum_{i=1}^{j} w_i x_i, \quad \text{and} \quad Z_j(x) = \sum_{i=1}^{j} w_i(\sum_{l=1}^{i} p_l x_l) x_i,$$

where $P_j^E(x)$ and $P_j^T(x)$ are the sums of the processing times of early jobs and late jobs, respectively, provided on a schedule of jobs $1,\dots,j$ according to x. On this schedule, $W_j(x)$ is the sum of the weights of the late jobs, and $Z_j(x)$ is the weighted tardiness when the first late job starts in time d. These values can be reformulated by using the following recursions: $P_j^E(x) = P_{j-1}^E(x) + p_j(1-x_j)$, $P_j^T(x) = P_{j-1}^T(x) + p_j x_j$, $W_j(x) = W_{j-1}(x) + w_j x_j$, $Z_j(x) = Z_{j-1}(x) + w_j P_j^T(x) x_j$, and $P_0^E(x) = P_0^T(x) = W_0(x) = Z_0(x) = 0$. Note that the value $Z^s(x)$ can be written as

$$Z^s(x) = Z_{n'}(x) + (W_{n'}(x) + w_s)(P_{n'}^E(x) + p_s - d)^+. \tag{1}$$

Kellerer–Strusevich's FPTAS needs an upper bound Z_{UB} and a lower bound Z_{LB} on the optimal value of the original problem, where $Z_{\mathrm{UB}}/Z_{\mathrm{LB}}$ is a constant. They use the value of Z_{UB} calculated using a 2-approximation algorithm [1], so that Z_{LB} is set by $Z_{\mathrm{UB}}/2$. The algorithm prepares intervals

$$I_1 = \left[0, \frac{Z_{\mathrm{UB}}}{w_{\pi(1)}}\right], \quad I_j = \left[\frac{Z_{\mathrm{UB}}}{w_{\pi(j-1)}}, \frac{Z_{\mathrm{UB}}}{w_{\pi(j)}}\right] \quad (j = 2,\dots,h),$$

where $w_{\pi(1)} > w_{\pi(2)} > \cdots > w_{\pi(h)}$ are the sorted distinct values among weights w_j $(j = 1,\dots,n')$. Additionally, each interval I_j $(j = 1,\dots,h)$ is split into subintervals

I^r_j with lengths less than $(\varepsilon Z_{\mathrm{LB}})/(4n' w_{\pi(j)})$. Let e^j be the unit vector, i.e., $e^j_j = 1$ and $e^j_i = 0$ for $i \neq j$, and $\mathbf{0} = (0, 0, \ldots, 0)$. Then the following procedure is Kellerer–Strusevich's FPTAS for (SP^s).

step 0 Set $X = \{\mathbf{0}\}$ and $j = 1$.

step 1 Update X by $X \cup \{x + e^j \mid x \in X\}$. (That is, generate two vectors by adding 0 or 1 in position j of each vector $x \in X$.) Calculate values $P^T_j(x)$, $W_j(x)$, and $Z_j(x)$ for each $x \in X$.

step 2 **(2-1)** Round up the values using

$$Z_j(x) = \lceil Z_j(x) \cdot \frac{4n'}{\varepsilon Z_{\mathrm{LB}}} \rceil, \quad W_j(x) = (1 + \frac{\varepsilon}{2})^{\frac{\lceil n' \log_{(1+\varepsilon/2)} W_j(x) \rceil}{n'}},$$

for all $x \in X$. Partition X into $(X^1, X^2, \ldots, X^k)$ such that, for any X^l $(l = 1, \ldots, k)$ and any pair $x, x' \in X^l$, $Z_j(x) = Z_j(x')$ and $W_j(x) = W_j(x')$, and $P^T(x)$ and $P^T(x')$ belong to the same interval I^r_j.

(2-2) Find a vector $x^l_{\min}$ attaining $\min\{P^T_j(x) \mid x \in X^l\}$ and a vector $x^l_{\max}$ attaining $\max\{P^T_j(x) \mid x \in X^l\}$ for each $l = 1, \ldots, k$. Update X by $\{x^l_{\min}, x^l_{\max} \mid l = 1, \ldots, k\}$.

(2-3) If $j < n'$, then update j by $j + 1$ and go to step 1.

step 3 Compute $Z^s(x)$ using Equation (1) for each $x \in X$ and return a minimum one.

The analysis of Kellerer–Strusevich [2] can be explained simply as follows. Let x^s be an optimal solution for (SP^s). The algorithm for (SP^s) finds a solution x such that $Z_{n'}(x) \leq Z_{n'}(x^s) + \frac{\varepsilon Z_{\mathrm{LB}}}{2}$, $P^T_{n'}(x) \geq P^T_{n'}(x^s)$, and $W_{n'}(x) \leq (1 + \frac{\varepsilon}{2}) W_{n'}(x^s)$. Since $P^E_{n'}(x) = \sum_{j=1}^{n'} p_j - P^T_{n'}(x) \leq \sum_{j=1}^{n'} p_j - P^T_{n'}(x^s) = P^E_{n'}(x^s)$, we have $Z^s(x) \leq Z_{n'}(x) + (W_{n'}(x) + w_s)(P^E_{n'}(x^s) + p_s - d)^+$. If s is the straddling job in an optimal schedule, then we obtain

$$\begin{aligned} Z^s(x) - Z^s(x^s) &\leq (Z_{n'}(x) - Z_{n'}(x^s)) + (W_{n'}(x) - W_{n'}(x^s))(P^E_{n'}(x^s) + p_s - d)^+ \\ &\leq \frac{\varepsilon}{2} Z_{\mathrm{LB}} + \frac{\varepsilon}{2} W'_n(x^s)(P^E_{n'}(x^s) + p_s - d)^+ \\ &\leq \varepsilon Z^s(x^s). \end{aligned}$$

Thus, the algorithm finds a $(1 + \varepsilon)$-approximate solution. We next estimate the complexity of the algorithm. The total number of subintervals is bounded by $\frac{Z_{\mathrm{UB}}}{w_{\pi(1)}} \cdot \frac{4n' w_{\pi(1)}}{\varepsilon Z_{\mathrm{LB}}} + \sum_{j=2}^{h} \left(\frac{Z_{\mathrm{UB}}}{w_{\pi(j)}} - \frac{Z_{\mathrm{UB}}}{w_{\pi(j-1)}} \right) \cdot \frac{4n' w_{\pi(j)}}{\varepsilon Z_{\mathrm{LB}}} = \mathrm{O}(\frac{n^2}{\varepsilon})$, and the total number of distinct values of $Z_j(x)$ is bounded by $\lceil Z_{\mathrm{UB}} \cdot \frac{4n'}{\varepsilon Z_{\mathrm{LB}}} \rceil = \mathrm{O}(\frac{n}{\varepsilon})$. Moreover, the number of distinct values of $W_j(x)$ is bounded by $n \log_{(1+\varepsilon/2)} W = \mathrm{O}(\frac{n}{\varepsilon} \log W)$. Thus, the number of subsets of any partition in step (2-1) is bounded by $\mathrm{O}(\frac{n^4}{\varepsilon^3}\log W)$. The original paper [2] stated that step 2 can be implemented in $\mathrm{O}(\frac{n^4}{\varepsilon^3}\log W)$ time. However, we need to determine to which interval I_j each $P_j^T(x)$ belongs. Applying a binary search, we can do it in $\mathrm{O}(\log n)$ time for each solution $x \in X$. Therefore, step 2 is performed in $\mathrm{O}(\frac{n^4}{\varepsilon^3}\log n \log W)$ time, and the total time complexity of the algorithm is $\mathrm{O}(\frac{n^6}{\varepsilon^3}\log n \log W)$.

3 Simple Algorithm

Our first modification of FPTAS for (SP^s) adopts the technique of Kovalyov–Kubiak [3], which does not require any prior knowledge of the lower and upper bounds on the optimal value.

We replace step 2 with the following procedure.

step 2' (2'-1) Compute

$$q_z(x) = \lceil \frac{\log Z_j(x)}{\log(1+\frac{\varepsilon}{2n})} \rceil, \quad q_p(x) = \lceil \frac{\log P_j^T(x)}{\log(1+\frac{\varepsilon}{2n})} \rceil, \quad q_w(x) = \lceil \frac{\log W_j(x)}{\log(1+\frac{\varepsilon}{2n})} \rceil,$$

for all $x \in X$. Partition X into $(X^1, X^2, \ldots, X^k)$ with respect to the same triple , $(q_z(x), q_p(x), q_w(x))$.

(2'-2) Find a vector x^l attaining $\min\{P_j^E(x) \mid x \in X^l\}$ for each $l = 1, \ldots, k$. Update X by $\{x^l \mid l = 1, \ldots, k, P_j^E(x^l) \leq d\}$.

(2-3) If $j < n'$, then update j by $j+1$ and go to step 1.

First we show properties of the obtained partition of X.

Lemma 1 *For a partition $(X^1, X^2, \ldots X^k)$ obtained at the jth iteration of the algorithm and any pair $x, x' \in X^l$ $(l = 1, \ldots, k)$, it holds that $|Z_j(x) - Z_j(x')| \leq \frac{\varepsilon}{2n} \cdot \min\{Z_j(x), Z_j(x')\}$, $|P_j^T(x) - P_j^T(x')| \leq \frac{\varepsilon}{2n} \cdot \min\{P_j^T(x), P_j^T(x')\}$, and $|W_j(x) - W_j(x')| \leq \frac{\varepsilon}{2n} \cdot \min\{W_j(x), W_j(x')\}$.* ∎

Define $\nu_1 = \varepsilon/(2n)$ and $\nu_{j+1} = \nu_j + (1+\nu_j)\varepsilon/(2n)$. Let x^s be an optimal solution of (SP^s)

Lemma 2 *At the end of the jth iteration, there exists $x^j \in X$ satisfying $|P_j^T(x^j) - P_j^T(x^s)| \leq \nu_j P_j^T(x^s)$, $|W_j(x^j) - W_j(x^s)| \leq \nu_j W_j(x^s)$, $|Z_j(x^j) - Z_j(x^s)| \leq \nu_j Z_j(x^s)$, and $P_j^E(x^j) \leq P_j^E(x^s)$.*

Proof. At the end of the first iteration, $X = \{\mathbf{0}, e^1\}$ because $Z_1(e^1) - Z_1(\mathbf{0}) = w_1p_1 - 0 > 0 = \frac{\varepsilon}{2n} \cdot Z_1(\mathbf{0})$. Since either $\mathbf{0}$ or e^1 has the same values of $P_1^T(x)$, $W_1(x)$, $Z_1(x)$, and $P_1^E(x)$ with x^s, the lemma holds at the first iteration.

Suppose that the statement is true at the $(j-1)$th iteration. That is, there exists a vector $x^{j-1} = (x_1^{j-1}, \dots, x_{j-1}^{j-1}, 0, \dots, 0)$ satisfying the conditions. Let $\tilde{x}^j = (x_1^{j-1}, \dots, x_{j-1}^{j-1}, x_j^s, 0 \dots, 0)$ and x^j be a minimizer identified in step (2'-2) in the subset containing $\tilde{x}^j$. Since we have

$$\begin{aligned} |P_j^T(\tilde{x}^j) - P_j^T(x^s)| &= |(P_{j-1}^T(x^{j-1}) + p_j x_j^s) - (P_{j-1}^T(x^s) + p_j x_j^s)| \\ &= |P_{j-1}^T(x^{j-1}) - P_{j-1}^T(x^s)| \\ &\leq \nu_{j-1} P_{j-1}^T(x^s) \quad \leq \quad \nu_{j-1} P_j^T(x^s), \end{aligned} \tag{2}$$

it holds that $P_j^T(\tilde{x}^j) \leq (1+\nu_{j-1})P_j^T(x^s)$. Hence, we obtain

$$\begin{aligned} |P_j^T(x^j) - P_j^T(x^s)| &\leq |P_j^T(x^j) - P_j^T(\tilde{x}^j)| + |P_j^T(\tilde{x}^j) - P_j^T(x^s)| \\ &\leq \frac{\varepsilon}{2n} P_j^T(\tilde{x}^j) + \nu_{j-1} P_j^T(x^s) \\ &\leq \frac{\varepsilon}{2n}(1+\nu_{j-1})P_j^T(x^s) + \nu_{j-1}P_j^T(x^s) = \nu_j P_j^T(x^s). \end{aligned}$$

Analogously, we can show that $|W_j(x^j) - W_j(x^s)| \leq \nu_j W_j(x^s)$. Using (2), we have

$$\begin{aligned} |Z_j(\tilde{x}^j) - Z_j(x^s)| &\leq |Z_{j-1}(x^{j-1}) - Z_{j-1}(x^s)| + w_j |P_j^T(\tilde{x}^j) - P_j^T(x^s)| x_j^s \\ &\leq \nu_{j-1} Z_{j-1}(x^s) + w_j \nu_{j-1} P_j^T(x^s) x_j^s = \nu_{j-1} Z_j(x^s). \end{aligned}$$

Hence, we obtain

$$\begin{aligned} |Z_j(x^j) - Z_j(x^s)| &\leq |Z_j(x^j) - Z_j(\tilde{x}^j)| + |Z_j(\tilde{x}^j) - Z_j(x^s)| \\ &\leq \frac{\varepsilon}{2n} Z_j(\tilde{x}^j) + \nu_{j-1} Z_j(x^s) \\ &\leq \frac{\varepsilon}{2n}(1+\nu_{j-1}) Z_j(x^s) + \nu_{j-1} Z_j(x^s) = \nu_j Z_j(x^s). \end{aligned}$$

Finally, it holds that

$$P_j^E(x^j) \leq P_j^E(\tilde{x}^j) = P_{j-1}^E(x^{j-1}) + p_j(1-x_j^s) \leq P_{j-1}^E(x^s) + p_j(1-x_j^s) = P_j^E(x^s).$$

Thus, we obtain $P_j^E(x^j) \leq P_j^E(x^s) \leq d$, which implies that vector x^j is in X at the end of the jth iteration. ▌

Lemma 3 *Let x be a solution obtained by the algorithm for (SP^s). Then $Z^s(x) - Z^s(x^s) \leq \varepsilon Z^s(x^s)$ holds.*

Proof. Let $x^{n'}$ be a solution satisfying the conditions of Lemma 2 at the end of the algorithm. Since $P_{n'}^E(x^{n'}) \leq P_{n'}^E(x^s)$, we have $Z^s(x^{n'}) \leq Z_{n'}(x^{n'}) + (W_{n'}(x^{n'}) + w_s)(P_{n'}^E(x^s) + p_s - d)^+$. Hence, we obtain

$$\begin{aligned} Z^s(x) - Z^s(x^s) &\leq Z^s(x^{n'}) - Z^s(x^s) \\ &\leq (Z_{n'}(x^{n'}) - Z_{n'}(x^s)) + (W_{n'}(x^{n'}) - W_{n'}(x^s))(P_{n'}^E(x^s) + p_s - d)^+ \\ &\leq \nu_{n'} Z_{n'}(x^s) + \nu_{n'} W_{n'}(x^s)(P_{n'}^E(x^s) + p_s - d)^+ \\ &< \nu_{n'} Z_{n'}(x^s) + \nu_{n'}(W_{n'}(x^s) + w_s)(P_{n'}^E(x^s) + p_s - d)^+ = \nu_{n'} Z^s(x^s), \end{aligned}$$

which, together with $\nu_{n'} \leq \varepsilon$ (See [3, Theorem 1]), establishes the lemma. ▮
The following shows that the whole algorithm finds a $(1+\varepsilon)$-approximate solution.

Theorem 4 *Let S^* be an optimal schedule. Assume that $\hat{x}$ minimizes $Z^s(x)$ among the solutions obtained by the algorithm for each job as the straddling job. We then have $T(S(\hat{x})) - T(S^*) \leq \varepsilon T(S^*)$.*

Proof. Let s^* be the straddling job in S^*. Then we have a 0-1 vector x^* for (SP^{s^*}) such that $T(S^*) = T(S(x^*))$. Since $\sum_{i \in \{1,\ldots,n\}, i \neq s^*} p_i(1-x_i^*) + p_{s^*} \leq d$, $T(S(x^*)) = Z^{s^*}(x^*)$ holds. From Lemma 3, the algorithm for (SP^{s^*}) finds a solution $\tilde{x}$ with $Z^{s^*}(\tilde{x}) - Z^{s^*}(x^*) \leq \varepsilon Z^{s^*}(x^*)$. Thus, we obtain $T(S(\hat{x})) - T(S^*) \leq T(S(\tilde{x})) - T(S^*) \leq Z^{s^*}(\tilde{x}) - Z^{s^*}(x^*) \leq \varepsilon Z^{s^*}(x^*)$. ▮

Finally, we establish the time complexity of our algorithm. The numbers of distinct values of $q_z(x)$, $q_p(x)$, and $q_w(x)$ are bounded by $\log_{(1+\varepsilon/(2n))} PW$, $\log_{(1+\varepsilon/(2n))} P$, and $\log_{(1+\varepsilon/(2n))} W$, respetively. Thus, the number of subsets in any partition in step (2'-1) is bounded by $\mathrm{O}(\frac{n^3}{\varepsilon^3} \log PW \log P \log W)$. We can summarize the following result.

Theorem 5 *The modified algorithm is also an FPTAS with running time $\mathrm{O}(\frac{n^5}{\varepsilon^3} \log PW \log P \log W)$.* ▮

4 Strongly Polynomial Time Algorithm

The second modification of Kellerer–Strusevich's FPTAS is designed for the running time that is only polynomial in n and $1/\varepsilon$. This modification replaces step (2-1) with the following procedure.

(2"-1) Round up the values

$$Z_j(x) = \lceil Z_j(x) \cdot \frac{4n'}{\varepsilon Z_{\mathrm{LB}}} \rceil$$

for all $x \in X$. Partition X into $(X^1, X^2, \ldots, X^k)$ such that for any X^l $(l = 1, \ldots, k)$ and any pair $x, x' \in X^l$, $Z_j(x) = Z_j(x')$, and $|W_j(x) - W_j(x')| \leq \frac{\varepsilon}{4n} \min\{W_j(x), W_j(x')\}$ hold, and $P^T(x)$ and $P^T(x')$ belong to the same interval I_j^r.

That is, instead of rounding up the values $W_j(x)$, we divide X into groups by using the values of $W_j(x)$. As Lemmas 1 and 2, we can show that, at the end of this modified algorithm for (SP^s), X contains a solution x such that $|W_{n'}(x) - W_{n'}(x^s)| \leq \frac{\varepsilon}{2} W_{n'}(x^s)$, where x^s is an optimal solution for (SP^s). Together with the analysis of Kellerer–Strusevich [2] described in Section 2, we can show that this modified algorithm also finds a $(1+\varepsilon)$-approximate solution.

Let $W_1 < W_2 < \cdots < W_m$ be the sorted distinct values among values $\{W_j(x) \mid x \in X\}$ at the jth iteration. Split the interval $[W_1, W_m]$ into h' intervals:

$$J_i = [W_{\rho(i)}, W_{\rho(i+1)}] \quad (i = 1, \ldots, h'),$$

where ρ satisfies $\rho(1) = 1, \rho(h' + 1) = m$, and $2W_{\rho(i)} \leq W_{\rho(i+1)}$ and $2W_{\rho(i)} > W_{\rho(i+1)-1}$ hold for each i. Additionally, each interval $J_i (i = 1, \ldots, h)$ is split into subintervals J_i^r with lengths less than $\frac{\varepsilon}{4n'} W_{\rho(i)}$. Then for any x, x' in the same subinterval J_i^r, we obtain $|W_j(x) - W_j(x')| \leq \frac{\varepsilon}{4n'} W_{\rho(i)} \leq \frac{\varepsilon}{4n'} \min\{W_j(x), W_j(x')\}$. Hence, we obtain a partition $(X^1, \ldots, X^k)$ of X by using these subintervals.

Finally, we estimate the complexity of this modified algorithm. The total number of subintervals in J_i can be bounded by $(W_{\rho(i+1)-1} - W_{\rho(i)})/(\frac{\varepsilon}{4n'} W_{\rho(i)}) \leq (2W_{\rho(i)} - W_{\rho(i)})/(\frac{\varepsilon}{4n'} W_{\rho(i)}) = \frac{4n'}{\varepsilon} = \mathrm{O}(\frac{n}{\varepsilon})$. We use as bound h' the result of Radzik [5] as follows.

Lemma 6 ([5], Corollary4.2) *Let $c = (c_1, c_2, \ldots, c_p) \in \mathbf{R}^p$, and $y^1, y^2, \ldots, y^q$ be vectors from $\{0,1\}^p$. If, for all $i = 1, 2, \ldots, q-1$, $0 < y^{i+1}c \leq (y^i\ c)/2$, then $q = \mathrm{O}(p \log p)$.* ▌

Thus, we obtain $h' = \mathrm{O}(n' \log n')$, and the number of total subintervals can be bounded above from $\mathrm{O}(\frac{n^2}{\varepsilon} \log n)$. From Kellerer–Strusevich [2], the total number of subintervals I_j^r is bounded by $\mathrm{O}(n^2/\varepsilon)$, and the total number of distinct values of $Z_j(x)$ is bounded by $\mathrm{O}(n/\varepsilon)$. Thus, the number of subsets of any partition is bounded by $\mathrm{O}((n^5/\varepsilon^3) \log n)$. In each iteration, because we need to sort the values $W_j(x)$ for all $x \in X$ in order to obtain subintervals, step (2"-1) can be performed in $\mathrm{O}(k \log k)$ time where $k = \mathrm{O}((n^5/\varepsilon^3) \log n)$. Therefore the modified algorithm for (SP^s) finds an approximate solution in $\mathrm{O}(\frac{n^5}{\varepsilon^3} \log n \log \frac{n}{\varepsilon}) \times n' = \mathrm{O}(\frac{n^6}{\varepsilon^3} \log n \log \frac{n}{\varepsilon})$ time and we conclude the following.

Theorem 7 *The second modified algorithm is also FPTAS with running time $\mathrm{O}(\frac{n^7}{\varepsilon^3} \log n \log \frac{n}{\varepsilon})$.* ▌

5 Conclusion

We have described two FPTASs for the single machine weighted tardiness problem with a common due date, thereby resolving the question of whether there is an FPTAS with a running time that is only polynomial in n and $1/\varepsilon$. The technique used can also be used to develop FPTASs for other combinatorial optimization problems: variations of one machine scheduling, partitioning problems, shortest weight-constrainted path problems, and so on.

References

[1] Fathi, Y. & Nuttle, H. W. L., Heuristics for the common due date weighted tardiness problem, 1990, IIE Trans., 22, 215-225.

[2] Kellerer, H. & Strusevich,V. A., A fully polynomial approximation scheme for the single machine weighted total tardiness problem with a common due date, 2006, Theoret. Comput. Sci., 369, 230-238.

[3] Kovalyov, M. Y. & Kubiak, W., A fully polynomial approximation scheme for the weighted earliness-tardiness problem, 1999, Operations Research, 47, 757-761.

[4] Lawler, E. L. & Moore, J. M., A functional equation and its application to resource allocation and sequencing problems, 1969, Management Sci., 16, 77-84.

[5] Radzik, T., Fractional combinatorial optimization, in *Handbook of Combinatorial Optimization*, Vol. 1, Du, D.Z. and Pardalos, P. M. (eds.) 1998, 429–478.

[6] Smith, W. E., Various optimizers for single stage production, 1956, Naval Res. Logistics Quart., 3, 59-66.

[7] Yuan, J., The NP-hardness of the single machine common due date weighted tardiness problem, 1992, Systems Sci. Math. Sci., 5, 328-333.