А. А. Лазарев, Е. Р. Гафаров

**Аннотация**

Показано, что трудоемкость известных алгоритмов [3, 6, 7] решения задачи $1 \,||\, \sum T_j$ не меньше $O(n2^{(n-1)/3-1})$ для канонических примеров и не меньше $O(n2^{(n-1)/2})$ для примеров случая **BF**. Предложены алгоритмы решения канонических примеров и примеров случая **BF**. Приводится экспериментальная оценка их трудоёмкости.

# 1 Введение

В работе рассматривается NP-трудная в обычном смысле проблема теории расписаний 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 $j \in N = \{1, 2, \dots, n\}$ заданы продолжительность обслуживания $p_j > 0$, $p_j \in Z^+$, и директивный срок окончания обслуживания $d_j$, где $N$ - множество требований. Задан момент освобождения прибора $t_0$, с которого прибор готов начать обслуживание требований. Все требования поступают на обслуживание одновременно в момент времени $t_0$. 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 $\pi = (j_1, j_2, \dots, j_n)$, тогда $c_{j_1}(\pi) = t_0 + p_{j_1}$ и $c_{j_k}(\pi) = c_{j_{k-1}}(\pi) + p_{j_k}$ для $k = 2, 3, \dots, n$. 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 $\pi$.

Исследуемая проблема является NP-трудной в обычном смысле [1]. Лаулер предложил [2] псевдополиномиальный алгоритм решения общего случая проблемы трудоёмкости $O(n^4 \sum p_j)$. Шварц и др. построили [3] алгоритмы решения проблемы, которые были протестированы для примеров $n < 600$ (тестовые примеры Поттса и Ван Вассенхова [4]). Исследование приближенных алгоритмов решения проблемы было проведено в работе [5], где построены примеры, на которых известные приближенные алгоритмы находят решение с относительной погрешностью порядка размерности примера $n$.

Будет показано, что основываясь на известных правилах сокращения перебора расписаний [2, 3, 6, 7] невозможно построить алгоритм, который за приемлемое время (за полином от размерности задачи) находит оптимальное расписание для достаточно больших $n$. То есть все известные алгоритмы, основанные на вышеперечисленных правилах будут работать экспоненциально долго, в частности на *канонических примерах*, примерах случая **BF**, для которых предлагаются псевдополиномиальный $O(n \sum p_j)$ и полиномиальный алгоритм $O(n^2)$.

## 2 Канонические примеры.

*Проблема Чётно-Нечётное Разбиение (ПЧНР):* Задано упорядоченное множество из $2\bar{n}$ положительных целых чисел $B = \{b_1, b_2, \ldots, b_{2\bar{n}}\}$, $b_i > b_{i+1}$, $1 \le i < 2\bar{n}$. 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 $i = 1, 2, \ldots, \bar{n}$ подмножество $B_1$ (следовательно, и $B_2$) содержит в точности один элемент из пары $\{b_{2i-1}, b_{2i}\}$.

Приведем полиномиальную схему сведения **ПЧНР** к проблеме $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$ [1]. Пусть $a_{2i-1} = b_{2i-1} + (9\bar{n}^2 + 3\bar{n} - i + 1)\delta + 5\bar{n}(b_1 - b_{2\bar{n}})$ и $a_{2i} = b_{2i} + (9\bar{n}^2 + 3\bar{n} - i + 1)\delta + 5\bar{n}(b_1 - b_{2\bar{n}})$, $i = 1, \ldots, \bar{n}$, где $\delta = \frac{1}{2}\sum_{i=1}^{\bar{n}}(b_{2i-1} - b_{2i})$.

Обозначим $\delta_i = b_{2i-1} - b_{2i}$.

Построим *канонический пример* [1] проблемы $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$ для множества из $n = 3\bar{n}+1$ требований $N = \{V_1, V_2 \ldots, V_{2\bar{n}}, W_1, W_2, \ldots, W_{\bar{n}+1}\}$. Пусть $b = (4\bar{n}+1)\delta$. Зададим параметры требований следующим образом:

$$p_{V_i} = a_i, \qquad i = 1, 2, \ldots, 2\bar{n},$$

$$p_{W_i} = b, \qquad i = 1, 2, \ldots, \bar{n} + 1,$$

$$d_{V_i} = \begin{cases} (j-1)b + \delta + (a_2 + a_4 + \ldots + a_{2i}) & \text{если } i = 2j - 1, \\ d_{V_{2j-1}} + 2(\bar{n} - j + 1)(a_{2j-1} - a_{2j}) & \text{если } i = 2j, \end{cases}$$

$$d_{W_i} = \begin{cases} ib + (a_2 + a_4 + \ldots + a_{2i}) & \text{если } 1 \le i \le \bar{n}, \\ d_{W_{\bar{n}}} + \delta + b & \text{если } i = 2\bar{n} + 1. \end{cases}$$

Пусть $\{V_{i,1}, V_{i,2}\} = \{V_{2i-1}, V_{2i}\}$, $i = 1, \ldots, \bar{n}$. Определим *каноническое расписание* как расписание вида

$$\pi = (V_{1,1}, W_1, V_{2,1}, W_2, \ldots, W_{\bar{n}-1}, V_{\bar{n},1}, W_{\bar{n}}, W_{\bar{n}+1}, V_{\bar{n},2}, V_{\bar{n}-1,2}, \ldots, V_{1,2}).$$

**Утверждение 1** [1] *Для примеров канонической задачи проблемы* $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$, *к которой сводится* **ПЧНР**, *существует оптимальное расписание, которое является каноническим.*

## 3 Свойства задачи.

Без ограничения общности предположим, что требования множества $N$ пронумерованы в порядке неубывания директивных сроков

$$d_1 \le d_2 \le \ldots \le d_n,$$

если $d_k = d_{k+1}$ то $p_k \le p_{k+1}$.

Через $j^*(N')$ обозначим требование с наибольшей продолжительностью обслуживания среди требований множества $N' \subseteq N$, если таких требований несколько, то выбирается требование с наибольшим директивным сроком, то есть $j^*(N') = \arg\max_{j\in N'}\{d_j : p_j = \max_{i\in N'} p_i\}$. Для сокращения записи вместо $j^*(N')$ будем записывать $j^*$.

Далее будет показано, что алгоритмы поиска оптимального расписания, в которых используются известные методы сокращения перебора (Правила исключения 1-4, использование $E_j$ и $L_j$, построение модифицированного примера) [3, 6], в случае *канонических примеров* имеют экспоненциальную трудоёмкость.

## 3.1 Правила исключения 1-3 (Elimination Rules).

Рассмотрим пример (подпример) обслуживания требований множества $N' \subseteq N$, $N' = \{1, 2, \ldots, n'\}$, с момента времени $t' \geq t_0$. Множество $L(N', t')$ есть множество всех индексов $k \in \{1, \ldots, n'\}$, $k \geq j^*(N')$, таких что:

**(a)** $t' + \sum_{j=1}^{k} p_j < d_{k+1}$ (**Elimination Rule 1**) и

**(b)** $d_j + p_j \leq t' + \sum_{j=1}^{k} p_j$, для всех $j = \overline{j^*(N') + 1, k}$ (**Elimination Rules 2,3**).

где $d_{n'+1} = +\infty$.

Множество $L(N, t_0)$ называется множеством *подходящих позиций* для требования $j^*$.

Для любого примера (подпримера) $\langle \{p_j, d_j\}_{j\in N}, t_0\rangle$ множество $L(N, t_0)$ не пусто.

**Утверждение 2** [2, 4, 6] *Для любого примера $\langle \{p_j, d_j\}_{j\in N}, t_0\rangle$ существует оптимальное расписание $\pi^*$ обслуживания требований множества $N$, при котором $(j \to j^*)_{\pi^*}$ для всех требований $j \in \{1, 2, \ldots, k\} \setminus \{j^*\}$ и $(j^* \to j)_{\pi^*}$ для всех требований $j \in \{k + 1, \ldots, n\}$ для некоторого $k \in L(N, t_0)$.*

На рис.1 представлено дерево поиска оптимального расписания для *канонических примеров* с использованием перебора подходящих позиций.

Обозначим $\delta_i = b_{2i-1} - b_{2i}$.
**Определение** *Канонические примеры, для которых выполняется*

$$\delta - \sum_{j:=1}^{i-1} \delta_j \geq \delta_i,\ 2 \leq i \leq (\bar{n} - 1),$$

*будем называть примерами случая $SD$.*

Для случая $SD$ выполняется:

$$\delta_i > \frac{\sum_{j:=1}^{i-1} \delta_j - \delta}{2(\bar{n} - i + 1)},\quad 2 \leq i \leq (\bar{n} - 1),$$

Например, случаю $SD$ удовлетворяют примеры, для которых

$$\delta_i > 2\sum_{j:=1}^{i-1} \delta_j,\quad 2 \leq i \leq \bar{n}.$$

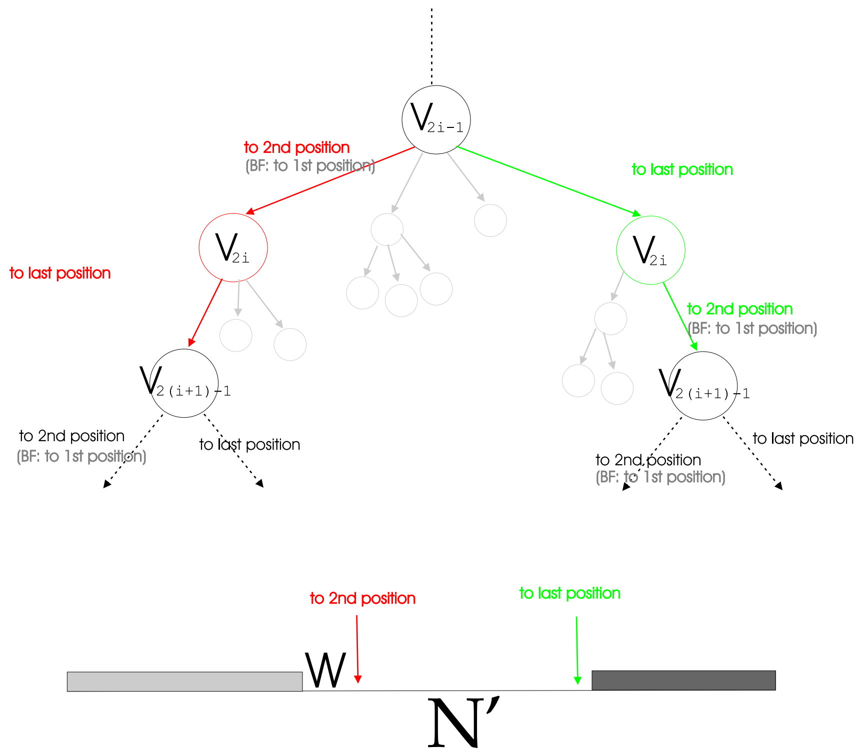


Рис. 1: Дерево поиска

Требования пронумерованы в порядке EDD: $(V_1, V_2, W_1, \ldots, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1})$.

**Определение.** *Остов дерева поиска, получаемый "двоичным ветвлением" (рис.1) будем называть "основным деревом".*

**Утверждение 3** *Дерево поиска содержит "двоичное ветвление" (рис.1). Ветвление происходит при выборе места для очередного требования $V_{2i-1}$. Для требования $V_{2i}$ допустимой становится "противоположная" позиция.*

**Доказательство.** Доказательство проведём методом математической индукции.

1. Покажем, что "двоичное ветвление" имеет место при выборе позиций для требований $V_1$ и $V_2$.

   Имеем множество требований $N = \{V_1, V_2, W_1, \ldots, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$. $j^* = V_1$. $t' = 0$.

   a. Покажем, что позиция 1 для требования $V_1$ подходящая.

      Очевидно, что правило **b** выполняется.

      Покажем, что выполняется правило **a**:

      $$0 + a_1 < d_{V_2} = \delta + a_2 + 2(\bar{n} - 1 + 1)(a_1 - a_2)$$

      То есть позиция 1 для требования $V_1$ подходящая.

Покажем, что при постановке $V_1$ в позицию 1 для требования $V_2$ становится подходящей позиция $3\bar{n}$ в списке $N' = \{V_2, W_1, \ldots, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

Покажем выполнение правила **b**. Легко убедиться, что

$$t' + \sum_{j=1}^{3\bar{n}} p_j = \sum_{j=1}^{2\bar{n}} a_j + (\bar{n}+1)b > (\bar{n}+1)b + \delta + a_2 + a_4 + \ldots + a_{2\bar{n}} + a_2.$$

$$(\bar{n}+1)b+\delta+a_2+a_4+\ldots+a_{2\bar{n}}+a_2 = d_{W_{\bar{n}+1}}+a_2 = \max_{j\in N'}\{d_j\}+\max_{j\in N'}\{p_j\} > d_j+p_j, \forall j \in N'$$

Правило **a**, очевидно, выполняется, т. к. позиция $3\bar{n}$ в списке $N'$ последняя.

b. Покажем, что позиция $3\bar{n}+1$ для требования $V_1$ подходящая.

Правило **a**, очевидно, выполняется, т. к. позиция $3\bar{n}+1$ в списке $N$ последняя.

Покажем выполнение правила **b**. Легко убедиться, что

$$0 + \sum_{j=1}^{3\bar{n}+1} p_j = \sum_{j=1}^{2\bar{n}} a_j + (\bar{n}+1)b > (\bar{n}+1)b + \delta + a_2 + a_4 + \ldots + a_{2\bar{n}} + a_1.$$

$$(\bar{n}+1)b+\delta+a_2+a_4+\ldots+a_{2\bar{n}}+a_1 = d_{W_{\bar{n}+1}}+a_1 = \max_{j\in N}\{d_j\}+\max_{j\in N}\{p_j\} > d_j+p_j, \forall j \in N$$

Покажем, что при постановке $V_1$ в позицию $3\bar{n}+1$ для требования $V_2$ становится подходящей позиция 1 в списке $N' = \{V_2, W_1, \ldots, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

Очевидно, что правило **b** выполняется.

Покажем, что выполняется правило **a**:

$$0 + a_2 < d_{W_1} = b + a_2$$

2. Предположим, что "двоичное ветвление" имело место в группах $1, 2, \ldots, i-1$. В основном дереве первое требование из каждой пары занимает "вторую" (после соответствующего требования W) или "последнюю" позицию, а второе требование из пары - "противоположную" позицию.

3. В предположении пункта 2:

$$a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b \le t' \le a_1 + a_3 + a_{2i-3} + (i-2)b.$$

Обозначим $\bar{\delta} = t' - (a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b)$, для которого выполняется $0 \le \bar{\delta} \le \sum_{j=1}^{i-1} \delta_j < 2\delta$. Тогда

$$t' = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + \bar{\delta}.$$

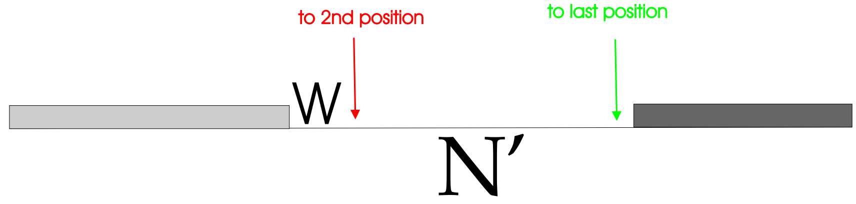


Рис. 2: Текущие "вторая" и "последняя" позиции.

Имеется множество требований $N' = \{W_{i-1}, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$. $j^* = V_{2i-1}$.

Учитывая, что $d_{W_{i-1}} < d_j$, $p_{W_{i-1}} \leq p_j$, $\forall j \in N' - W_{i-1}$, то при любом оптимальном расписании $(W_{i-1} \to j)$, $\forall j \in N' - W_{i-1}$.

Покажем, что "вторая" и "последняя" позиции для требования $V_{2i-1}$ подходящие.

a. Покажем, что текущая "вторая" позиция для требования $V_{2i-1}$ подходящая.

   Правило **b** выполняется, т. к. $t' + b + a_{2i-1} = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + \bar{\delta} + b + a_{2i-1} > d_{W_{i-1}} + p_{W_{i-1}} = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + b$.

   Покажем, что выполняется правило **a**:

   Для случая $SD$ выполняется: $t' + b + a_{2i-1} = a_2 + a_4 + \ldots + a_{2i-2} + (i-1)b + \bar{\delta} + a_{2i-1} < (i-1)b + \delta + (a_2 + a_4 + \ldots + a_{2i-2} + a_{2i}) + 2(\bar{n} - i + 1)(a_{2i-1} - a_{2i}) = d_{V_{2i}}$.

   Покажем, что при постановке $V_{2i-1}$ во "вторую" позицию для требования $V_{2i}$ становится подходящей "последняя" позиция в списке $N'' = \{V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$. Покажем выполнение правила **b**:

   $t' + b + a_{2i-1} + \sum_{j \in N''} p_j = a_2 + a_4 + \ldots + a_{2i-2} + (i-1)b + \bar{\delta} + (n - i + 2)b + \sum_{j:=2i-1}^{2n} a_j > a_2 + a_4 + \ldots + a_{2i-2} + a_{2i} + a_{2i+2} + \ldots + a_{2n} + (\bar{n}+1)b + a_{2i}$.

   $a_2 + a_4 + \ldots + a_{2i-2} + a_{2i} + a_{2i+2} + \ldots + a_{2n} + (\bar{n}+1)b + a_{2i} = d_{W_{\bar{n}+1}} + a_{2i} = \max_{j \in N''} d_j + \max_{j \in N''} p_j > d_j + p_j, \forall j \in N''$.

   Правило **a** для "последней" позиции, очевидно, выполняется, т. к. $d_{|N''|+1} = +\infty > t' + b + a_{2i-1} + \sum_{j \in N''} p_j$.

b. Покажем, что текущая "последняя" позиция для требования $V_{2i-1}$ подходящая.

   Правило **a**, очевидно, выполняется, т. к. $d_{|N'|+1} = +\infty > t' + \sum_{j \in N'} p_j$.

   Покажем выполнение правила **b**:

   $t' + b + \sum_{j \in N'} p_j = a_2 + a_4 + \ldots + a_{2i-2} + (i-1)b + \bar{\delta} + (n - i + 2)b + \sum_{j:=2i}^{2n} a_j > a_2 + a_4 + \ldots + a_{2i-2} + a_{2i} + a_{2i+2} + \ldots + a_{2n} + (\bar{n}+1)b + a_{2i-1}$.

   $a_2 + a_4 + \ldots + a_{2i-2} + a_{2i} + a_{2i+2} + \ldots + a_{2n} + (\bar{n}+1)b + a_{2i-1} = d_{W_{\bar{n}+1}} + a_{2i-1} = \max_{j \in N'} d_j + \max_{j \in N'} p_j > d_j + p_j, \forall j \in N'$.

   Покажем, что при постановке $V_{2i-1}$ в "последнюю" позицию для требования $V_{2i}$ становится подходящей "вторая" позиция в списке $N'' = \{W_{i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

Правило **b** выполняется, т. к. $t' + b + a_{2i} = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + \bar{\delta} + b + a_{2i} > d_{W_{i-1}} + p_{W_{i-1}} = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + b$.

Покажем, что выполняется правило **a**:

$t' + b + a_{2i} = a_2 + a_4 + \ldots + a_{2i-2} + (i-1)b + \bar{\delta} + a_{2i} < ib + (a_2 + a_4 + \ldots + a_{2i-2} + a_{2i}) = d_{W_i}$, т. к. $b \gg 2\delta > \bar{\delta}$.

■

## 3.2 Правило исключения 4. Elimination Rule 4.

Обозначим $T(j^*, k)$ - суммарное запаздывание частичного модифицированного EDD расписания, когда работа $j^*$ обслуживается $k$-м по порядку.

**Утверждение 4 Elimination Rule 4**. *[7, 3] Если $T(j^*, k) > T(j^*, k+1)$ или $T(j^*, k) \geq T(j^*, i)$ для некоторого $j^* \leq i < k$, то позиция $k$ исключается из списка "подходящих" позиций $L(N', t')$, если $L(N', t') \neq \{k\}$.*

**Утверждение 5** *Elimination Rule 4 не сокращает "двоичное ветвление" при выборе позиции для очередного требования $V_{2i-1}$ в случае* SD.

**Доказательство.** Доказательство проводится аналогично доказательству **Утверждения 2**.

Пусть "двоичное ветвление" имело место начиная с первой группы до группы $i-1$.

$$t' = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + \bar{\delta}, \quad 0 \leq \bar{\delta} \leq \sum_{j=1}^{i-1} \delta_j < 2\delta$$

1. Покажем, что Elimination Rule 4 не исключает из списка подходящих позиций для требования $V_{2i-1}$ "вторую" позицию в списке $N' = \{W_{i-1}, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

   Пусть $\pi_1 = (W_{i-1}, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1})$, $\pi_2 = (W_{i-1}, V_{2i}, V_{2i-1}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1})$.

   Покажем, что $T(\pi_2) - T(\pi_1) > 0$.

   $T(\pi_2) - T(\pi_1) = \max\{t' + b + a_{2i} - d_{a_{2i}},\ 0\} + \max\{t' + b + a_{2i} + a_{2i-1} - d_{V_{2i-1}},\ 0\} - \max\{t' + b + a_{2i-1} - d_{V_{2i-1}},\ 0\} - \max\{t' + b + a_{2i-1} + a_{2i} - d_{a_{2i}},\ 0\}$.

   Очевидно, что $\max\{t' + b + a_{2i} + a_{2i-1} - d_{V_{2i-1}},\ 0\} - \max\{t' + b + a_{2i-1} + a_{2i} - d_{V_{2i}},\ 0\} = 2(n-i+1)(a_{2i-1} - a_{2i})$ (из условия).

   a. Пусть $t' + b + a_{2i} - d_{V_{2i}} > 0$, то $t' + b + a_{2i-1} - d_{V_{2i-1}} > 0$ (из условия).

      Тогда $T(\pi_2) - T(\pi_1) = 2(n-i+1)(a_{2i-1} - a_{2i}) + (a_{2i} - a_{2i-1} - 2(n-i+1)(a_{2i-1} - a_{2i})) < 0$. То есть Elimination Rule 4 сокращает список "подходящих" позиций, но:

      $t' + b + a_{2i} - d_{V_{2i}} > 0 \Rightarrow a_2 + a_4 + \ldots + a_{2i-2} + a_{2i} + (i-1)b + \bar{\delta} > (i-1)b + (a_2 + a_4 + \ldots + a_{2i-2} + a_{2i}) + \delta + 2(n-i+1)(a_{2i-1} - a_{2i}) \Rightarrow \bar{\delta} > \delta + 2(n-i+1)(a_{2i-1} - a_{2i})$. Для случая $SD$ это не выполняется, т. к. $\bar{\delta} \leq \sum_{j=1}^{i-1} \delta_j < \delta$.

b. $t' + b + a_{2i} - d_{V_{2i}} \leq 0$ и $t' + b + a_{2i-1} - d_{V_{2i-1}} > 0$.
$t'+b+a_{2i}-d_{V_{2i}} \leq 0 \Rightarrow a_2+a_4+\ldots+a_{2i-2}+a_{2i}+(i-1)b+\bar{\delta} \leq (i-1)b+(a_2+a_4+\ldots+a_{2i-2}+a_{2i})+\delta+2(n-i+1)(a_{2i-1}-a_{2i}) \Rightarrow \bar{\delta} \leq \delta+2(n-i+1)(a_{2i-1}-a_{2i})$.

$T(\pi_2) - T(\pi_1) = 2(n-i+1)(a_{2i-1} - a_{2i}) - (\bar{\delta} - \delta + (a_{2i-1} - a_{2i})) \geq 2(n-i+1)(a_{2i-1}-a_{2i})-(\delta+2(n-i+1)(a_{2i-1}-a_{2i})-\delta+(a_{2i-1}-a_{2i})) = -(a_{2i-1}-a_{2i}) < 0$.
То есть Elimination Rule 4 сокращает список "подходящих" позиций, но:

$t' + b + a_{2i-1} - d_{V_{2i-1}} > 0 \Rightarrow a_2 + a_4 + \ldots + a_{2i-2} + a_{2i-1} + (i-1)b + \bar{\delta} > (i-1)b + (a_2 + a_4 + \ldots + a_{2i-2} + a_{2i}) + \delta \Rightarrow \bar{\delta} > \delta + (a_{2i} - a_{2i-1})$. Для случая $SD$ это не выполняется, т. к. $\bar{\delta} \leq \sum_{j=1}^{i-1} \delta_j < \delta$.

c. Пусть $t' + a_{2i} - d_{V_{2i}} < 0$ и $t' + a_{2i-1} - d_{V_{2i-1}} < 0$. $T(\pi_2) - T(\pi_1) = 2(n-i+1)(a_{2i-1} - a_{2i}) > 0$

2. Покажем, что Elimination Rule 4 не исключает из списка подходящих позиций для требования $V_{2i-1}$ "последнюю" позицию $N' = \{W_{i-1}, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

   Рассмотрим частичное расписание $\pi = (W_{i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}, V_{2i-1})$, исполнение которого начинается с времени $t'$. Очевидно, что $d_{V_{2i}} \leq t' + p_{W_{i-1}} + p_{V_{2i}}$, $d_{W_i} \leq t' + p_{W_{i-1}} + p_{V_{2i}} + p_{W_i}$, $d_{W_{i-1}} \leq t' + p_{W_{i-1}}$. Очевидно, что все остальные требования из $\pi$ запаздывают (см. условие *канонического примера*).

   Обозначим расписание $\pi = (W_{i-1}, \pi_1, \pi_2, V_{2i-1})$. Рассмотрим расписание $\pi' = (W_{i-1}, \pi_1, V_{2i-1}, \pi_2, )$. Очевидно, что в расписаниях $\pi$ и $\pi'$ требования из $\pi_2$ и требование $V_{2i-1}$ запаздывают.

   В расписании $\pi'$ суммарное запаздывание требований из $\pi_2$ увеличилось на $|\pi_2|a_{2i-1}$, а запаздывание требования $V_{2i-1}$ сократилось на $p(\pi_2)$.

   $T(\pi') - T(\pi) = |\pi_2|a_{2i-1} - p(\pi_2) > 0$, т. к. $a_{2i-1} > p_j, j \in \pi_2$.

   Таким образом, при $T(\pi) < T(j^*, i), \quad j^* \leq i \leq k$.

■

**Утверждение 6** *Правило исключения 4 исключает из списка "подходящих" позиций для очередного требования $V_{2i-1}$ все позиции, кроме текущей "второй" и "последней".*

**Доказательство:** Пусть

$$t' = a_2 + a_4 + \ldots + a_{2i-2} + (i-2)b + \bar{\delta}, \quad 0 \leq \bar{\delta} \leq \sum_{j=1}^{i-1} \delta_j < 2\delta.$$

Имеем множество требований $N' = \{W_{i-1}, V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}}, W_{\bar{n}+1}\}$.

Покажем, что $T(V_{2i-1}, k) > T(V_{2i-1}, k+1)$, где $k$ - не "вторая" и не "последняя" позиция.

Рассмотрим 2 расписания, построенных с момента времени $t'$: $\pi_1 = (W_{i-1}, V_{2i}, W_i, \ldots, V_{2i-1}, X, \ldots, W_{n+1})$ и $\pi_2 = (W_{i-1}, V_{2i}, W_i, \ldots, X, V_{2i-1}, \ldots, W_{n+1})$.

Не трудно показать, что все требования из $\pi_1$ и $\pi_2$ запаздывают (кроме, быть может $W_{i-1}, V_{2i}$ и $W_i$).

Тогда $T(\pi_1) > T(\pi_2)$, т. к. $p(V_{2i-1}) > p(X)$. ■

Аналогичные рассуждения можно провести для требования $V_{2i}$.
Следовательно при помощи алгоритма, использующего Правило исключения 4, строятся только канонические расписания.

## 3.3 Условия Эммонса.

Пусть $B_j$ - список работ, обслуживающихся перед требованием $j$, а $A_j$ - список работ обслуживающихся после требования $j$ при любом оптимальном расписании. Множества $B_j$ и $A_j$ могут быть пустыми одновременно.

Определим $E_j = p(B_j)+p_j$, $L_j = p(N-A_j)$ как наименьшее и наибольшее значение $C_j$ в этих расписаниях (где $t = 0$).

**Утверждение 7 Условия Эммонса** [8] *Существует оптимальное расписание, при котором*

1. *$i$ предшествует $j$ $(i \to j)$ если $d_i \leq \max(E_j, d_j),\ p_j \geq p_i$*
2. *$j$ предшествует $i$ $(j \to i)$ если $d_i + p_i \geq L_j$ и $d_i > \max(E_j, d_j),\ p_j \geq p_i$*

**Утверждение 8** *При использовании значений $E_j,\ L_j$ "двоичное ветвление"в дереве поиска для примеров случая* SD *не сокращается.*

**Доказательство:** Для списка требований $N$, при использовании правила Эммонса выполняется только $(W_{i-1} \to j),\ \forall j \in \{V_{2i-1}, V_{2i}, W_i, \ldots, W_{\bar{n}+1}\}$.

Тогда $E_{V_{2i-1}} = (i-1)b + a_{2i-1},\ E_{V_{2i}} = (i-1)b + a_{2i},\ E_{W_i} = ib = d_{W_i}$.

■

## 3.4 Построение модифицированного примера.

**Утверждение 9** [2] *Пусть $C_i$ время завершения обслуживания требования $j^*$ при некотором оптимальном расписании. Пусть*

$$\min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}.$$

*Любое оптимальное расписание $\pi'$ для модифицированного примера с директивными сроками $d'_1, d'_2, \ldots, d'_n$ является оптимальным для исходного примера с директивными сроками $d_1, d_2, \ldots, d_n$.*

Предлагается искать оптимальное решение модифицированного примера, в котором $p'_j = p_j,\ d'_j = \max\{E_j, d_j\}$ [3].

**Утверждение 10** *Для модифицированного примера ветвление в основном дереве сохранится.*

**Доказательство:** Из **Утверждения 1** и вида *канонических расписаний*:

$E_{V_{2i-1}} = (i-1)b + a_{2i-1} < d_{V_{2i-1}}$.

$E_{V_{2i}} = (i-1)b + a_{2i} < d_{V_{2i}}$.

$E_{W_i} = ib < d_{W_i}$.

То есть $d'_j = d_j, \forall j$. Модифицированный пример не отличается от исходного, т. к. $p'_j = p_j,\ d'_j = \max\{E_j, d_j\} = d_j,\ \forall j \in N$.

■

**Утверждение 11** *Величины $b_i$ не влияют на ветвления основного дерева! Влияют только $\delta_i$.*

J. Du and J. Y.-T. Leung показали, что если ответ соответствующего примера **ЧНР** "ДА", то количество оптимальных расписаний чётно [1].

## 3.5 Трудоёмкость известных алгоритмов для канонических примеров.

**Утверждение 12** *Для случая* SD *канонических примеров алгоритмы, использующие только следующие способы сокращения перебора: "Правила исключения 1-4, использование $E_j$ и $L_j$, построение модифицированного примера", имеют трудоёмкость не меньше $O(n2^{(n-1)/3-1})$.*

**Доказательство:** В **Утверждении 3** показано, для очередного требования $V_{2i-1},\ i = 1, 2, \dots, \bar{n}-1$ "подходящими" являются две позиции. Правило исключения 4 не сокращает этот список из двух позиций (**Утверждение 5**). В этом случае рассматривается две подзадачи (когда требование $V_{2i-1}$ занимает текущую "вторую" или текущую "последнюю" позицию).

Процедура поиска подходящих позиций для очередного требования с максимальной продолжительностью в каждой подзадаче имеет трудоёмкость $O(n)$.

Двоичное ветвление (рис.1) выполняется для каждого требования $V_{2i-1},\ i = 1, 2, \dots, \bar{n}-1$, количество которых $\bar{n} - 1 = (n-1)/3 - 1$. ■

Заметим, что если пример не удовлетворяет случаю $SD$, то основное дерево не будет полным. Ветвления будут продолжаться пока $\bar{\delta} < \delta + 2(n-i+1)(a_{2i-1} - a_{2i})$. Если $a_{2k-1} - a_{2k} \approx a_{2l-1} - a_{2l},\ \forall k, l = 1, \dots, n$, то сохраниться по крайней мере "половина дерева" (ветвление имеет место для $i := 1, \dots, n/2$). То есть экспоненциальная трудоёмкость алгоритма сохранится.

Можно сделать вывод: если пронумеровать корзинки в порядке $\delta_1 \geq \dots \geq \delta_n$, то трудоёмкость алгоритма, использующего Правила исключения 1-4, будет наименьшей.

# 4 Алгоритмы решения канонических примеров

Если $\delta \notin Z$, рассмотрим исходный пример ПЧНР, где все $b_i$ умножены на 2. Несложно показать эквивалентность исходного и нового примера.

Введем следующие обозначения: $d_j(t) = d_j - d_{W_{\bar{n}+1}} + t$, $j \in N$. Пусть $\pi_l(t)$ и $F_l(t)$ оптимальное расписание и соответствующее ему значение суммарного запаздывания для параметрического примера с множеством требований $N_l = \{V_{2l-1}, V_{2l}, W_l, \ldots, V_{2\bar{n}-1}, V_{2\bar{n}}, W_{\bar{n}}, W_{\bar{n}+1}\}$ и директивными сроками окончания обслуживания $d_j(t)$, $j = V_{2l-1}, V_{2l}, W_l, \ldots, V_{2\bar{n}-1}, V_{2\bar{n}}, W_{\bar{n}}, W_{\bar{n}+1}$, $l = \bar{n}+1, \ldots, 1$.

**Алгоритм B-1 canonical**

**0.** $\pi_{\bar{n}+1}(t) := (W_{\bar{n}+1})$, $F_{\bar{n}+1}(t) := \max\{0, b-t\}$ для
$t \in T_{\bar{n}+1} := [d_{W_{\bar{n}+1}} - \sum_{i=1}^{\bar{n}} a_{2i} - \bar{n}b - 2\delta, d_{W_{\bar{n}+1}} - \sum_{i=1}^{\bar{n}} a_{2i} - \bar{n}b]$

**1.** **FOR** $l = \bar{n}, \bar{n}-1, \ldots, 1$, для
$t \in T_l := [d_{W_{\bar{n}+1}} - \sum_{i=1}^{l-1} a_{2i} - (l-1)b - (2\delta - \sum_{i=l-1}^{\bar{n}} \delta_i), d_{W_{\bar{n}+1}} - \sum_{i=1}^{l-1} a_{2i} - (l-1)b]$:

$\pi^1 := (V_{2l-1}, W_l, \pi_{l+1}(t - a_{2l-1} - b), V_{2l})$, $\pi^2 := (V_{2l}, W_l, \pi_{l+1}(t - a_{2l} - b), V_{2l-1})$;

$F(\pi^1) := \max\{0, a_{2l-1} - d_{V_{2l-1}}(t)\} + \max\{0, a_{2l-1} + b - d_{W_l}(t)\} + F_{l+1}(t - a_{2l-1} - b) + \max\{0, \sum_{j=l}^{\bar{n}}(a_{2j-1} + a_{2j} + b) + b - d_{V_{2l}}(t)\}$;

$F(\pi^2) := \max\{0, a_{2l} - d_{V_{2l}}(t)\} + \max\{0, a_{2l} + b - d_{W_l}(t)\} + F_{l+1}(t - a_{2l} - b) + \max\{0, \sum_{j=l}^{\bar{n}}(a_{2j-1} + a_{2j} + b) + b - d_{V_{2l-1}}(t)\}$;

$F_l(t) := \min\{F(\pi^1), F(\pi^2)\}$; $\pi_l(t) := \arg\min\{F(\pi^1), F(\pi^2)\}$.

**2.** Алгоритм возвращает расписание $\pi_1(d_{W_{\bar{n}+1}})$ и соответствующее значение суммарного запаздывания $F_1(d_{W_{\bar{n}+1}})$.

Оптимальное расписание для исходного примера будет равно $\pi_1(d_{W_{\bar{n}+1}})$. Заметим, что шаги основного цикла алгоритма выполняются только для каждого целого $t$ из интервала, длина которого не превышает $2\delta$.

**Утверждение 13 Алгоритм B-1 canonical** *строит оптимальное каноническое расписание за время $O(n\delta)$ для всех канонических примеров проблемы* $\mathbf{1} \mid\mid \sum \boldsymbol{T_j}$, *к которым сводятся примеры ПЧНР.*

Предложен другой точный **Алгоритм B-1 модифицированный**, в котором перебираются лишь те точки $t$, в которых происходит "изменение расписания". Трудоёмкость этого алгоритма полиномиально зависит от количества таких точек.

Были проведены эксперименты, с целью выяснить количество "точек изменения расписания". Описание экспериментов и их результаты приводятся в п.6.

# 5 Частный случай проблемы $1||\sum T_j$

Рассмотрим частный случай задачи:

$$\begin{cases} p_1 \geq p_2 \geq \ldots \geq p_n, \\ d_1 \leq d_2 \leq \ldots \leq d_n, \\ d_n - d_1 \leq p_n. \end{cases} \tag{1}$$

Этот случай относится к т. н. *hard* примерам из работы [5]. Проведенные ранее эксперименты показали, что для этого частного случая дерево поиска наибольшее [6].

Рассмотрим подслучай *BF*:

$$\begin{cases} p_1 \geq p_2 \geq \ldots \geq p_n, \\ d_1 \leq d_2 \leq \ldots \leq d_n, \\ d_n - d_1 \leq p_n, \\ n = 2k, \\ \sum_{i:=1}^{n/2} p_i < d_j < \sum_{i:=n/2}^{n} p_i, \ j = 1, 2, \ldots, n, \\ p_1 - p_n \ll p_n, \\ \sum_{i:=n/2+j+1}^{n} (p_{V_{2j-1}} - p_{V_i}) > d_{V_{n/2+j}} - d_{V_{2j}}, j = 1, ..., (n/2-1), \\ \sum_{i:=n/2+j+1}^{n} (p_{V_{2j}} - p_{V_i}) > d_{V_{n/2+j}} - d_{V_{2j}}, j = 1, ..., (n/2-1), \end{cases} \tag{2}$$

где $(1, 2, \ldots, n) = (V_1, V_2, \ldots, V_{2j-1}, V_{2j}, \ldots, V_n)$.

Не трудно убедиться, что в любом расписании соответствующем примеру случая 2 запаздывать будет ровно $k$ требований.

## 5.1 Правила исключения 1-4.

**Утверждение 14** *Для примеров случая (2) дерево поиска содержит "двоичные ветвления" (рис.1). Ветвление происходит при выборе места для очередного требования $V_{2i-1}$. Для требования $V_{2i}$ допустимой становится "противоположная" позиция ($i = 1, \ldots, (k-1)$). Правило исключения 4 не сокращает "двоичные ветвления".*

**Доказательство.** Доказательство проводится аналогично доказательству **Утверждения 3** и **Утверждения 5**.

Предположим, что двоичное ветвление (когда первое требование из пары занимает "первую" или "последнюю" позицию, а второе требование из пары - "противоположную" позицию) имело место в группах $1, 2, \ldots, i-1$ $(i < k)$.

Рассматривается множество требований $N' = \{V_{2i-1}, V_{2i}, \ldots, V_n\}$.

Множество требований $N'$ начинает обслуживаться с момента времени $t' = p_2 + p_4 + \ldots + p_{2i-2} + \bar{\delta}$, где $0 \leq \bar{\delta} \leq \sum_{j=1}^{i-1}(p_{2j-1} - p_{2j}) < p_n$.

Требование с максимальной продолжительностью $j^* = V_{2i-1}$. Покажем, что "первая" и "последняя" позиции для требования $V_{2i-1}$ подходящие.

a. Покажем, что текущая "первая" позиция для требования $V_{2i-1}$ подходящая.

   Очевидно, что правило **b** выполняется, т. к. $d_j + p_j \leq t' + p_{2i-1}, \ \forall j = \overline{1+1, 1}$.

   Покажем, что выполняется правило **a**:

$t' + p_{2i-1} = p_2 + p_4 + \ldots + p_{2i-2} + \bar{\delta} + p_{2i-1} < d_{V_{2i}}$. То есть требование $V_{2i-1}$ при постановке на "первую" позицию не запаздывает. Можно показать, что не запаздывает и следующее требование $V_{2i}$.

Покажем, что Правило исключения 4 не исключает "первую" позицию. Рассмотрев 2 расписания $\pi' = (\pi_1, V_{2i-1}, V_{2i}, \pi_2)$ и $\pi'' = (\pi_1, V_{2i}, V_{2i-1}, \pi_2)$, где $p(\pi_1) = t'$ Легко убедиться, что $T(\pi') = T(\pi'')$, т. к. при обоих расписаниях требования $V_{2i-1}$ и $V_{2i}$ не запаздывают.

Проводя аналогичные рассуждения не трудно доказать, что при постановке требования $V_{2i-1}$ в "последнюю" позицию, для требования $V_{2i}$ допустимой становиться "первая" позиция.

b. Покажем, что текущая "последняя" позиция для требования $V_{2i-1}$ подходящая.

Очевидно, что правило **a** выполняется, т. к. $d_{|N'|+1} = +\infty > t' + \sum_{j \in N'} p_j$.

Покажем выполнение правила **b**.

$$t' + \sum_{j \in N'} p_j = p_2 + p_4 + \ldots + p_{2i-2} + \bar{\delta} + \sum_{j:=2i-1}^{2n} p_j > \sum_{j:=n/2}^{n} p_j + p_{2i-1}.$$

$$\sum_{i:=n/2}^{n} p_i + p_{2i-1} > \max_{j \in N'} d_j + \max_{j \in N'} p_j > d_j + p_j, \forall j \in N'.$$

Покажем, что Правило исключения 4 не сокращает перебор.

Рассмотрим два частичных модифицированных EDD расписания $\pi' = (\pi_1, \pi_2, V_{2i-1})$ и $\pi'' = (\pi_1, V_{2i-1}, \pi_2)$, составленные из требований множества $N'$ и выполняющихся с момента времени $t'$.

Рассмотрим 2 случая:

1. Пусть при расписании $\pi''$ требование $V_{2i-1}$ запаздывает. Очевидно, что при расписаниях $\pi'$ и $\pi''$ запаздывают одни и те же требования. Для случая *B-1* запаздывающие требования упорядочены в порядке **SPT**. Тогда $T(\pi') < T(\pi'')$.

2. Пусть в расписании $\pi''$ требование $V_{2i-1}$ не запаздывает. Представим расписания в виде: $\pi' = (\pi_1, \pi_{21}, \pi_{22}, V_{2i-1})$ и $\pi'' = (\pi_1, V_{2i-1}, \pi_{21}, \pi_{22})$. Требования из множества $\pi_{22}$ запаздывают при обоих расписаниях. $|\pi_{21}| > 0$. Требования из множества $\pi_{21}$ при расписании $\pi'$ не запаздывают. Учитывая, что количество запаздывающих требований при любом полном расписании равно $n/2 = k$, то не трудно вычислить $|\pi_{22}|$.

   К шагу $i$ уже выбрано $i-1$ запаздывающих требований, расположенных "в конце" полного расписания. Тогда $|\pi_{22}| = n/2 - (i-1) - 1$.

   Множество $\pi_{22} = \{V_{n-|\pi_{22}|+1} \ldots, V_n\} = \{V_{n/2+i+1} \ldots, V_n\}$. Тогда при расписании $\pi''$ стало запаздывать еще одно требование $V_{n/2+i} \in \pi_{21}$.

   $T(\pi') - T(\pi'') = C_{V_{2i-1}} - d_{V_{2i-1}} - |\pi_{22}| p_{V_{2i-1}} - (C_{V_{n/2+i}} - d_{V_{n/2+i}}) = t' + \sum_{i \in \pi_1 \bigcup \pi_{21}} p_i + \sum_{i \in \pi_{22}} p_i + p_{V_{2i-1}} - d_{V_{2i-1}} - |\pi_{22}| p_{V_{2i-1}} - (t' + \sum_{i \in \pi_1 \bigcup \pi_{21}} p_i + p_{V_{2i-1}} - d_{V_{n/2+i}}) = d_{V_{n/2+i}} - d_{V_{2i-1}} - \sum_{j:=n/2+i+1}^{n} (p_{V_{2i-1}} - p_{V_j})$.

Для случая (2) выполняется $d_{V_{n/2+i}} - d_{V_{2i-1}} < \sum_{j:=n/2+i+1}^{n}(p_{V_{2i-1}} - p_{V_j})$, то $T(\pi') - T(\pi'') < 0$. Тогда $T(V_{2i-1}, i) > T(V_{2i-1}, |N'|),\ 1 \le i < |N'|$. Также выполняется $T(V_{2i-1}, |N'|) < T(V_{2i-1}, |N'| + 1) = +\infty$. Значит Правило исключения 4 не сокращает перебор.

Аналогичные рассуждения можно провести для "последней" позиции требования $V_{2i}$, когда требование $V_{2i-1}$ поставлено на "первую" позицию.

■

**Утверждение 15** *Для примеров случая (2) дерево поиска содержит только "двоичные ветвления" (рис.1).*

**Доказательство:** Предположим, что двоичное ветвление имело место в группах $1, 2, \ldots, i-1$ $(i < k)$. То есть в основном дереве первое требование из пары занимает "первую" или "последнюю" позицию, а второе требование из пары - "противоположную" позицию.

Рассматривается множество требований $N' = \{V_{2i-1}, V_{2i}, \ldots, V_n\}$.

Множество требований $N'$ начинает обслуживаться с момента времени $t' = p_2 + p_4 + \ldots + p_{2i-2} + \bar{\delta}$, где $0 \le \bar{\delta} \le \sum_{j=1}^{i-1}(p_{2j-1} - p_{2j}) < p_n$.

Требование с максимальной продолжительностью $j^* = V_{2i-1}$. Покажем, что только "первая" и "последняя" позиции для требования $V_{2i-1}$ подходящие.

1. Рассмотрим модифицированное EDD расписание $\pi = (\pi_1, V_{2i-1}, X, \pi_2)$, в котором $V_{2i-1}$ смещено с "первой" позиции в позицию $k$. Пусть требование $V_{2i-1}$ запаздывает при расписании $\pi$, тогда запаздывает и требование $X$. В расписании $\pi' = (\pi_1, X, V_{2i-1}, \pi_2)$ оба требования также запаздывают (см. случай **BF**). Тогда $T(\pi') < T(\pi)$, т. к. $p_{V_{2i-1}} > p_X$. Тогда по Правилу Исключения 4 позиция $k$ не является подходящей.

2. Рассмотрим модифицированное EDD расписание $\pi = (\pi_1, X, V_{2i-1}, \pi_2)$, в котором $V_{2i-1}$ смещено с "первой" позиции в позицию $k$. Пусть требование $V_{2i-1}$ не запаздывает при расписании $\pi$. То $C_{V_{2i-1}}(\pi) \le d_{V_{2i-1}}$. Тогда $C_{V_{2i-1}}(\pi) \le d_{V_{2i-1}} \le d_X < d_X + p_X$. Тогда по правилу **b** позиция $k$ не является подходящей.

■

## 5.2 Условия Эммонса. Построение модифицированного примера.

**Утверждение 16** [6] *Для всех $l \in N$, когда параметры требований удовлетворяют (1), существует оптимальное расписание $\pi^* = (\pi_1^*, \pi_l, \pi_2^*)$, где $\{\pi_l\} = N_l = \{l, \ldots, n\}$.*

Не трудно показать, что $E_j = p_j,\ L_j = \sum_{i:=1}^{n} p_i$. Тогда модифицированный пример в котором $p'_j = p_j,\ d'_j = \max\{E_j = p_j, d_j\} = d_j$ является оригинальным. Значения $E_j,\ L_j$ не противоречат "двоичному ветвлению".

### 5.3 Алгоритмы решения случая (1)

**Утверждение 17** *Для случая (2) алгоритмы, использующие только следующие правила исключения: "Правила исключения 1-4, использование $E_j$ и $L_j$, построение модифицированного примера", имеют трудоёмкость $O(n2^{n/2})$.*

Для случая (1) построены точный псевдополиномиальный **Алгоритм B-1** трудоёмкости $O(n\sum p_j)$ и точный **Алгоритм B-1 модифицированный** предположительно трудоёмкости $O(n^4)$.

Для случая (2) построен точный **Алгоритм BF** трудоёмкости $O(n^2)$.

# 6 Проведение экспериментов

Были проведены многочисленные эксперименты для выяснения количества *точек изменения расписания* для *канонических примеров*. Результаты экспериментов позволили оценить трудоемкость **Алгоритма B-1 модифицированный**. Результаты проведенных экспериментов (№№ 6, 8, 8.2) представлены в сводной таблице после эксперимента №8.

В ходе экспериментов последовательно перебирались примеры **ПЧНР** из некоторого класса. Каждый пример за полиномиальное время приводился к каноническому примеру задачи $1||\sum T_j$. Для канонического примера запускался **Алгоритм B1 canonical**, на каждом шаге $l$ которого подсчитывалось количество *точек изменения расписания.*

**Эксперимент №6.** Перебирались все примеры ПЧНР с количеством групп $\bar{n} = 2, 3, 4, 5$ для которых выполняется:

$$\begin{cases} 200 \geq b_1 > b_2 \geq b_3 > b_4 \geq \ldots \geq b_{2\bar{n}-1} > b_{2\bar{n}} \geq 1. \\ b_i \in Z^+ \end{cases}$$

Ограничение $200 \geq b_1$ обусловлено вычислительными мощностями. Т.к. перебираются все примеры $200 \geq b_1 > b_2 \geq b_3 > b_4 \geq \ldots \geq b_{2\bar{n}-1} > b_{2\bar{n}}$, то их число превосходит количество сочетаний $C_{200}^{2*\bar{n}}$. Для размерности $n = 5$ число примеров превосходит:

$$C_{200}^{10} = \frac{200!}{(200-10)!10!} = 22'451'004'309'013'280$$

Увеличение $b_1$ значительно увеличит количество рассматриваемых примеров, что в разумные сроки не могло быть проанализировано.

Получены следующие результаты:

| n (количество групп) | Макс. кол-во точек изм. расписания |
|---|---|
| 2 | 1 |
| 3 | 4 |
| 4 | 11 |
| 5 | 21 |

Количество *точек изменения расписания* считается на каждом шаге $l = n, n-1, \ldots, 1$ и суммируется.

Детальный анализ результатов позволил выявить следующее важное свойство:
**Свойство 1.** *Два примера ПЧНР*
$\{(a_1, a_2), (a_3, a_4), \ldots, (a_{2j-1}, a_{2j}), \ldots, (a_{2\bar{n}-1}, a_{2\bar{n}})\}$
*и* $\{(a_1, a_2), (a_3, a_4), \ldots, (a_{2j-1} + \triangle, a_{2j} + \triangle), \ldots, (a_{2\bar{n}-1}, a_{2\bar{n}})\}$, *где* $\triangle \in Z^+$, $a_{2j-1} + \triangle \leq a_{2j-2}$, $a_{2j} + \triangle \leq a_{2j+1}$ *(примеры с одинаковым распределением* $\delta_i$*) равнозначны. Соответствующие канонические примеры имеют одинаковое количество* точек изменения расписания *и одинаковое оптимальное расписание.*

Учитывая данное свойство примеры ПЧНР будем обозначать $(\delta_1, \delta_2, \ldots, \delta_{\bar{n}})$. Можно сократить количество рассматриваемых примеров (см. эксперимент №8).

**Эксперимент №8.** Перебирались все примеры ПЧНР $(\delta_1, \delta_2, \ldots, \delta_{\bar{n}})$. В полной мере эксперимент завершен для примеров с количеством групп $n = 2, 3, \ldots, 7$ для которых выполняется:

$$\begin{cases} 50 \geq \sum_{i:=1}^{\bar{n}} \delta_i. \\ \delta_i \in Z^+ \end{cases}$$

Обозначим $CS_l$ - количество *точек изменения расписания* на шаге $l = n, n-1, \ldots, 1$ **Алгоритма B1 canonical**.

Были найдены *сложные примеры* ПЧНР, для которых $CS_n = 1, CS_{n-1} = 3$, $CS_{n-2} \geq 7$, $CS_{n-3} \geq 15$, $CS_{n-4} \geq 31, \ldots$
Выявлены следующие свойства:
**Свойство 2.** Пусть $S$ - множество примеров, для которых $\delta_1 = \sum_{i=2}^{n} \delta_i$ (3). Пусть $U$ - множество сложных примеров. То примеры (3) в своем большинстве принадлежат примерам с повышенной сложностью, $S \bigcap U \neq \emptyset$.
**Свойство 3.** Среди примеров с повышенной сложностью, обладающих свойством (3), около 70% имеют сложность (1,3,7,15,31,0) (для размерности $n = 6$)
**Свойство 4.** Если из сложного примера размерности $\bar{n}$ "убрать" первую группу, то полученный пример размерности $\bar{n} - 1$ не обязательно сложный.
**Свойство 5.** Пусть пример $(\delta_1, \delta_2, \ldots, \delta_n)$ имеет сложность (1,3,7,15,31), то такую же сложность имеет пример $(\delta_1 + k * 2, \delta_2, \ldots, \delta_n)$.
**Свойство 6.** К классу сложных примеров относятся не только "разрешимые" примеры ПЧНР (т.е. решение примера "НЕТ").
**Свойство 7.** Очевидно что, если перенумеровать пары чисел таким образом, что $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$, то получим эквивалентный пример. Но при этом снижается трудоемкость **Алгоритма B-1 canonical** и часто уменьшается количество *точек изменения расписания.* Для такой нумерации групп проведен аналогичный эксперимент №8.2. Стоит отметить, что примеры $SD$ удовлетворяют такой нумерации.

Получены следующие результаты:

| n | B-1 | B-1 canonical $(CS_n, CS_{n-1}, \ldots, CS_1)$ | |
|---|---|---|---|
| | | $\delta_1, \delta_2, \ldots, \delta_n$ | $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$ |
| 2 | 1 | (1,0) | (1,0) |
| 3 | 3 | (1,3,0) | (1,1,0) |
| 4 | 8 | (1,3,8,0) | (1,3,2,0) |
| 5 | 15 | (1,3,8,18,0) | (1,3,5,5,0) |
| 6 | 23 | (1,3,8,18,32,0) | (1,3,7,7,7,0) |
| 7 | 38 | (1,3,8,18,32,63,0) | (1,3,7,13,19,19,0) |
| 8 | 44 | - | - |
| 9 | 51 | - | - |

# 7 Выводы.

Для случаев (2) и *SD* проблемы $1||\sum T_j$ алгоритмы, использующие только следующие правила исключения: "Правила исключения 1-4, использование $E_j$ и $L_j$, построение модифицированного примера", имеют экспоненциальную трудоёмкость не менее ($O(n2^{n/2})$ и $O(n2^{(n-1)/3})$). Вызывает сомнение, что такими алгоритмами можно решать примеры размерностью $n \geq 100$.

При оценке эффективности алгоритмов для задачи $1||\sum T_j$ не достаточно использовать тестовые примеры из работы [4].

Для случаев (1) и *SD* предложены точные псевдополиномиальные и полиномиальные алгоритмы.

# Список литературы


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

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

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

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

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

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

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

[8] H. Emmons (1969). *One machine sequencing to minimize certain functions of job tardiness*, Oper. Res., 17, pp. 701–715.