УДК 519.854.2

# Доказательство NP-трудности частного случая задачи минимизация суммарного запаздывания для одного прибора $1||\sum T_j$

©2006 г. А.А. Лазарев, Е.Р. Гафаров

Москва, ВЦ РАН

Поступила в редакцию 15.12.05

**Аннотация**

Показано, что частный случай **B-1** задачи минимизация суммарного запаздывания для одного прибора $1||\sum T_j$ является NP-трудным в обычном смысле. Для данного случая существует псевдополиномиальный алгоритм трудоемкости $O(n\sum p_j)$. Получено полиномиальное сведение NP - полной проблемы Четно-Нечетного Разбиения к случаю **B-1**.

## Введение

В работе рассматриваются: NP - полная проблема Четно-Нечетного Разбиения (ЧНР) [1] и NP-трудная в обычном смысле задача теории расписаний минимизации суммарного запаздывания для одного прибора [2].

Для исследуемой задачи теории расписаний $1 \ || \ \sum T_j$ Лаулер [3] предложил псевдополиномиальный алгоритм решения общего случая проблемы трудоемкости $O(n^4 \sum p_j)$. Шварц и др. построили [4, 5] алгоритмы решения проблемы, которые были протестированы для примеров $n < 600$ (тестовые примеры Поттса и Ван Вассенхова [6]). Исследование приближенных алгоритмов решения проблемы было проведено в [7], где построены примеры, на которых известные приближенные алгоритмы находят решение с относительной погрешностью порядка размерности примера $n$.

Будет показано, что подслучай **B-1** [8], для которого разработан псевдополиномиальный алгоритм трудоемкости $O(n\sum p_j)$, является NP-трудным в обычном смысле. Приведена полиномиальная схема сведения NP-полной проблемы ЧНР к данному подслучаю.

# 1 Постановка задач.

## 1.1 Задача минимизации суммарного запаздывания для одного прибора $1||\sum T_j$.

Необходимо обслужить $n$ требований на одном приборе. Прерывания при обслуживании и обслуживание более одного требования в любой момент времени запрещены. Для требования $j \in N = \{1, 2, \ldots, n\}$ заданы продолжительность обслуживания $p_j > 0$ и директивный срок его окончания $d_j$, где $N$ – множество требований, которые необходимо обслужить. Задан момент времени $t_0$, с которого прибор готов начать обслуживание требований, поступающих одновременно. Без ограничения общности будем полагать, что $t_0 = 0$. Расписание обслуживания требований $\pi$ строится с момента времени $t_0$ и однозначно задаётся перестановкой элементов множества $N$.

Требуется построить расписание $\pi^*$ обслуживания требований множества $N$, при котором достигается минимум функции $F(\pi) = \sum_{j=1}^{n} \max\{0, c_j(\pi) - d_j\}$, где $c_j(\pi)$ – момент завершения обслуживания требования $j$ при расписании $\pi$. Пусть $\pi = (j_1, j_2, \ldots, j_n)$, тогда $c_{j_1}(\pi) = t_0 + p_{j_1}$ и $c_{j_k}(\pi) = c_{j_{k-1}}(\pi) + p_{j_k}$ для $k = 2, 3, \ldots, n$. Величина $T_j(\pi) = \max\{0, c_j(\pi) - d_j\}$ называется *запаздыванием* требования $j$ при расписании $\pi$, а $F(\pi)$ - *суммарным запаздыванием* требований при расписании $\pi$.

### 1.2 Проблема Четно-Нечетного Разбиения.

Задано упорядоченное множество из $2n$ положительных целых чисел $B = \{b_1, b_2, \ldots, b_{2n}\}$, $b_i > b_{i+1}$, $1 \leq i \leq 2n-1$. Требуется определить, существует ли разбиение множества $B$ на два подмножества $B_1$ и $B_2$, такое, что $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$ и для каждого $i = 1, 2, \ldots, n$ подмножество $B_1$ (следовательно, и $B_2$) содержит в точности один элемент из пары $\{b_{2i-1}, b_{2i}\}$.

Обозначим $\delta_i = b_{2i-1} - b_{2i},\ i = 1, \ldots, n,\ \delta = \sum_{i=1}^n \delta_i$.

Построим модифицированный пример ЧНР.

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

где $b \gg n\delta,\ M \geq n^3 b$.

Очевидно, выполняется $a_i > a_{i+1},\ \forall i = 1, 2, \ldots, 2n-1,\ \delta_i = b_{2i-1} - b_{2i} = a_{2i-1} - a_{2i},\ i = 1, \ldots, n$.

**Лемма 1** *Исходный пример* **ЧНР** *имеет решение "ДА" тогда и только тогда, когда модифицированный пример* **ЧНР** *имеет решение "ДА".*

**Доказательство.** Пусть для исходного примера существует два подмножества $B_1$ и $B_2$, таких что $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$. Возьмём $A_1 = \{a_i | b_i \in B_1\},\ A_2 = \{a_i | b_i \in B_2\}$. Тогда выполняется $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$.

Допустим, что для модифицированного примера имеется два подмножества $A_1$ и $A_2$, таких что $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$. Возьмём $B_1 = \{b_i | a_i \in A_1\},\ B_2 = \{b_i | a_i \in A_2\}$. Очевидно, $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$. ■

# 2 Частные случаи задачи $1 \,||\, \sum T_j$.

Рассмотрим частный случай **B-1** задачи $1 \,||\, \sum T_j$:

$$\begin{cases} p_1 \geq p_2 \geq \ldots \geq p_n, \\ d_1 \leq d_2 \leq \ldots \leq d_n, \\ d_n - d_1 \leq p_n. \end{cases} \tag{2.1}$$

Этот случай относится к так называемым *сложным* примерам из работы [7]. Проведённые ранее эксперименты показали, что для этого частного случая дерево поиска известных алгоритмов [4, 8, 9] имеет наибольшее количество ветвлений [8]. Введем необходимые определения.

Расписание $\pi = (j_1, j_2, \ldots, j_n)$ будем называть *SPT-расписанием* (short processing time), если $p_{j_k} \leq p_{j_{i+k}}$, для $p_{j_k} = p_{j_{k+1}}$ выполняется $d_{j_k} \leq d_{j_{k+1}}$, $k = 1, 2, \ldots, n-1$.

Расписание $\pi = (j_1, j_2, \ldots, j_n)$ будем называть *EDD-расписанием* (early due date), если $d_{j_k} \leq d_{j_{k+1}}$, для $d_{j_k} = d_{j_{k+1}}$ выполняется $p_{j_i} \leq p_{j_{k+1}}$, $k = 1, 2, \ldots, n-1$.

Для случая (2.1) расписание $\pi = (1, 2, \ldots, n)$ является EDD-расписанием. Расписание $\pi = (n, n-1, \ldots, 1)$ – SPT- расписанием.

Расписание $\pi'$ будем называть *частичным расписанием*, если при нём обслуживается некоторое подмножество требований $N' \subset N$. Обозначим $P(N') = \sum_{i \in N'} p_i$, $\{\pi'\} = N'$, $P(\pi') = \sum_{i \in \{\pi'\}} p_i$.

**Лемма 2** [8] *Для случая (2.1) существует оптимальное расписание вида $\pi^* = (\pi_{EDD}, l, \pi_{SPT})$, где $l$ – первое запаздывающее требование при расписании $\pi^*$, $\pi_{EDD}$ и $\pi_{SPT}$ – частичные расписания, построенные по правилам EDD и SPT соответственно.*

**Следствие.** Для случая (2.1) запаздывающие требования в оптимальном расписании расположены в порядке SPT, кроме, быть может, первого запаздывающего требования.

Приведём полиномиальную схему сведения модифицированного примера **ЧНР** к частному случаю (2.1) задачи $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$. Количество требований $(2n+1)$. Требования обозначим следующим образом $V_1, V_2, V_3, V_4, \ldots, V_{2i-1}, V_{2i}, \ldots, V_{2n-1}, V_{2n}, V_{2n+1}$, $N = \{1, 2, \ldots, 2n, 2n+1\}$. Для упрощения записи будем использовать величины $p_{V_i} = p_i$, $d_{V_i} = d_i$, $T_{V_i} = T_i$, $C_{V_i} = C_i$, $i = 1, \ldots, 2n+1$. Пример, удовлетворяющий следующим ограничениям, назовем *каноническим LG-примером.*

$$
\begin{cases}
p_1 > p_2 > \ldots > p_{2n+1}, \\
d_1 < d_2 < \ldots < d_{2n+1}, \\
d_{2n+1} - d_1 < p_{2n+1}, \\
p_{2n+1} = M = n^3 b, \\
p_{2n} = p_{2n+1} + b = a_{2n}, \\
p_{2i} = p_{2i+2} + b = a_{2i},\ i = n-1, \ldots, 1, \\
p_{2i-1} = p_{2i} + \delta_i = a_{2i-1},\ i = n, \ldots, 1, \\
d_{2n+1} = \sum_{i:=1}^{n} p_{2i} + p_{2n+1} + \frac{1}{2}\delta, \\
d_{2n} = d_{2n+1} - \delta, \\
d_{2i} = d_{2i+2} - (n-i)b + \delta,\ i = n-1, \ldots, 1, \\
d_{2i-1} = d_{2i} - (n-i)\delta_i - \varepsilon\delta_i,\ i = n, \ldots, 1,
\end{cases} \tag{2.2}
$$

где $b = n^2\delta$, $0 < \varepsilon < \frac{\min_i \delta_i}{\max_i \delta_i}$.

Директивные сроки примера (2.2) представлены на рис. 1.

Обозначим $L = \frac{1}{2}\sum_{i:=1}^{2n} p_i$, тогда $d_{2n+1} = L + p_{2n+1}$, так как $\frac{1}{2}\sum_{i:=1}^{2n} p_i = \sum_{i:=1}^{n} p_{2i} + \frac{1}{2}\delta$.

Необходимо отметить, что канонические примеры DL из [2] не удовлетворяют случаю (2.2). Первые три неравенства показывают, что (2.2) является подслучаем (2.1).

# 3 Свойства примеров случая (2.2) проблемы $1||\sum T_j$.

Сформулируем следующую лемму.

**Лемма 3** *Для (2.2) при любом расписании количество запаздывающих требований равно или $n$, или $n+1$.*

**Доказательство.** Доказательство проведем в два этапа.

1. Рассмотрим подмножество требований $N'$ состоящее из $(n+2)$ самых коротких по продолжительности обслуживания требований

и упорядочим их вначале расписания. Очевидно, что $\sum_{i\in N'} p_i > (n+2)p_{min} = (n+2)n^3b$, где $p_{min} = \min_{j\in N}\{p_j\} = p_{2n+1}$.

Из условий 4-8 в (2.2) найдём

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

для которого будет выполняться

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

т.е. при любом расписании требование, обслуживаемое $n+2$ по порядку и все последующие будут запаздывать, так как согласно третьему неравенству в (2.2), разность между директивными сроками любых двух требований меньше продолжительности обслуживания каждого требования. Таким образом, при любом расписании $\pi$ количество незапаздывающих требований не превышает $n+1$.

2. Рассмотрим подмножество $N''$, состоящее из $n$ самых длинных по продолжительности обслуживания требований, и упорядочим их вначале расписания. Возможны два случая:

   а) пусть $n = 2k$, тогда $N'' = \{V_1, V_2, \ldots, V_{2k-1}, V_{2k}\}$, будем иметь

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

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

   Из условий 8-11 в (2.2) будем иметь

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

b) пусть $n = 2k+1$, тогда $N'' = \{V_1, V_2, \ldots, V_{2k-1}, V_{2k}, V_{2(k+1)-1}\}$.

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

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

$d_{min} = d_1 = d_{2n+1} - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 - \varepsilon\delta_1) = (n+1)n^3 b + (b + 2b + \ldots + nb) + \frac{1}{2}\delta - (\sum_{i:=1}^{n-1}((n-i)b - \delta) + \delta + (n-1)\delta_1 + \varepsilon\delta_1) > P(N'')$,

т.е. при любом расписании первые $n$ требований не будут запаздывать. Таким образом, при любом расписании $\pi$ количество незапаздывающих требований больше или равно $n$.

Следовательно, для случая (2.2) при любом расписании количество запаздывающих требований равно или $n$, или $n+1$. ■

**Лемма 4** *Для случая (2.2) для каждого расписания* $\pi = (\pi_1, \pi_2)$ *существует расписание вида* $\pi' = (\pi_{EDD}, \pi_{SPT})$, *где* $\{\pi_1\} = \{\pi_{EDD}\}$, $\{\pi_2\} = \{\pi_{SPT}\}$, $|\{\pi_1\}| = n+1$, $|\{\pi_2\}| = n$. *Причём,* $F(\pi) \geq F(\pi')$.

**Доказательство.**

Рассмотрим частичное расписание $\pi_1$. Так как $n$ первых требований при $\pi_1$ не запаздывают, а может запаздывать лишь последнее, то EDD порядок обслуживания требований множества $\{\pi_1\}$ будет оптимальным для него. В этом случае на $n+1$ месте будет обслуживаться требование $j = argmax\{d_i : i \in \{\pi_1\}\}$.

Рассмотрим частичное расписание $\pi_2$. Так как все $n$ требований при $\pi_2$ запаздывают, то порядок SPT обслуживания требований множества $\{\pi_2\}$ будет оптимальным. ■

Расписание $(V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n,1}, V_{2n+1}, V_{n,2}, \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$ будем называть *каноническим LG-расписанием*, где $\{V_{i,1}, V_{i,2}\} = \{V_{2i-1}, V_{2i}\},\ i = 1, 2, \ldots, n$.

**Лемма 5** *Если расписание $\pi = (\pi_1, \pi_2)$, $|\{\pi_1\}| = (n+1)$, $|\{\pi_2\}| = n$, неканоническое LG или не может быть преобразовано к таковому применением правил EDD и SPT к частичным расписаниям $\pi_1$ и $\pi_2$ соответственно, то при расписании $\pi$ некоторая пара требований $\{V_{2i-1},\ V_{2i}\}$, $i < n$, не запаздывает или расписание $\pi$ имеет вид*

$$(V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n}, V_{2n+1}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2}), \tag{3.1}$$

*т.е. пара требований $\{V_{2n-1}, V_{2n}\}$ обслуживается до требования $V_{2n+1}$, одно требование из каждой пары $\{V_{2i-1}, V_{2i}\}$, $i = 1, \ldots, n-1$, обслуживается до требования $V_{2n+1}$, другое требование из пары после $V_{2n+1}$, а требования $V_{2n+1}$ обслуживается $n+2$ по порядку.*

**Доказательство.** Рассмотрим некоторое расписание $\pi = (\pi_1, \pi_2)$, где $|\{\pi_1\}| = n+1$, $|\{\pi_2\}| = n$. Проанализируем возможные случаи.

1. Если $\{\pi_2\} = \{V_{1,2}, \ldots, V_{n,2}\}$, т.е. в $\pi_2$ упорядочены $n$ требований по одному требованию из всех $n$ пар $\{V_{2i-1}, V_{2i}\}$, $i = 1, \ldots, n$. Упорядочим требования из $\pi_2$ по правилу SPT, а требования из $\pi_1$ по правилу EDD. Тогда получим каноническое расписание $\pi'$, причём $F(\pi') \le F(\pi)$, согласно лемме 4.

2. Если $\{\pi_2\} \ne \{V_{1,2}, \ldots, V_{n,2}\}$. То возможны ситуации:

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

   b) существует пара требований $\{V_{2j-1}, V_{2j}\} \subset \{\pi_2\}$.

Тогда, учитывая, что $|\{\pi_2\}| = n$, для некоторого $i$ будем иметь $\{V_{2i-1}, V_{2i}\} \subset \{\pi_1\}$. ■

Далее в теореме 1 будет показано, что для случая (2.2) все оптимальные расписания являются каноническими LG. Доказано, что произвольное неканоническое LG-расписание $\pi$ может быть преобразовано к каноническому LG $\pi'$, причем $F(\pi) > F(\pi')$. При доказательстве теоремы используются выводы лемм 6- 9.

**Лемма 6** *Пусть расписание $\pi$ имеет вид (3.1), при котором требование $V_{2n+1}$ обслуживается на $n+2$ по порядку месте. Тогда для канонического LG расписания $\pi' = (V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n-1,1}, V_{2n-1}, V_{2n+1}, V_{2n}, V_{n-1,2} \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$ выполняется $F(\pi) > F(\pi')$.*

**Доказательство.** При расписании $\pi$ требование $V_{2n-1}$ обслуживается на позиции $n$ "слева". Согласно лемме требование $V_{2n-1}$ не запаздывает, так как оно обслуживается $n$-ым по порядку. Требование $V_{2n+1}$ обслуживается на позиции $n+2$ "слева" поэтому запаздывает.

Для требований множества $\{V_2, V_4, \ldots, V_{2i}, \ldots, V_{2n-2}, V_{2n-1}\}$ имеем

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

согласно условию 8 из (2.2). Далее, очевидно,

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

поэтому

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

Следовательно, требование $V_{2n}$, которое обслуживается на $n+1$ по порядку месте, при расписании $\pi$ запаздывает.

Расписание $\pi$ можно записать как $\pi = (\pi_{11}, V_{2n}, V_{2n+1}, \pi_{21})$. Рассмотрим каноническое LG-расписание $\pi' = (\pi_{11}, V_{2n+1}, V_{2n}, \pi_{21})$. Покажем, что $F(\pi) > F(\pi')$. Выделим два случая.

1. Пусть при расписании $\pi'$ требование $V_{2n+1}$ не запаздывает. Тогда из условия 8 в (2.2) следует, что $d_{2n+1} - C_{2n+1}(\pi') \le \frac{1}{2}\delta$, так как расписание $\pi'$ каноническое.

   Из рис. 2 видно, что

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

2. Пусть при расписании $\pi'$ требование $V_{2n+1}$ запаздывает, тогда

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

**Лемма 7** *Пусть при некотором расписании* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *пара требований* $\{V_{2i-1}, \ V_{2i}\}$, $i < n$, *не запаздывает, а на позиции* $i$ *"справа" обслуживается требование* $X \in \{V_{2j-1}, V_{2j}\}$, $j \ge i+1$. *Тогда для расписания* $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *выполняется* $F(\pi) > F(\pi')$.

**Доказательство.**

Пусть при расписании $\pi$ запаздывают требования только из множества $\{\pi_{21}, \ X, \ \pi_{22}\}$, где $|\{\pi_{22}\}| = i - 1$. Требование $X$ занимает позицию с номером $i$ "справа" (см. рис. 3), в которой при каноническом расписании будет обслуживаться требование $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$.

Построим расписание $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$. При обоих расписаниях запаздывает не меньше $n$ требований. Поэтому при расписании $\pi'$ перед требованием $V_{2i}$ будет запаздывать не меньше, чем $n - i$ требований (и не больше, чем $n - i + 1$), согласно лемме 3. Поэтому

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

Возможны следующие ситуации.

1. Если $X = V_{2j}$, то $p_{2i} - p_X = (j-i)b$,

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

   Следовательно,

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

2. Если $X = V_{2j-1}$, то
   $p_{2i} - p_X = ((j-i)b - \delta_j)$,

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

   Следовательно,

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

■

**Лемма 8** *Пусть при некотором расписании* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *пара требований* $\{V_{2i-1},\ V_{2i}\}$, $i < n$, *не запаздывает, а на позиции* $i$ *"справа" обслуживается требование* $X \in \{V_{2j-1}, V_{2j}\}$, $j < i-1$. *Тогда для расписания* $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *выполняется* $F(\pi) > F(\pi')$.

**Доказательство.**

Допустим, что при расписании $\pi$ запаздывают требования только из множества $\{\pi_{21},\ X,\ \pi_{22}\}$, где $|\{\pi_{22}\}| = i-1$. Требование $X$ занимает позицию $i$ "справа" (рис. 3), в которой при каноническом расписании будет обслуживаться требование $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$.

Построим расписание $\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$. При обоих расписаниях запаздывает не меньше $n$ требований. Поэтому при расписании $\pi'$ перед требованием $V_{2i}$ будет запаздывать не меньше, чем $n-i$ требований (и не больше, чем $n-i+1$), согласно лемме 4. Поэтому

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

По аналогии с доказательством леммы 7 выделим два случая.

1. Если $X = V_{2j}$, то $p_X - p_{2i} = (i-j)b$,

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

   Следовательно,

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

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

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

   Тогда имеем

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

   ■

**Лемма 9** *Пусть при некотором расписании* $\pi = (\pi_{11}, V_{2i-1}, V_{2i}, \pi_{12}, \pi_{21}, X, \pi_{22})$ *пара требований* $\{V_{2i-1}, V_{2i}\}$, $i < n$, *не запаздывает, а на позиции* $i$ *"справа" обслуживается требование* $X \in \{V_{2(i-1)-1}, V_{2(i-1)}\}$. *Предположим, что для расписания*

$\pi' = (\pi_{11}, V_{2i-1}, X, \pi_{12}, \pi_{21}, V_{2i}, \pi_{22})$ *требование* $Y$ *занимает позицию* $(n+1)$ *и* $T_Y(\pi') < 2\delta$. *Тогда выполняется* $F(\pi) > F(\pi')$.

**Доказательство.**

Пусть при расписании $\pi$ запаздывают требования только из множества $\{\pi_{21},\ X,\ \pi_{22}\}$, где $|\{\pi_{22}\}| = i-1$. Требование $X$ занимает позицию $i$ "справа" (рис. 3), в которой при каноническом расписании будет обслуживаться требование $V_{i,2} \in \{V_{2i-1}, V_{2i}\}$.

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

При расписании $\pi$ перед требованием $V_{2i}$ будет запаздывать не меньше $n-i$ требований. Поэтому

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

$$(d_{2i} - d_X) - (p_X - p_{2i})(n-i) - 2\delta.$$

Возможны ситуации:

1. $X = V_{2(i-1)}$, т.е. $p_X - p_{2i} = b$,

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

   Поэтому

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

2. $X = V_{2(i-1)-1}$, т.е. $p_X - p_{2i} = b + \delta_{i-1}$,

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

   Следовательно,

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

Результаты леммы 9 используются для доказательства теоремы 1. При этом имеет место $T_Y(\pi') < 2\delta$. Ситуация $T_Y(\pi') \geq 2\delta$ нами не рассматривается, так как она не встречается.

На основе полученных лемм докажем следующую теорему.

**Теорема 1** *Для случая (2.2) все оптимальные расписания являются каноническими или могут быть преобразованы к таковым применением правила EDD к $n+1$ требованиям, обслуживаемым в начале расписания.*

**Доказательство.**

Пусть $\pi$ - произвольное расписание. Согласно лемме 4 можно рассматривать только расписание вида $\pi = (\pi_{EDD}, \pi_{SPT})$, где $|\{\pi_{EDD}\}| = n+1$. При этом расписании требование $V_{2n+1}$ занимает или позицию $(n+1)$, или позицию $n+2$. Допустим, что расписание $\pi$ не каноническое.

Тогда в силу леммы 6 имеем при расписании $\pi$ пару незапаздывающих требований $\{V_{2i-1},\ V_{2i}\}$, $i < n$, или расписание $\pi$ имеет вид (3.1), при котором требование $V_{2n+1}$ обслуживается на $n+2$ месте.

Если расписание имеет вид (3.1), то по лемме 6 для канонического расписания $\pi' = (V_{1,1}, V_{2,1}, \dots, V_{i,1}, \dots, V_{n-1,1}, V_{2n-1}, V_{2n+1}, V_{2n}, V_{n-1,2} \dots, V_{i,2}, \dots, V_{2,2}, V_{1,2})$. Выполняется $F(\pi) > F(\pi')$. Переобозначим $\pi = \pi'$.

Далее опишем алгоритм, состоящий из двух циклов преобразования исходного расписания $\pi$ к каноническому виду.

**Цикл 1.** Данный цикл продолжается, пока среди незапаздывающих требований при очередном расписании $\pi$ присутствует пара $V_{2i-1}, V_{2i}$, причем на позиции $i$ "справа" обслуживается требование $X \notin \{V_{2(i-1)-1}, V_{2(i-1)}\}, X \neq V_{2n+1}$. Применяем для требований $V_{2i}$ и $X$ перестановку, описанную в леммах 7 и 8. В результате значение целевой функции уменьшится.

**Конец цикла 1.** Переобозначим $\pi := \pi'$.

Количество шагов в цикле 1, очевидно, не превышает $n$.

Первые $n+1$ требований при расписании $\pi$ упорядочим по правилу **EDD**.

Требование $V_{2n+1}$ занимает позицию $n+1$ или позицию $n+2$ при расписании $\pi$. Если $V_{2n+1}$ занимает позицию $n+2$ "слева" , то позиции $n$ и $n+1$ "слева" занимают требования $V_{2n-1}$ и $V_{2n}$ соответственно, согласно **Циклу 1** и сортировке по правилу EDD.

Рассмотрим две возможные ситуации.

Ситуация I. Пусть требование $V_{2n+1}$ занимает позицию $n+2$.

Рассматривается расписание $\pi = (\pi_1, V_{2n-1}, V_{2n}, V_{2n+1}, \pi_2)$, при котором требование $V_{2n}$ обслуживается $n+1$ по порядку.

При частичных расписаниях $\pi_1$ и $\pi_2$ обслуживается по $(n-1)$ требованию, т.е. $|\{\pi_1\}| = n-1 = |\{\pi_2\}|$.

Учитывая, что после выполнения **Цикла 1** останутся только ситуации описанные в лемме 9, поэтому $P(\pi_1) + 2qb + \delta > P(\pi_2) > P(\pi_1) + 2qb - \delta$, где $q$ – количество ситуаций представленные в лемме 10 при расписании $\pi$.

**Примером** подобной ситуации может служить расписание, при котором $\{\pi_1\} = \{V_{2i-1}, V_{2i}\} \cup \{V_{1,1}, V_{2,1}, \ldots, V_{i-2,1}, V_{i+1,1}, \ldots, V_{n-1,1}\}$,

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

Тогда $q = 1$ и $P(\pi_1) + 2b + \delta > P(\pi_2) > P(\pi_1) + 2b - \delta$, так как

$-(\delta - \delta_{i-1} - \delta_i - \delta_n) < P(\{V_{1,1}, V_{2,1}, \ldots, V_{i-2,1}, V_{i+1,1}, \ldots, V_{n-1,1}\}) - P(\{V_{1,2}, V_{2,2}, \ldots, V_{i-2,2}, V_{i+1,2}, \ldots, V_{n-1,2}\}) < \delta - \delta_{i-1} - \delta_i - \delta_n$ и

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

Выделим два случая, когда $q = 1$ и $q > 1$. При $q = 0$ расписание $\pi$ имеет вид (3.1) (см. лемму 6). Этот случай мы рассмотрели выше.

Пусть $q = 1$.

Известно, что $\sum_{i:=1}^{2n+1} p_i = 2L + p_{2n+1} = 2L + n^3 b$.

Обозначим $\Delta = P(\pi_2) - (P(\pi_1) + 2b)$, где $-\delta < \Delta < \delta$.

Примем $S = P(\pi_1)$, тогда $2S + 2b + \Delta + p_{2n-1} + p_{2n} + p_{2n+1} = 2S + \Delta + 2b + 3n^3b + 2b + \delta_n = 2L + n^3b$.

Поэтому

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

а значит

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

Известно, что $L + n^3b = d_{2n+1}$, тогда $-\delta < C_{2n}(\pi) - d_{2n+1} < \delta$.

Необходимо рассмотреть два подслучая.

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

   Рассмотрим расписание $\pi' = (\pi_1, V_{2n-1}, V_{2n+1}, V_{2n}, \pi_2)$.

   $F(\pi) - F(\pi') = T_{2n}(\pi) + T_{2n+1}(\pi) - (T_{2n}(\pi') + T_{2n+1}(\pi')) = (T_{2n+1}(\pi) - T_{2n+1}(\pi')) - (T_{2n}(\pi') - T_{2n}(\pi)) = (p_{2n+1} + (C_{2n}(\pi) - d_{2n+1})) - p_{2n+1} = C_{2n}(\pi) - d_{2n+1} \geq 0$.

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

   При этом $C_{2n}(\pi) > d_{2n}$, так как $d_{2n+1} - d_{2n} = \delta$ и $d_{2n+1} - C_{2n}(\pi) < \delta$.

   Расписание $\pi$ имеет следующую структуру:

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

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

   Если $X = V_{2(i-1)-1}$, то транспозиция соседних требований $V_{2(i-1)-1}$ и $V_{2(i-1)}$ согласно правилу SPT не увеличит значение целевой функции.

   Пусть $X = V_{2(i-1)}$. При расписании $\pi$ запаздывает $n+1$ требование.

   Построим расписание

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

Выполняется

$F(\pi) - F(\pi') = (d_{2i} - d_{2(i-1)}) - (n - i + 1)(p_{2(i-1)} - p_{2i}) = (n - i + 1)b - \delta - (n - i + 1)b = -\delta$, т.е. значение целевой функции увеличилось на $\delta$.

Тогда $C_{2n}(\pi') - d_{2n+1} > b - \delta$. Сформируем расписание

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

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

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

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

Если $q = 2$ при расписании $\pi'$, рассмотренном в лемме 9, для требования $Y = V_{2n}$ выполняется $T_Y(\pi') < 2\delta$. Другими словами, перестановка, описанная в лемме 9, уменьшит значение целевой функции.

Когда $q > 2$ при расписании $\pi'$ будет запаздывать $n$ требований, поэтому, согласно лемме 9, имеет место неравенство $F(\pi) > F(\pi')$

Ситуация II. Предположим, что требование $V_{2n+1}$ занимает позицию $n+1$. Тогда в ситуации, описанной в лемме 10, $T_Y(\pi') = T_{2n+1}(\pi') < \frac{1}{2}\delta$. Преобразование расписания согласно лемме 10 уменьшит значение целевой функции.

**Цикл 2.** Выполняется, пока среди незапаздывающих требований при очередном расписании $\pi$ присутствует пара $V_{2i-1}, V_{2i}$, причем на позиции $i$ "справа" обслуживается требование $X \in \{V_{2(i-1)-1}, V_{2(i-1)}\}$. Применяем для требований $X$ и $V_{2i}$ перестановку, описанную для ситуаций **I** и **II**. При этом значение целевой функции уменьшается.

**Конец Цикла 2** и **алгоритма преобразования исходного неканонического расписания.**

Таким образом, произвольное неканоническое расписание $\pi$ за время $O(n)$ можно преобразовать к каноническому $\pi^*$ . Причем $F(\pi) > F(\pi^*)$. ■

**Теорема 2** *Решение примера ЧНР будет "ДА" тогда и только тогда, когда при оптимальном каноническом расписании* $C_{2n+1}(\pi) = d_{2n+1}$.

**Доказательство.**

Рассмотрим каноническое расписание вида

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

Известно, что требования $V_{n,2}, \dots, V_{i,2}, \dots, V_{2,2}, V_{1,2}$ запаздывают. Требование $V_{2n+1}$ может запаздывать, тогда $F(\pi) = \sum_{i:=1}^{n} T_{V_{i,2}}(\pi) + T_{V_{2n+1}}(\pi)$.

Обозначим $\sum_{i:=1}^{2n+1} p_i = C$.

Тогда

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

Введем функцию

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

имеем

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

поэтому

$$\sum_{i:=1}^{n} T_{V_{i,2}}(\pi) = nC - \sum_{i:=1}^{n-1} (n-i) p_{V_{i,2}} -$$

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

Задача $\min_\pi F(\pi) = \min(\sum_{i:=1}^{n} T_{V_{i,2}}(\pi) + T_{V_{2n+1}}(\pi))$ сводится к максимизации функции $\Phi$, где $\Phi = \sum_{i:=1}^{n-1} (n-i) p_{V_{i,2}} - \sum_{i:=1}^{n} \phi(i)((n-i)\delta_i + \varepsilon\delta_i) - T_{V_{2n+1}}(\pi)$.

Рассмотрим два возможных случая.

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

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

Функция $\Phi$ достигает максимальное значение $\Phi_1 + \frac{1}{2}\delta - \frac{1}{2}\sum_{i:=1}^{n}\varepsilon\delta_i$, когда $\sum_{i:=1}^{n}\phi(i)(\varepsilon\delta_i) = \frac{1}{2}\sum_{i:=1}^{n}\varepsilon\delta_i$, что равнозначно $\sum_{i:=1}^{n}\phi(i)\delta_i = \frac{1}{2}\sum_{i:=1}^{n}\delta_i$. Следовательно, для модифицированного примера существуют два подмножества $A_1$ и $A_2$, таких что $\sum_{a_i\in A_1} a_i = \sum_{a_i\in A_2} a_i$ (ответ "ДА"). При этом $C_{2n+1}(\pi) = d_{2n+1}$.

Если решение модифицированного примера ЧНР "НЕТ", то не выполняется равенство $\sum_{i:=1}^{n}\phi(i)\delta_i = \frac{1}{2}\sum_{i:=1}^{n}\delta_i$. Учитывая значение $d_{2n+1}$, будем иметь $C_{2n+1}(\pi) \neq d_{2n+1}$

Если $C_{2n+1}(\pi) = d_{2n+1}$, то из этого следует, что $\sum_{i:=1}^{n}p_{V_{i,1}} = \sum_{i:=1}^{n}p_{V_{2i}} + \frac{1}{2}\delta = \sum_{i:=1}^{n}p_{V_{i,2}}$, т.е. решение модифицированного примера **ЧНР** будет иметь ответ "ДА". $\blacksquare$

# Заключение.

В заключение стоит отметить, что для решения примеров (2.1) и (2.2) существует псевдополиномиальный алгоритм трудоёмкости $O(n\sum p_j)$ [8]. Предложен алгоритм **B-1 канонический** трудоемкости $O(n\delta)$ решения канонических примеров DL [2] и LG. Известен алгоритм **B-1 модифицированный**, который решает примеры в случае $p_j > 0$, т.е. исходные параметры проблем ЧНР и $1||\sum T_j$ могут быть нецелочисленными.

# Список литературы


[1] Гэри М., Джонсон Д. *Вычислительные машины и трудно решаемые задачи*// М. Мир. 1982.

[2] Du J., Leung J. Y.-T. *Minimizing total tardiness on one processor is NP-hard* // Math. Oper. Res.. 1990. № 15.

[3] Lawler E.L. *A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness* // Ann. Discrete Math. 1977. № 1.

[4] Szwarc W., Della Croce F., Grosso A. *Solution of the single machine total tardiness problem* // J. of Scheduling. 1999. № 2.

[5] Szwarc W., Grosso A., Della Croce F. *Algorithmic paradoxes of the single machine total tardiness problem* // J. of Scheduling. 2001. № 4.

[6] Potts C.N., Van Wassenhove L.N. *A decomposition algorithm for the single machine total tardiness problem* // Oper. Res. Lett.. 1982. № 1.

[7] Della Croce F., Grosso A., Paschos V. *Lower bounds on the approximation ratios of leading heuristics for the single-machine total tardiness problem* // J. of Scheduling. 2004. № 7.

[8] Lazarev A., Kvaratskhelia A., Tchernykh A. *Solution algorithms for the total tardiness scheduling problem on a single machine* // Workshop Proceedings of the ENC'04 International Conference. Colima, Mexico, 2004.

[9] Chang S., Lu Q., Tang G. et al. *On decomposition of total tardiness problem* // Oper. Res. Lett.. 1995. № 17.

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

**Authors A. A. Lazarev, E. R. Gafarov.**

In this paper we show that the special case **B-1** of the single machine total tardiness problem $1||\sum T_j$ is NP-hard in ordinary sense. For the case there exist pseudo-polynomial algorithm $O(n\sum p_j)$ time.

## Список рисунков к статье Лазарева А.А. и др.

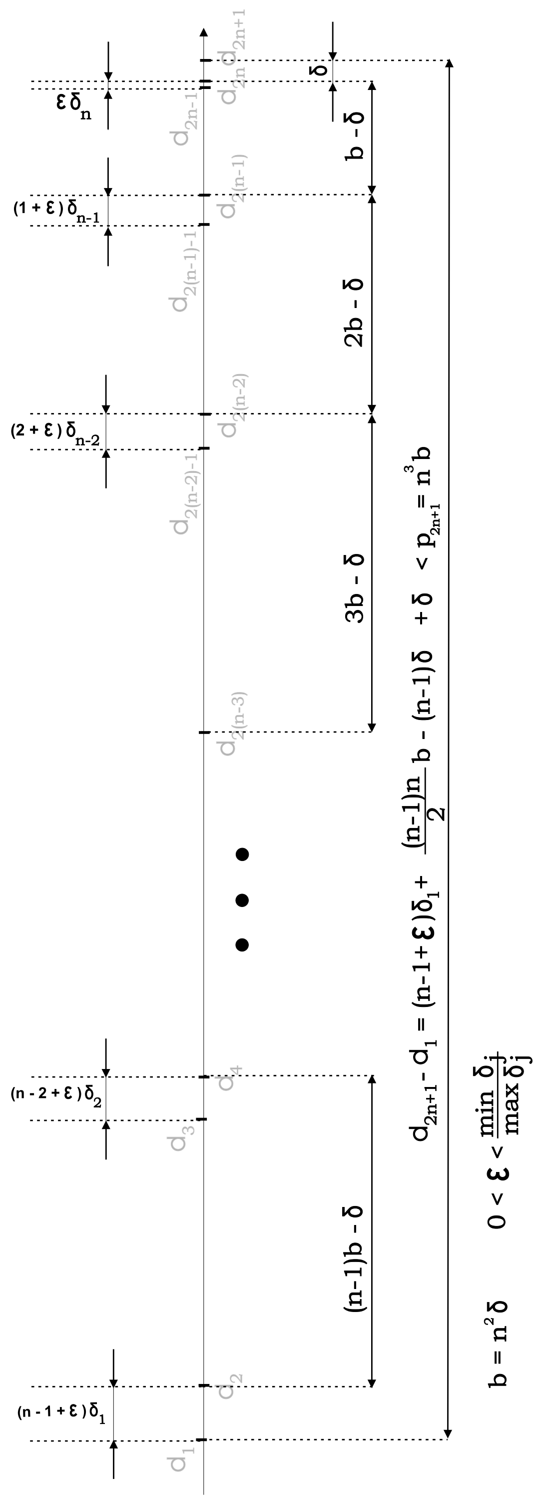


Рис. 1: Директивные сроки канонического LG примера.

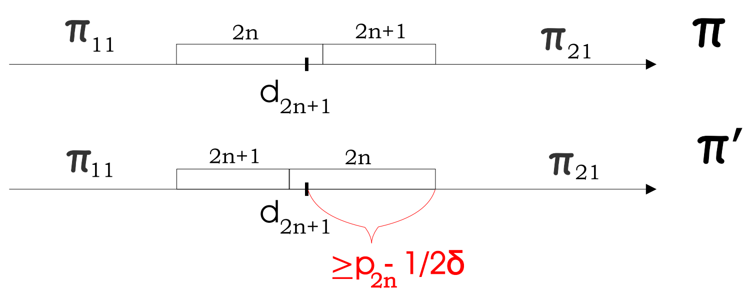


Рис. 2: Перестановка требований $V_{2n}$ и $V_{2n+1}$.

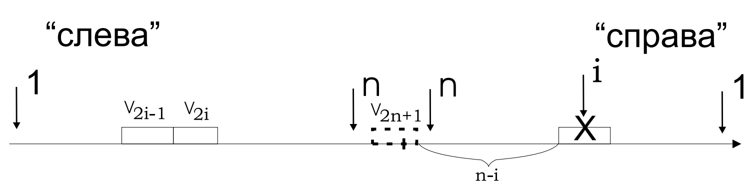


Рис. 3: Перестановка в неканоническом расписании.