Special case of the single machine total tardiness problem is NP-hard

Alexander A. Lazarev, Eugene R. Gafarov

Computing Centre of the Russian Academy of Sciences, Vavilov st. 40, 119991 Moscow GSP-1, Russia

Alexandr.Lazarev@mail.ru, axel73@mail.ru

Summary

Given a single machine and set of jobs with processing times (p_j) and due dates (d_j). The classical NP-hard scheduling problem $1||\sum T_j$ to minimize the total tardiness is a well-understood one. In this paper, we show that the special case **B-1** of the problem when $d_{max} - d_{min} \le p_{min}$ is NP-hard in the ordinary sense. For this case we have constructed a pseudo-polynomial algorithm with run time $O(n \sum p_j)$.

1 Introduction.

Given a set N of n independent jobs that must be processed on a single machine. Preemptions of jobs are not allowed. The single machine can handle only one job at a time. The jobs are available for processing at time 0. For each job j, $j \in N$, a processing time $p_j > 0$ and a due date d_j are given. A schedule π is uniquely determined by a permutation of elements of N. We need to construct an optimal schedule π^* which minimizes the total tardiness value $F(\pi) = \sum_{j=1}^{n} \max\{0, C_j(\pi) - d_j\}$, where $C_j(\pi)$ is the completion time of job j in schedule π. $T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$ is the tardiness of job j in schedule π. The problem $1 \mid\mid \sum T_j$ is NP-hard in the ordinary sense [1]. A pseudo-polynomial time $O(n^4 \sum p_j)$ dynamic programming algorithm has been proposed by Lawler [2]. The state-of-the-art algorithms of Szwarc et al.[3, 4] handle special instances [5] of the problem for $n \le 500$.

We show that the special case **B-1** [6] is NP-hard in the ordinary sense. Notice that there exists a pseudo-polynomial algorithm with run time $O(n \sum p_j)$ for the case **B-1**. We propose a polynomial scheme of reduction from NP-complete **Even-Odd Partition Problem** to the special case **B-1** of the problem $1 \mid\mid \sum T_j$.

2 Even-Odd Partition Problem (EOP).

Given a set of $2n$ positive integers $B = \{b_1, b_2, \ldots, b_{2n}\},\ b_i \geq b_{i+1},\ i = 1, 2, \ldots, 2n-1$. Is there a partition of B into two subsets B_1 and B_2 such that $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$ and such that for each $i = 1, \ldots, n$ B_1 (and hence, B_2) contains exactly one number of $\{b_{2i-1}, b_{2i}\}$? The EOP problem is well known NP-complete problem.

Let $\delta_i = b_{2i-1} - b_{2i},\ i = 1, \ldots, n,\ \delta = \sum_{i=1}^{n} \delta_i$. Now we construct the modified Even-Odd Partition Problem. There is the following set of integers $A = \{a_1, a_2, \ldots, a_{2n}\}$.

$$\begin{cases} a_{2n} = M + b, \\ a_{2i} = a_{2i+2} + b,\ i = n-1, \ldots, 1, \\ a_{2i-1} = a_{2i} + \delta_i,\ i = n, \ldots, 1, \end{cases} \tag{1}$$

where $b \gg n\delta$ (for example $b = n^2\delta$), $M \geq n^3 b$. Obviously, we have $a_i \geq a_{i+1},\ \forall i = 1, 2, \ldots, 2n-1$. Notice that $\delta_i = b_{2i-1} - b_{2i} = a_{2i-1} - a_{2i},\ i = 1, \ldots, n$.
The modified problem is equivalent to the original one.
For example, let $B = \{b_1, b_2, \ldots, b_6\} = \{15, 12, 12, 7, 5, 3\},\ \delta = 10,\ n = 3$. For $B_1 = \{b_1, b_4, b_5\}$ and $B_2 = \{b_2, b_3, b_6\}$ we have $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i = 27$. We denote $b = n^2\delta = 3^2\delta = 90$, $M = n^3 b = 2430$, $A = \{2703, 2700, 2615, 2610, 2522, 2520\}$. The modified EOP instance has a solution too: $A_1 = \{2703, 2610, 2522\},\ A_2 = \{2700, 2615, 2520\}$.

Lemma 1 *The original EOP problem has a solution if and only if the modified EOP problem does.*

Proof. Let for the original problem there exist two subsets B_1 and B_2 that $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$. We denote $A_1 = \{a_i | b_i \in B_1\},\ A_2 = \{a_i | b_i \in B_2\}$. Then we have $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$.

Let for the modified problem there exist two subsets A_1 and A_2 that $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$. Let's denote $B_1 = \{b_i | a_i \in A_1\},\ B_2 = \{b_i | a_i \in A_2\}$. We have $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$.$\square$

3 Special case of the $1 \mid\mid \sum T_j$ problem.

The following case **B-1** of the problem $1 \mid\mid \sum T_j$ is considered [6]:

$$\begin{cases} p_1 \geq p_2 \geq \ldots \geq p_n, \\ d_1 \leq d_2 \leq \ldots \leq d_n, \\ d_n - d_1 \leq p_n. \end{cases} \tag{2}$$

This case is called "hard" instances in the paper [7]. The research of known algorithms [3, 6, 8] has shown that for case **B-1** the number of branchings in the search tree is large [6].

Definitions. The sequence $\pi = (j_1, j_2, \ldots, j_n)$ is an **SPT schedule** (shortest processing time), if $p_{j_i} \leq p_{j_{i+1}}$, for $p_{j_i} = p_{j_{i+1}}$ it holds $d_{j_i} \leq d_{j_{i+1}}$, $i = 1, 2, \ldots, n-1$. The sequence $\pi = (j_1, j_2, \ldots, j_n)$ is an **EDD schedule** (earliest due date), if $d_{j_i} \leq d_{j_{i+1}}$, for $d_{j_i} = d_{j_{i+1}}$ it holds $p_{j_i} \leq p_{j_{i+1}}$, $i = 1, 2, \ldots, n-1$.

For the case (2) the sequence $\pi = (1, 2, \ldots, n)$ is an EDD schedule. The sequence $\pi = (n, n-1, \ldots, 1)$ is an SPT schedule.

The sequence π' is an *partial schedule*, if it contains only jobs from subset $N' \subset N$. Let $\{\pi'\}$ is subset $N' \subset N$ of jobs processed in π', and we denote $P(\pi') = \sum_{i \in \{\pi'\}} p_i$.

Lemma 2 [6] *For the case (2) there exists an optimal sequence $\pi^* = (\pi_{EDD}, \pi_{SPT})$, where π_{EDD} and π_{SPT} are partial sequences constructed according to EDD and SPT rules.*

Corollary.[6] For the case (2) late jobs for all optimal schedules are processed according to the SPT order, except, may be, the first one.

Now we present the polynomial reduction from the modified **EOP** problem to the special subcase (2) of the problem $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$. This case we denote as canonical LG case.

We denote the jobs as $V_1, V_2, V_3, V_4, \ldots, V_{2i-1}, V_{2i}, \ldots, V_{2n-1}, V_{2n}, V_{2n+1}$, $N = \{1, 2, \ldots, 2n, 2n+1\}$.

$$
\begin{cases}
p_1 > p_2 > \ldots > p_{2n+1}, & (3.1)\\
d_1 < d_2 < \ldots < d_{2n+1}, & (3.2)\\
d_{2n+1} - d_1 < p_{2n+1}, & (3.3)\\
p_{2n+1} = M = n^3 b, & (3.4)\\
p_{2n} = p_{2n+1} + b = a_{2n}, & (3.5)\\
p_{2i} = p_{2i+2} + b = a_{2i},\ i = n-1, \ldots, 1, & (3.6)\\
p_{2i-1} = p_{2i} + \delta_i = a_{2i-1},\ i = n, \ldots, 1, & (3.7)\\
d_{2n+1} = \sum_{i:=1}^{n} p_{2i} + p_{2n+1} + \frac{1}{2}\delta, & (3.8)\\
d_{2n} = d_{2n+1} - \delta, & (3.9)\\
d_{2i} = d_{2i+2} - (n-i)b + \delta,\ i = n-1, \ldots, 1, & (3.10)\\
d_{2i-1} = d_{2i} - (n-i)\delta_i - \varepsilon\delta_i,\ i = n, \ldots, 1, & (3.11)
\end{cases}
\qquad (3)
$$

where $b = n^2\delta$, $0 < \varepsilon < \frac{\min_i \delta_i}{\max_i \delta_i}$. The due dates pattern of the *canonical LG instance* is presented on the Fig. 1.

Let $L = \frac{1}{2}\sum_{i:=1}^{2n} p_i$, then we have $d_{2n+1} = L + p_{2n+1}$ because $\frac{1}{2}\sum_{i:=1}^{2n} p_i = \sum_{i:=1}^{n} p_{2i} + \frac{1}{2}\delta$. *Canonical DL instances* from paper [1] do not correspond to the case (3).

4 Properties of the special case (3) of the problem $1||\sum T_j$.

Lemma 3 *For the case (3), for all sequences, the number of tardy jobs equals n or $(n+1)$.*

Proof.

1. We consider set N' of $(n+2)$ jobs with the smallest processing times and process its in the begin of schedule. We have $\sum_{i\in N'} p_i > (n+2)p_{min} = (n+2)n^3 b$, where $p_{min} = \min_{j\in N}\{p_j\} = p_{2n+1}$.

   According to (3.4)-(3.8),

   $$d_{max} = \max_{j\in N}\{d_j\} = d_{2n+1} = (n+1)n^3 b + (b + 2b + \ldots + nb) + \frac{1}{2}\delta,$$

   therefore

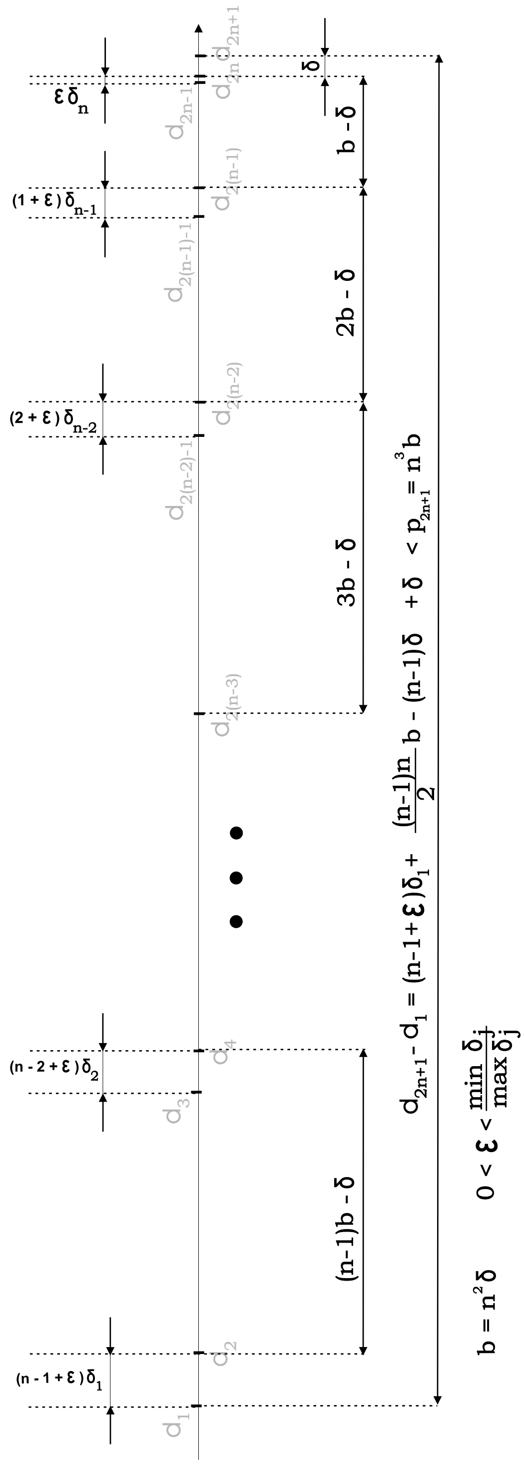

Figure 1: Due date pattern of the canonical LG instance.

$$d_{max} = d_{2n+1} = (n+1)n^3 b + \frac{n(n+1)}{2} b + \frac{1}{2}\delta < (n+2)n^3 b < \sum_{i \in N'} p_i,$$

Thus the job processed on the $(n+2)th$ position is tardy in all schedules. Following jobs are tardy too according to (3.3). So in all schedules the number of tardy jobs is greater or equal to n.

2. Let's consider set of n long processing time jobs N' and process its in the begin of schedule. Two cases are considered

   a) let $n = 2k$, then $N'' = \{V_1, V_2, \dots, V_{2k-1}, V_{2k}\}$, we have

$$P(N'') = nn^3 b + 2(nb + (n-1)b + \dots + (n-k+1)b) + \sum_{i:=1}^{k} \delta_i,$$

$$P(N'') = nn^3 b + 2(\frac{n(n+1)}{2} - \frac{(n-k)(n-k+1)}{2})b + \sum_{i:=1}^{k} \delta_i.$$

   According to (3.8) - (3.11),
   $d_{min} = \min_{j \in N}\{d_j\} = d_1 = d_{2n+1} - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 - \varepsilon\delta_1) = (n+1)n^3 b + (b + 2b + \dots + nb) + \frac{1}{2}\delta - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 + \varepsilon\delta_1) > P(N'')$;

   b) let $n = 2k+1$, then $N'' = \{V_1, V_2, \dots, V_{2k-1}, V_{2k}, V_{2(k+1)-1}\}$ and

$$P(N'') = nn^3 b + 2(nb + (n-1)b + \dots + (n-k+1)b) + (n-k)b + \sum_{i:=1}^{k+1} \delta_i,$$

$$P(N'') = nn^3 b + 2(\frac{n(n+1)}{2} - \frac{(n-k)(n-k+1)}{2})b + (n-k)b + \sum_{i:=1}^{k+1} \delta_i,$$

   $d_{min} = d_1 = d_{2n+1} - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 - \varepsilon\delta_1) = (n+1)n^3 b + (b + 2b + \dots + nb) + \frac{1}{2}\delta - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 + \varepsilon\delta_1) > P(N'')$, Jobs from the set N' aren't tardy. That's why in all sequences the number of tardy jobs is less or equal $(n+1)$.

Thus for the case (3) in all sequences the number of tardy jobs equals n or $(n+1)$. ■

Lemma 4 *For the case (3), for all schedules* $\pi = (\pi_1, \pi_2)$, *there exist a schedule* $\pi' = (\pi_{EDD}, \pi_{SPT})$, *where* $\{\pi_1\} = \{\pi_{EDD}\}$, $\{\pi_2\} = \{\pi_{SPT}\}$, $|\{\pi_1\}| = (n+1)$, $|\{\pi_2\}| = n$, *and what is more* $F(\pi) \geq F(\pi')$ *holds.*

Proof.

The partial sequence π_1 are considered. Because first n jobs in π_1 aren't tardy that's why the EDD order is optimal for set of jobs $\{\pi_1\}$. In this case on the $(n+1)th$ position job $j = argmax\{d_i : i \in \{\pi_1\}\}$ are processed.

Now we consider the sequence π_2. The EDD order is optimal for set of jobs $\{\pi_2\}$, because all n jobs are tardy. ■

Let $(V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n,1}, V_{2n+1}, V_{n,2}, \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$ is *canonical LG schedule*, where $\{V_{i,1}, V_{i,2}\} = \{V_{2i-1}, V_{2i}\},\ i = 1, 2, \ldots, n$.

Lemma 5 *If the sequence* $\pi = (\pi_1, \pi_2)$, $|\{\pi_1\}| = (n+1)$, $|\{\pi_2\}| = n$ *is not canonical LG schedule or we cannot reduce it to canonical LG schedule by EDD and SPT rules to* $\{\pi_1\}$ *and* $\{\pi_2\}$ *sets, then in the schedule* π *two jobs* $\{V_{2i-1},\ V_{2i}\}$ *are on-time or*

$$(V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n}, V_{2n+1}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2}), \tag{4}$$

jobs $\{V_{2n-1}, V_{2n}\}$ *precede* V_{2n+1}.

Proof. Let $\pi = (\pi_1, \pi_2)$, where $|\{\pi_1\}| = (n+1)$, $|\{\pi_2\}| = n$. Let's consider following cases:

1. If $\{\pi_2\} = \{V_{1,2}, \ldots, V_{n,2}\}$ so π_2 consists only one job from set $\{V_{2i-1}, V_{2i}\}$, for all $i = 1, \ldots, n$. Let's arrange jobs from $\{\pi_2\}$ by SPT rule. We have new schedule π'. According to lemma 4, $F(\pi') \leq F(\pi)$.

2. If $\{\pi_2\} \neq \{V_{1,2}, \ldots, V_{n,2}\}$. Following cases are possible:

    a) $V_{2n+1} \in \{\pi_2\}$,

b) there exist the pair of jobs $\{V_{2j-1}, V_{2j}\} \subset \{\pi_2\}$.

Then for some i we have $\{V_{2i-1}, V_{2i}\} \subset \{\pi_1\}$, because $|\{\pi_2\}| = n$.

■

In Theorem 1 we show that for the case (3) all optimal schedules are canonical LG schedules. We will prove that a schedule π can be transformed to a canonical LG schedule π' and $F(\pi) \geq F(\pi')$. In the proof of Theorem 1 Lemmas 6, 7, 8, 9 are used.

Lemma 6 *Let*

$$\pi = (V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n}, V_{2n+1}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2}),$$

where the job V_{2n+1} is processed on the $(n+2)$th position. For schedule $\pi' = (V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n+1}, V_{2n}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$ we'll have $F(\pi) > F(\pi')$.

Proof. In schedule π the job V_{2n-1} on the n-th position are processed. According to lemma 3, the job V_{2n-1} isn't tardy. The job V_{2n+1} on the $(n+2)$th position are processed, that's why it's a tardy job.

For jobs $\{V_2, V_4, \ldots, V_{2i}, \ldots, V_{2n-2}, V_{2n-1}\}$ we have

$$P(\{V_2, V_4, \ldots, V_{2i}, \ldots, V_{2n-2}, V_{2n-1}\}) = nn^3b + \sum_{k:=1}^{n} kb + \delta_n = d_{V_{2n+1}} - n^3b - \frac{1}{2}\delta + \delta_n,$$

according to (3.8). Obviously,

$$P(\{V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}\}) + p_{V_{2n}} \geq P(\{V_2, V_4, \ldots, V_{2i}, \ldots, V_{2n-2}, V_{2n-1}\}) + p_{V_{2n}}$$

holds, thus

$$C_{2n}(\pi) \geq d_{2n+1} + b - \frac{1}{2}\delta + \delta_n > d_{2n}.$$

So the job V_{2n} in schedule π is tardy.

Let $\pi = (\pi_{11}, V_{2n}, V_{2n+1}, \pi_{21})$. Consider the canonical LG schedule $\pi' = (\pi_{11}, V_{2n+1}, V_{2n}, \pi_{21})$. We aims to show $F(\pi) > F(\pi')$.

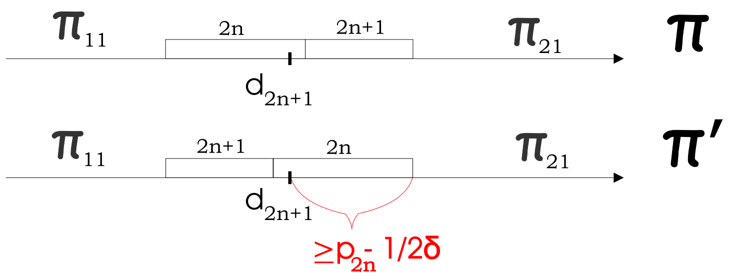

Figure 2: The permutation of V_{2n} and V_{2n+1}.

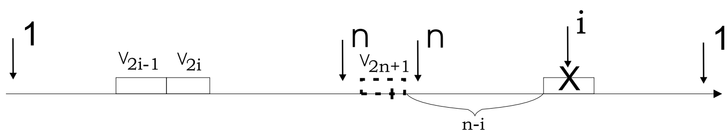

Figure 3: The permutation.

a) Let in the schedule π' the job V_{2n+1} isn't tardy. According to (3.8) $d_{2n+1} - C_{2n+1}(\pi') \leq \frac{1}{2}\delta$ holds, because the schedule π' is canonical LG.

   From Fig. 2 we can look the equation $F(\pi) - F(\pi') = T_{2n}(\pi) + T_{2n+1}(\pi) - (T_{2n}(\pi') + T_{2n+1}(\pi')) = (T_{2n+1}(\pi) - T_{2n+1}(\pi')) - (T_{2n}(\pi') - T_{2n}(\pi)) \geq (p_{2n} - \frac{1}{2}\delta) - p_{2n+1} = p_{2n+1} + b - \frac{1}{2}\delta - p_{2n+1} > 0$ is right.

b) Let in the schedule π' the job V_{2n+1} is tardy.

   $F(\pi) - F(\pi') = T_{2n}(\pi) + T_{2n+1}(\pi) - (T_{2n}(\pi') + T_{2n+1}(\pi')) = p_{2n} - p_{2n+1} = b > 0$.

■

Lemma 7 *Let in the schedule* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *jobs* $\{V_{2i-1},\ V_{2i}\}$, $i < n$, *aren't tardy and on the position* i *("right") the job* $X \in \{V_{2j-1}, V_{2j}\}$, $j \geq i+1$, *is processed,* $|\{\pi_{22}\}| = (i-1)$. *Then for the schedule* $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *we'll have* $F(\pi) > F(\pi')$.

Proof. Let in the schedule π only jobs from $\{\pi_{21},\ X,\ \pi_{22}\}$ are tardy, where $|\{\pi_{22}\}| = (i-1)$. The job X is processed on the position i ("right") (Fig.

3). In canonical LG schedule the job $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$ is processed on the position i ("right").

Construct schedule $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$. According to lemma 3, in all schedules the number of tardy jobs is great or equal n. So the number of tardy jobs following V_{2i} in π' is greater or equal $(n-i)$. Thus,

$$F(\pi) - F(\pi') \geq (p_{2i} - p_X)(n-i) - (d_X - d_{2i}).$$

a) If $X = V_{2j}$ then $p_{2i} - p_X = (j-i)b$,

$d_X - d_{2i} = \sum_{k:=i}^{j-1}(n-k)b - (j-i)\delta = n(j-i)b - \sum_{k:=i}^{j-1} kb - (j-i)\delta = n(j-i)b - i(j-i)b - \sum_{k:=0}^{j-1-i} kb - (j-i)\delta.$

So

$F(\pi) - F(\pi') \geq (j-i)b(n-i) - (n(j-i)b - i(j-i)b - \sum_{k:=0}^{j-1-i} kb - (j-i)\delta) = \sum_{k:=0}^{j-1-i} kb + (j-i)\delta > 0.$

b) If $X = V_{2j-1}$ then
$p_{2i} - p_X = ((j-i)b - \delta_j)$,

$d_X - d_{2i} = \sum_{k:=i}^{j-1}(n-k)b - (j-i)\delta - (n-j)\delta_j - \varepsilon\delta_j = n(j-i)b - \sum_{k:=i}^{j-1} kb - (j-i)\delta - (n-j)\delta_j - \varepsilon\delta_j = n(j-i)b - i(j-i)b - \sum_{k:=0}^{j-1-i} kb - (j-i)\delta - (n-j)\delta_j - \varepsilon\delta_j.$

So

$F(\pi) - F(\pi') \geq ((j-i)b - \delta_j)(n-i) - (n(j-i)b - i(j-i)b - \sum_{k:=0}^{j-1-i} kb - (j-i)\delta - (n-j)\delta_j - \varepsilon\delta_j) = \sum_{k:=0}^{j-1-i} kb + (j-i)\delta - (j-i)\delta_j + \varepsilon\delta_j > 0.$

■

Lemma 8 *Let in the schedule* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *jobs* $\{V_{2i-1}, V_{2i}\}$*,* $i < n$*, aren't tardy and on the position* i *("right") the job* $X \in \{V_{2j-1}, V_{2j}\}$*,* $j < i-1$*, is processed,* $|\{\pi_{22}\}| = (i-1)$*. Then for the schedule* $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *we have* $F(\pi) > F(\pi')$*.*

Proof.

Let in the schedule π only jobs from $\{\pi_{21},\ X,\ \pi_{22}\}$ are tardy, where $|\{\pi_{22}\}| = (i-1)$. The job X is processed on the position i ("right") (Fig. 3). In canonical LG schedule the job $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$ is processed on the position i ("right").

Construct schedule $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$. According to lemma 3, in all schedules the number of tardy jobs is great or equal n. So the number of tardy jobs following V_{2i} in π' is greater or equal $n-i$. Thus,

$$F(\pi) - F(\pi') \geq (d_{2i} - d_X) - (p_X - p_{2i})(n-i+1).$$

a) If $X = V_{2j}$ then $p_X - p_{2i} = (i-j)b$,

$d_{2i} - d_{2j} = \sum_{k:=j}^{i-1}(n-k)b - (i-j)\delta = n(i-j)b - \sum_{k:=j}^{i-1} kb - (i-j)\delta =$
$n(i-j)b - (i-1)(i-j)b + \sum_{k:=0}^{i-1-j} kb - (i-j)\delta.$

So
$F(\pi) - F(\pi') \geq n(i-j)b - (i-1)(i-j)b + \sum_{k:=0}^{i-1-j} kb -$
$(i-j)\delta - (i-j)b(n-i+1) = \sum_{k:=0}^{i-1-j} kb - (i-j)\delta > 0.$

b) If $X = V_{2j-1}$ then $p_X - p_{2i} = (i-j)b + \delta_j$,

$d_{2i} - d_{2j-1} = \sum_{k:=j}^{i-1}(n-k)b - (i-j)\delta + (n-j)\delta_j + \varepsilon\delta_j =$
$n(i-j)b - \sum_{k:=j}^{i-1} kb - (i-j)\delta + (n-j)\delta_j + \varepsilon\delta_j =$
$n(i-j)b - (i-1)(i-j)b + \sum_{k:=0}^{i-1-j} kb - (i-j)\delta + (n-j)\delta_j + \varepsilon\delta_j.$

So
$F(\pi) - F(\pi') \geq n(i-j)b - (i-1)(i-j)b + \sum_{k:=0}^{i-1-j} kb - (i-j)\delta +$
$(n-j)\delta_j + \varepsilon\delta_j - ((i-j)b + \delta_j)(n-i+1) = \sum_{k:=0}^{i-1-j} kb - (i-j)\delta - \delta_j + \varepsilon\delta_j > 0.$

■

Lemma 9 *Let in the schedule* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *jobs* $\{V_{2i-1},\ V_{2i}\}$, $i < n$, *aren't tardy and on the position* i *("right") the job* $X \in \{V_{2(i-1)-1}, V_{2(i-1)}\}$ *is processed,* $|\{\pi_{22}\}| = (i-1)$. *Let in the schedule* $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *the job* Y *are processed on the position* $(n+1)$ *and* $T_Y(\pi') < 2\delta$. *Then we'll have* $F(\pi) > F(\pi')$.

Proof. Let in the schedule π only jobs from $\{\pi_{21},\ X,\ \pi_{22}\}$ are tardy, where $|\{\pi_{22}\}| = i-1$. The job X is processed on the position i ("right") (Fig. 3). In canonical LG schedule the job $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$ is processed on the position i ("right").

Construct schedule $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$. According to lemma 3 in all schedules the number of tardy jobs is great or equal n. So the number of tardy jobs following V_{2i} in π' is greater or equal $(n-i)$. Thus,

$$F(\pi) - F(\pi') > (d_{2i} - d_X) - (p_X - p_{2i})(n-i) - (T_Y(\pi') - T_Y(\pi)) >$$
$$(d_{2i} - d_X) - (p_X - p_{2i})(n-i) - 2\delta.$$

a) If $X = V_{2(i-1)}$ then $p_X - p_{2i} = b$,

$d_{2i} - d_{2i-2} = (n-i+1)b - \delta.$

Thus,
$F(\pi) - F(\pi') > (n-i+1)b - \delta - (n-i)b - 2\delta = b - 3\delta > 0.$

b) If $X = V_{2(i-1)-1}$ then $p_X - p_{2i} = b + \delta_{i-1}$,

$d_{2i} - d_{2i-2} = (n-i+1)b - \delta + (n-i+1)\delta_{i-1} + \varepsilon\delta_{i-1}.$

So
$F(\pi) - F(\pi') > (n-i+1)b - \delta + (n-i+1)\delta_{i-1} + \varepsilon\delta_{i-1} -$
$(n-i)(b+\delta_{i-1}) - 2\delta = b - 3\delta + \delta_{i-1} + \varepsilon\delta_{i-1} > 0$, because $b = n^2\delta$.

■

Theorem 1 *For the case (3) all optimal schedules are canonical LG schedules or can be reduced to canonical LG schedules if EDD rule is applied for first $(n+1)$ jobs.*

Proof. Let π be arbitrary schedule. According to lemma 4 we can reduce to schedule $\pi = (\pi_{EDD}, \pi_{SPT})$ where $|\{\pi_{EDD}\}| = (n+1)$. The job V_{2n+1} is processed on the position $(n+1)$th or $(n+2)$th. Let the schedule π isn't canonical LG schedule.

Then in π two jobs $\{V_{2i-1}, \; V_{2i}\}$, $i < n$ aren't tardy or π has structure (4) (see lemma 5). If (4) holds then according to lemma 6 there exist a canonical LG schedule $\pi' = (V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n+1}, V_{2n}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$ so that $F(\pi) > F(\pi')$. Denote $\pi := \pi'$.

The following algorithm transform a schedule π to a canonical LG schedule. The algorithm consists two cycles.

Denote $\pi' := \pi$

Cycle 1.WHILE in the next schedule π' exist i that on the position i "right" a job $X \notin \{V_{2(i-1)-1}, V_{2(i-1)}\}, X \neq V_{2n+1}$ is processed AND jobs V_{2i-1}, V_{2i} aren't tardy DO

We apply permutation for V_{2i} and X are denoted in lemmas 7 and 8. We have new schedule π'. The total tardiness decreased.

End of cycle 1.

Denote $\pi := \pi'$. Obviously, the step's number of **Cycle 1** is less n. Then to apply the **EDD** rule for first $(n+1)$ jobs in π.

The job V_{2n+1} is processed on the position $(n+1)$ or $(n+2)$ in schedule π. If the job V_{2n+1} is processed on the position $(n+2)$ ("left") then the job V_{2n-1} has the position n and V_{2n} has the position $(n+1)$ according to **Cycle 1** and **EDD** rule.

Following cases are probable:

I. Let the job V_{2n+1} is processed on the position $(n+2)$.

   We consider the schedule $\pi = (\pi_1, V_{2n-1}, V_{2n}, V_{2n+1}, \pi_2)$ where V_{2n} is processed on the $(n+1)$th position. There $|\{\pi_1\}| = (n-1) = |\{\pi_2\}|$ holds.

   According to **Cycle 1** there only situations described in lemma 9 are probable. So $P(\pi_1) + 2qb + \delta > P(\pi_2) > P(\pi_1) + 2qb - \delta$, where q – the number of situations in schedule π.

   For example
   $\{\pi_1\} = \{V_{2i-1}, V_{2i}\} \cup \{V_{1,1}, V_{2,1}, \ldots, V_{i-2,1}, V_{i+1,1}, \ldots, V_{n-1,1}\}$,

   $\{\pi_2\} = \{V_{2(i-1)-1}, V_{2(i-1)}\} \cup \{V_{1,2}, V_{2,2}, \ldots, V_{i-2,2}, V_{i+1,2}, \ldots, V_{n-1,2}\}$.

   Then $q = 1$ and $P(\pi_1) + 2b + \delta > P(\pi_2) > P(\pi_1) + 2b - \delta$ holds, because $-(\delta - \delta_{i-1} - \delta_i - \delta_n) < P(\{V_{1,1}, V_{2,1}, \ldots, V_{i-2,1}, V_{i+1,1}, \ldots, V_{n-1,1}\}) - P(\{V_{1,2}, V_{2,2}, \ldots, V_{i-2,2}, V_{i+1,2}, \ldots, V_{n-1,2}\}) < \delta - \delta_{i-1} - \delta_i - \delta_n$ and

$P(\{V_{2(i-1)-1}, V_{2(i-1)}\}) - P(\{V_{2i-1}, V_{2i}\}) = 2b + \delta_{i-1} - \delta_i.$

Consider two cases when $q = 1$ and $q > 1$.

In the case $q = 0$ we have (4) (see lemma 6).

a) Let $q = 1$.
It's known $\sum_{i:=1}^{2n+1} p_i = 2L + p_{2n+1} = 2L + n^3 b$.

We denote $\Delta = P(\pi_2) - (P(\pi_1) + 2b)$, where $-\delta < \Delta < \delta$.

Let $S = P(\pi_1)$. Then $2S + 2b + \Delta + p_{2n-1} + p_{2n} + p_{2n+1} = 2S + \Delta + 2b + 3n^3 b + 2b + \delta_n = 2L + n^3 b$.
Thus

$$L = S + \frac{1}{2}\Delta + 2b + n^3 b + \frac{1}{2}\delta_n,$$

then

$$C_{2n}(\pi) = P(\pi_1) + p_{2n-1} + p_{2n} = S + 2n^3 b + 2b + \delta_n = L + n^3 b + \frac{1}{2}\delta_n - \frac{1}{2}\Delta.$$

It's known $L + n^3 b = d_{2n+1}$, then $-\delta < C_{2n}(\pi) - d_{2n+1} < \delta$.
There exist two subcases when $C_{2n}(\pi) \geq d_{2n+1}$ and $C_{2n}(\pi) < d_{2n+1}$.

1. $C_{2n}(\pi) \geq d_{2n+1}$.

   For schedule $\pi' = (\pi_1, V_{2n-1}, V_{2n+1}, V_{2n}, \pi_2)$ we have

   $F(\pi) - F(\pi') = T_{2n}(\pi) + T_{2n+1}(\pi) - (T_{2n}(\pi') + T_{2n+1}(\pi')) = (T_{2n+1}(\pi) - T_{2n+1}(\pi')) - (T_{2n}(\pi') - T_{2n}(\pi)) = (p_{2n+1} + (C_{2n}(\pi) - d_{2n+1})) - p_{2n+1} = C_{2n}(\pi) - d_{2n+1} \geq 0$.
2. $C_{2n}(\pi) < d_{2n+1}$.
   And $C_{2n}(\pi) > d_{2n}$ holds, because $d_{2n+1} - d_{2n} = \delta$ and $d_{2n+1} - C_{2n}(\pi) < \delta$.

   Let's describe the schedule π.

   $$\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, V_{2n-1}, V_{2n}, V_{2n+1}, \pi_{21}, X, \pi_{22}),$$

   where $|\{\pi_{22}\}| = (i-1)$, $X \in \{V_{2(i-1)-1}, V_{2(i-1)}\}$.

If $X = V_{2(i-1)-1}$ then permutation of neighboring jobs $V_{2(i-1)-1}$ and $V_{2(i-1)}$ according to **SPT rule** doesn't increase the total tardiness.
Let $X = V_{2(i-1)}$. In π $(n+1)$ jobs are tardy. We construct the schedule

$$\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, V_{2n-1}, V_{2n}, V_{2n+1}, \pi_{21}, V_{2i}, \pi_{22}).$$

There
$F(\pi) - F(\pi') = (d_{2i} - d_{2(i-1)}) - (n-i+1)(p_{2(i-1)} - p_{2i}) = (n-i+1)b - \delta - (n-i+1)b = -\delta$ holds so the total tardiness is increased to δ.
Then $C_{2n}(\pi') - d_{2n+1} > b - \delta$. We construct the schedule

$$\pi'' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, V_{2n-1}, V_{2n+1}, V_{2n}, \pi_{21}, V_{2i}, \pi_{22}).$$

We have $F(\pi') - F(\pi'') > (p_{2n+1} + b - \delta) - p_{2n+1} > b - \delta$.

Then $F(\pi) - F(\pi'') = b - \delta - \delta > 0$.

b) Let $q > 1$. Then $d_{2n} - C_{2n}(\pi) > b - 2\delta$.

If $q = 2$ then in the schedule π' considered in lemma 9, for job $Y = V_{2n}$ we have $T_Y(\pi') < 2\delta$. So we can use the permutation described in lemma 9.

If $q > 2$ then in the schedule π' n jobs are tardy and according to lemma 9 $F(\pi) > F(\pi')$ holds.

II. Let the job V_{2n+1} is processed on the position $(n+1)$. Then from lemma 9 we have $T_Y(\pi') = T_{2n+1}(\pi') < \frac{1}{2}\delta$. So we can use the permutation described in lemma 9.

Cycle 2. WHILE in the next schedule π' exist two jobs V_{2i-1}, V_{2i} so that on the position i ("right") a job $X \in \{V_{2(i-1)-1}, V_{2(i-1)}\}$ is processed AND jobs V_{2i-1}, V_{2i} aren't tardy DO

We apply permutation for V_{2i} and X denoted in cases **I** and **II**. We have a new schedule π'. The total tardiness decreased.

End of cycle 2.

End of algorithm.

So we can transform a schedule π to canonical LG schedule π^* in $O(n)$ time and $F(\pi) > F(\pi^*)$ holds.

$\blacksquare$

Theorem 2 *The modified EOP problem has a solution if and only if in an optimal canonical LG schedule $C_{2n+1}(\pi) = d_{2n+1}$.*

Proof.

Let's consider a canonical LG schedule

$$\pi = (V_{1,1}, V_{2,1}, \dots, V_{i,1}, \dots, V_{n,1}, V_{2n+1}, V_{n,2}, \dots, V_{i,2}, \dots, V_{2,2}, V_{1,2})$$

It's known jobs $V_{n,2}, \dots, V_{i,2}, \dots, V_{2,2}, V_{1,2}$ are tardy. The job V_{2n+1} can be tardy, then $F(\pi) = \sum_{i:=1}^{n} T_{V_{i,2}}(\pi) + T_{V_{2n+1}}(\pi)$.

We denote $G = \sum_{i:=1}^{2n+1} p_i$.

Then

$$\sum_{i:=1}^{n} C_{V_{i,2}}(\pi) = nG - \sum_{i:=1}^{n-1}(n-i)p_{V_{i,2}}.$$

Let's denote

$$\phi(i) = \begin{cases} 1, & V_{i,2} = V_{2i-1}, \\ 0, & V_{i,2} = V_{2i}, \end{cases}$$

then

$$d_{V_{i,2}} = d_{2n+1} - (\sum_{k:=i}^{n-1}(n-k)b + (n-i+1)\delta + \phi(i)((n-i)\delta_i + \varepsilon\delta_i)),$$

So

$$\sum_{i:=1}^{n} T_{V_{i,2}}(\pi) = nG - \sum_{i:=1}^{n-1}(n-i)p_{V_{i,2}} - \sum_{i:=1}^{n}(d_{2n+1} - (\sum_{k:=i}^{n-1}(n-k)b + (n-i+1)\delta + \phi(i)((n-i)\delta_i + \varepsilon\delta_i))).$$

The problem $\min_\pi F(\pi) = \min(\sum_{i:=1}^{n} T_{V_{i,2}}(\pi) + T_{V_{2n+1}}(\pi))$ is reduced to problem $\max \Phi$, where $\Phi = \sum_{i:=1}^{n-1}(n-i)p_{V_{i,2}} - \sum_{i:=1}^{n} \phi(i)((n-i)\delta_i + \varepsilon\delta_i) - T_{V_{2n+1}}(\pi)$.

1. If $V_{i,2} = V_{2i},\ i = 1, \dots, n$ then $T_{V_{2n+1}}(\pi) = \frac{1}{2}\delta$, $\Phi_1 = \sum_{i:=1}^{n-1}(n-i)p_{2i} - \frac{1}{2}\delta$.

2. If $V_{i,2} = V_{2i-1},\ i = 1, \ldots, n$ then $T_{V_{2n+1}}(\pi) = \max\{-\frac{1}{2}\delta, 0\} = 0$,
$\Phi = \sum_{i:=1}^{n-1}(n-i)p_{2i-1} - \sum_{i:=1}^{n}((n-i)\delta_i + \varepsilon\delta_i) = \sum_{i:=1}^{n-1}(n-i)p_{2i} + \sum_{i:=1}^{n-1}(n-i)\delta_i - \sum_{i:=1}^{n}((n-i)\delta_i + \varepsilon\delta_i) = \Phi_1 + \frac{1}{2}\delta - \sum_{i:=1}^{n}\varepsilon\delta_i$.

The function Φ has the maximal value $\Phi_1 + \frac{1}{2}\delta - \frac{1}{2}\sum_{i:=1}^{n}\varepsilon\delta_i$ when $\sum_{i:=1}^{n}\phi(i)(\varepsilon\delta_i) = \frac{1}{2}\sum_{i:=1}^{n}\varepsilon\delta_i$ so $\sum_{i:=1}^{n}\phi(i)\delta_i = \frac{1}{2}\sum_{i:=1}^{n}\delta_i$. So for modified problem there exist two subsets A_1 and A_2 so that $\sum_{a_i\in A_1} a_i = \sum_{a_i\in A_2} a_i$ (the modified EOP problem has a solution). There $C_{2n+1}(\pi) = d_{2n+1}$ holds.

If the modified EOP problem hasn't a solution then $\sum_{i:=1}^{n}\phi(i)\delta_i = \frac{1}{2}\sum_{i:=1}^{n}\delta_i$ doesn't holds. According to value d_{2n+1} we have $C_{2n+1}(\pi) \neq d_{2n+1}$.

If $C_{2n+1}(\pi) = d_{2n+1}$ then $\sum_{i:=1}^{n} p_{V_{i,1}} = \sum_{i:=1}^{n} p_{V_{2i}} + \frac{1}{2}\delta = \sum_{i:=1}^{n} p_{V_{i,2}}$ so the modified EOP problem has a solution. ■

5 Algorithm B-1 for the case (2)

We denote $d_j(t) = d_j - d_n + t,\ j \in 1, \ldots, n$.

Algorithm B-1

1: $\pi_n(t) := (n)$, $F_n^*(t) := \max\{0, p_n - t + t_0\}$;
2: **for** $k = n-1, n-2, \ldots, 1$ **do**
3: $\quad\pi^1 := (k, \pi_{k+1}^*(t - p_k))$;
4: $\quad\pi^2 := (\pi_{k+1}^*(t), k)$;
5: $\quad F(\pi^1) := \max\{0, p_k - d_k(t)\} + F_{k+1}^*(t - p_k)$;
6: $\quad F(\pi^2) := F_{k+1}^*(t) + \max\{0, \sum_{j=k}^{n} p_j - d_k(t)\}$;
7: $\quad F_k^*(t) := \min\{F(\pi^1), F(\pi^2)\}$;
8: $\quad\pi_k^*(t) := \arg\min\{F(\pi^1), F(\pi^2)\}$;
9: **end for**
10: **return** the schedule $\pi_1^*(d_n)$ and its value of the total tardiness $F_1^*(d_n)$.

Notice that lines 1 and 3-8 of the algorithm are performed for each integer t from the interval $[t_0, t_0 + \sum_{j=1}^{n} p_j]$.

Lemma 10 *There exists an optimal schedule π^* for the case (2) where either $(k \to i)_{\pi^*}$ or $(j \to k)_{\pi^*}$ holds for each triad of jobs i, j, k such that $(i \to j)_{\pi^*}$, $\mid d_i - d_j \mid \leq \min\{p_i, p_j\}$, and $k < \min\{i, j\}$.*

Proof. Assuming existence of an optimal schedule $\pi^* = (\pi_1, i, \pi_2, k, \pi_3, j, \pi_4)$ for certain jobs i, j, and k, let construct two schedules $\pi' = (\pi_1, \pi_2, k, i, \pi_3, j, \pi_4)$ and $\pi'' = (\pi_1, i, \pi_2, j, \pi_3, k, \pi_4)$. In the following, we show that either $F(\pi') \leq F(\pi^*)$ or $F(\pi'') \leq F(\pi^*)$ holds.

Since $k < \min\{i, j\}$, it follows that $p_k \geq p_i$, $p_k \geq p_j$, $d_k \leq d_i$, and $d_k \leq d_j$. Let consider three cases.
Case 1: $c_k(\pi^*) \leq d_k$. For the schedule π' we have $c_i(\pi') = c_k(\pi^*) \leq d_k \leq d_i$ and both jobs i and k are early in both schedules π^* and π'. Notice that for each $q \in \{\pi_2\}$ we have $c_q(\pi') \leq c_q(\pi^*)$. This implies $F(\pi') \leq F(\pi^*)$.
Case 2: $c_k(\pi^*) > d_k$ and $c_k(\pi^*) \leq d_j$. Hence, the job k is tardy in π^*; i.e., $T_k(\pi^*) > 0$. Since $\mid d_i - d_j \mid \leq \min\{p_i, p_j\} \leq p_k$ and $c_k(\pi^*) \leq d_j$, it follows that $c_i(\pi^*) \leq c_k(\pi^*) - p_k \leq d_j - p_k \leq d_i$. That means the job i is early in π^*; i.e., $T_i(\pi^*) = 0$. Because of $c_i(\pi') = c_k(\pi^*)$ and $c_k(\pi') = c_k(\pi^*) - p_i$, we have

$$F(\pi') - F(\pi^*) \leq \max\{0, c_k(\pi^*) - d_i\} - \max\{0, c_k(\pi^*) - p_i - d_k\} - (c_k(\pi^*) - d_k) \leq 0.$$

Case 3: $c_k(\pi^*) > d_k$ and $c_k(\pi^*) > d_j$. Hence, the jobs k and j are tardy in the schedule π^* and the job k is tardy in the schedule π''. Additionally, we have $T_j(\pi'') = \max\{0, c_k(\pi^*) - p_k + p_j - d_j\}$. Therefore, $F(\pi'') - F(\pi^*) \leq \max\{0, c_k(\pi^*) - p_k + p_j - d_j\} + c_j(\pi^*) - d_k - c_k(\pi^*) + d_k - c_j(\pi^*) + d_j \leq \max\{0, c_k(\pi^*) - p_k + p_j - d_j\} - c_k(\pi^*) + d_j \leq 0$.

Finally, if $F(\pi') = F(\pi^*)$ or $F(\pi'') = F(\pi^*)$ then either π' or π'' is an optimal schedule too. If $F(\pi') < F(\pi^*)$ or $F(\pi'') < F(\pi^*)$ then we have the contradiction with optimality of π^*. This means that there is no optimal schedule π^* such that $(i \rightarrow k \rightarrow j)_{\pi^*}$ and each optimal schedule has the property proposed in the lemma. The proof is completed. ■

Theorem 3 Algorithm B-1 *constructs an optimal schedule for the case (2) in* $O(n \sum p_j)$ *time.*

Proof. Optimality of **Algorithm B-1** for the case (2) directly follows from Lemma 10. To evaluate complexity of the algorithm, let notice that on each step (for each $k = n, n-1, \ldots, 1$) we need to consider integer points in the interval $[t_0; t_0 + \sum_{j=1}^{n} p_j]$. For certain k and t, each step of the algorithm perform in constant time. Consequently, Algorithm *B-1* constructs the optimal schedule in $O(n \sum p_j)$ time.■

6 Conclusion.

When $p_j \in Z^+$, $j \in N$, for canonical DL instances [1] and case (2) we have exact algorithm **B-1** with $O(n \sum p_j)$ run time. For the special case (3) there exist pseudo-polynomial algorithm **B-1 canonical** with $O(n\delta)$ time.

Algorithm **B-1 modified** has decided instances when $p_j \notin Z^+$, so we can find a solution for not integer EOP problem.

In the conclusion we would like to express next proposition: for any NP-hard case of the problem $1||\sum T_j$ don't exist algorithm with run time less than $O(n\delta)$.

References

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

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

[3] W. Szwarc, F. Della Croce and A. Grosso, Solution of the single machine total tardiness problem, *Journal of Scheduling*, **2**, 55–71 (1999).

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

[5] C.N. Potts and L.N. Van Wassenhove, A decomposition algorithm for the single machine total tardiness problem, *Oper. Res. Lett.*, **1**, 177–182 (1982).

[6] A. Lazarev, A. Kvaratskhelia, A. Tchernykh, Solution algorithms for the total tardiness scheduling problem on a single machine, *Workshop Proceedings of the ENC'04 International Conference*,, 474–480 (2004).

[7] F. Della Croce, A. Grosso, V. Paschos, Lower bounds on the approximation ratios of leading heuristics for the single-machine total tardiness problem, *Journal of Scheduling*, **7**, 85–91 (2004).

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