# ЗАДАЧА МИНИМИЗАЦИИ МАКСИМАЛЬНОГО ВЗВЕШЕННОГО ВРЕМЕННОГО СМЕЩЕНИЯ ВЫПОЛНЕНИЯ ЗАКАЗА ДЛЯ ДВУХ СТАНЦИЙ[1]

**Лазарев А.А., Мусатова Е.Г., Архипов Д.И.**
*(Институт проблем управления им. В.А. Трапезникова РАН, г. Москва)*
lazarev@ipu.rssi.ru, musatova@ipu.ru, miptrafer@gmail.com

*Аннотация:*
*Рассматривается задача составления плана движения грузовых составов между двумя станциями, соединенными двухпутной железной дорогой. Предлагается алгоритм динамического программирования, минимизирующий максимальное взвешенное временное смещение доставки заказов за полиномиальное число шагов.*
Ключевые слова: теория расписания, взвешенное временное смещение, железнодорожный транспорт.

## 1. Математическая постановка задачи

### *1.1. Постановка задачи*

Имеется две станции, соединенные двухпутной железной дорогой. Необходимо выполнить множества заказов $N^1=\{J_1^1, J_2^1 \dots J_{n_1}^1\}$, $N^2=\{J_1^2, J_2^2 \dots J_{n_2}^2\}$ на поставку грузов между станциями. Заказы множества $N^1$ необходимо доставить с первой станции на вторую, а заказы множества $N^2$ − со второй на первую. Каждый заказ состоит из одного вагона. Все вагоны однотипные. Так как железная дорога двухпутная, то расписания для множеств $N^1$ и $N^2$ не зависимы, поэтому будем обозначать множество заказов $N = \{J_1, \dots, J_n\}$. Пусть $r_j$ − время поступления заказа $J_j$ на станцию. Без потери общности предположим, что заказы пронумерованы в порядке их поступления. Каждый заказ имеет свою ценность $w_j > 0$.

Доставка вагонов с одной станции на другую осуществляется составами, каждый из которых состоит из $k$ вагонов. Пусть $p$ − время движения состава между станциями, а α − время, которое должно разделять моменты отправки двух поездов. Каждый заказ имеет директивный срок $d_j = r_j + \delta$ − момент времени, до которого заказ может быть доставлен на станцию назначения без опоздания, где $\delta$ − запас времени на доставку. Все заказы должны быть обслужены $q$ Поездами, где $q = \frac{n}{k}$. Пусть $C_j$ − время доставки заказа $J_j \in N$. Целевая функция задачи записывается как $\min \max_{j=\overline{\{1,n\}}}(w_j(C_j - d_j))$.

Расписание, удовлетворяющее данной целевой функции, будем называть оптимальным и обозначать $\pi(N)$.

Рассмотрим следующий пример задачи. Пусть $n = 6$, $k = 2$, $p = 4$, $\delta = 4$, $\alpha = 2$, $r_1 = r_2 = 0, r_3 = r_4 = 1, r_5 = r_6 = 3$, $w_1 = 3, w_2 = w_3 = w_4 = 10, w_5 = 30, w_6 = 5$. Оптимальное расписание для данного примера проиллюстрировано на рисунке 1. Здесь, начало прямоугольника − время отправки поезда, конец − время прибытия. Цифры внутри прямоугольника обозначают номера заказов, перевозимых поездом. Значение целевой функции равно 3 и достигается на заказе $J_1$ (см. рис. 1).

[1]*Статья выполнена при финансовой поддержке гранта 11-08-13121-офи-м-2011-РЖД.*

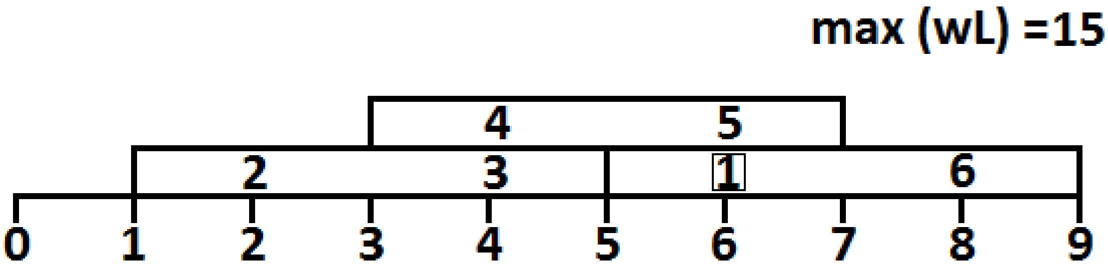


*Рис. 1. Оптимальное расписание для примера 1.*

*1.2. Постановка задачи минимизации общего времени выполнения заказов при ограничении на максимальное взвешенное временное смещение*

Пусть задано множество $N$ и положительное число $y$. Задача заключается в составлении расписания $\Theta(N,y)$ удовлетворяющего условию $\min_{j=\overline{\{1,n\}}}(C_j) \mid \max_{j=\overline{\{1,n\}}}(w_j(C_j-d_j)) < y$.

Введем дополнительные обозначения. Для данного значения $y$, задающего ограничение, для каждого заказа $j$ может быть определён момент времени $t'_j$, до которого данный заказ должен быть отправлен. Из ограничения на максимальное временное смещение имеем:

$$y > w_j(C_j - d_j) = w_j(C_j - r_j - \delta) \Rightarrow C_j < \frac{y}{w'_j} + r_j + \delta.$$

Так как после отправки заказа требуется ещё некоторое время $p$, чтобы он доехал до станции назначения, то получаем условие на момент отправки заказа $j$:

$$t'_j = C_j - p < \frac{y}{w'_j} + r_j + \delta \Rightarrow t'_j < \frac{y}{w'_j} + r_j + \delta - p.$$

Таким образом, задача может быть сформулирована следующим образом. Имеется множество $n = kq$ заказов, для каждого из которых определены момент поступления $r_j$ и момент обязательной отправки $t'_j$, после которого будет нарушено ограничение сверху на максимальное взвешенное временное смещение. Необходимо максимально быстро перевезти заказы на $q$ поездах, каждый из которых вмещает ровно $k$ заказов и тратит время на дорогу, равное $p$, так чтобы каждый заказ уехал на поезде, отправляющемся в промежутке $[r_j, t'_j)$.

## ***2.Алгоритм построения оптимального расписания***

*2.1 Свойства задачи*

Введем семейство множеств $S_1 \subset S_2 \subset \cdots \subset S_q$ – заказы, которые должны уехать первым поездом ($S_1$), первым или вторым поездом ($S_2$), первым или вторым или третьим ($S_3$) и т.д. Введем функцию $r(X) = \max_{\{J_i \in X\}}(r_i)$, где $X$ – некоторое множество заказов. Обозначим также за $X_i$ подмножество множества $X$, состоящее из первых (в порядке поступления) $i$ заказов множества $X$.

Заметим, что до начала работы алгоритма построения $\Theta(N,y)$ не пусто только множество $S_q$, которое состоит из всех $n = kq$ заказов.

Задача построения расписания $\Theta(N,y)$ может быть решена с помощью алгоритма динамического программирования. Алгоритм основан на следующих достаточно очевидных, но важных свойствах задачи.

В расписании $\Theta(N,y)$ отправка поезда с номером $m$ в момент времени $t^m$ должна удовлетворять следующим условиям:

1. если в момент отправки доступно больше, чем $k$ заказов, то необходимо отправлять те, у которых моменты обязательной отправки $t'_j$ меньше;
2. поезд с номером $m$ не может отправиться раньше момента $t^m \geq max(r_{km}, r(S_m), t^{m-1} + \alpha)$;
3. все заказы $l$, для которых выполняется $t^m + \alpha \geq t'_l$, должны уехать одним из поездов с номерами от 1 до $m$ т.е. $J_l \in S_m$;

iv. после отправки -го поезда все заказы из множества $S_m$ должны быть отправлены.

Условие (i) следует из того, что, если есть 2 заказа $a$ и $b$ такие, что $t'_a > t'_b$, то заказ $a$ может быть отправлен после момента $t^m$ не меньшим количеством поездов, чем заказ $b$.

Условие (ii) справедливо, поскольку поезд $m$ не может быть отправлен до того момента, как:

- поступит $km$ заказов;
- поступят все заказы из $S_m$;
- $(m-1)$-ый поезд освободит пути.

Условие (iii) следует из того, что поезд, отправляющийся в момент времени $t^m$, делает невозможным отправку следующего поезда до момента $t^m + \alpha$, а значит, все заказы $l$, для которых выполняется неравенство $t^m + \alpha \geq t'_l$, не смогут отправиться поездами, номер которых больше $m$.

Условие (iv) следует из определения множества $S_m$.

Обозначим за $T_m$ текущее множество заказов, перевозимых -ым поездом.

*2.2 Алгоритм построения расписания* $\Theta(N, y)$

На каждом шаге алгоритма производится попытка отправить некоторый поезд $m$. Для этого выбираем первый момент $t^m$ такой, что для него выполняется условие (ii). После этого выбираются $k$ доступных заказов по условию (i). Затем проверяется верность свойства (iii). В случае, если существует заказ $J_l$ такой, что $r_l > t^m$и $t^m + \alpha \geq t'_l$, то согласно условию (iii) мы включаем заказ $J_l$ во множества $S_m, S_{m+1}, \dots, S_q$ и возвращаемся к проверке условия (ii). Если же такого заказа не существует, то условие (iii) может быть не выполнено только в том случае, если к моменту времени $t^m$ доступно $x > k$ неотправленных заказов из множества $S_m$. Множество этих заказов назовем $X^0$. Следовательно, $x - k$ заказов должны будут отправиться первыми $m-1$ поездами. Чтобы отправить данные $x-k$ заказов первыми $m-1$ поездами будем по порядку во множествах $T_{m-1}, T_{m-2}$ ... искать такие заказы $J_j$, что $t'_j > t^m + \alpha$, до тех пор, пока их количество не достигнет значения $x-k$. Пусть последний такой заказ был найден на поезде с номером $s$. Обозначим за $T_s^{m-1} = T_s \cup \dots \cup T_{m-1}$. Пусть $X'$ − множество, состоящее из $x-k$ заказов $j \in T_s^{m-1}$, для которых верно $t'_j > t^m + \alpha$, имеющих минимальные моменты $t'$. Далее будем отправлять заказы из множества $X = (T_s^{m-1} \setminus X') \cup X^0$ поездами $s, \dots, m$. Условие отправки (ii) изменится с учетом того, что отправлять мы можем только заказы из $X$, тем самым вместо $r_{ki}$ появится $r(X_{(i-s+1)k})$. При этом противоречие условию (i) невозможно, так как любые из доступных в этот момент заказов, не принадлежащих $X$, имеют $t' > t^m + \alpha$. В случае противоречия условию (iii) происходи изменение множества $S_i$, осуществляется переход к поезду $i$ и начинается новый шаг. Если противоречий не встретилось, то отправляем по этим правилам поезда с номерами $s, \dots, m$ и переходим к следующему шагу − отправке поезда $m+1$. Алгоритм прерывает свою работу, если на каком-то шаге мощность какого-либо множества $S_i$ стала больше $ki$.

Обозначим за $Add\_S(j, m)$ процедуру, которая добавляет заказ $J_j$ во множества $S_m, S_{m+1}, \dots, S_q$ , а в случае, когда $|S_m| > km$, прерывает работу алгоритма. Положим $t^0 := -\alpha$.

**Алгоритм $\boldsymbol{\Theta(N, y)}$**

**0.** Определить моменты $t'_j, j = 1,2, \dots, n$, в зависимости от $y$. Положить $i := 1$.

**1.** Если $i = n+1$, то **Stop**, иначе, определить момент возможной отправки
$$t^i := max(r_{ki}, r(S_i), t^{i-1} + \alpha).$$

**2.** Если существует не отправленный ранее заказ $J_j \in S_i : t'_j < t^i$, то $Add\_S(j, i-1)$ (в случае $i = 1$, **Stop**), $i := i-1$, перейти на шаг 1.

**3.** Если существует заказ $J_j : t'_j \leq t^i + \alpha,\ r_j > t^m$, то $Add\_S(j, i)$, перейти на шаг 1.

**4.** Найти $x = |X^0|$ – количество не отправленных ранее заказов из множества $S_m$.

**5.** Если $x \leq k$, то построить множество $T_i$ в соответствии с (i), $i := i+1$, перейти на шаг 1.

**6.** Начиная с $(i-1)$-го поезда, найти множество заказов $X'$, которые будут исключены из рассмотрения.
**7.** Распределить множество заказов $(T_s^{i-1} \setminus X') \cup X^0$ между поездами $s, s+1, \dots, i$ в соответствии с условиями (i)-(iv). Если некоторый заказ $J_j$ должен быть добавлен в $S_m$, $m \in \{s-1, \dots, i\}$, то $Add\_S(j, m)$, $i \coloneqq m$ и перейти на шаг 1.

Данный алгоритм строит расписание $\Theta(N, y)$, так как на каждом шаге производится попытка отправить поезд в первый момент времени, который возможен при текущих условиях, что дает минимальность $C_{max}$, и при этом прерывает свою работу только в случае несоответствия условий, откуда напрямую следует отсутствие допустимого расписания $\Theta(N, y)$.

*2.3 Оценка сложности алгоритма построения $\Theta(N, y)$*

Оценим сложность работы алгоритма. На каждом шаге происходит проверка условий, пополнение множества $S_m$, изменение моментов $t^i$ (не более, чем для $q$ поездов), подсчет заказов, удовлетворяющих определенным условиям во множествах $T_s, \dots, T_m$, формирование множеств $X^0, X'$ и $X$, а также формирование поездов с $T_s$ по $T_m$. Эти операции имеют трудоемкость $O(n^2)$. На каждом шаге алгоритма в одно из множеств $S_1, S_2, \dots, S_{q-1}$ добавляется не менее 1 заказа. При этом во множестве $S_m$ может находиться не более $mk$ заказов, а значит, количество шагов не больше чем $k + 2k + \dots + (q-1)k = k\frac{(q-1)(q-1)}{2}$, т.е. будем иметь $O(q^{2k}) = O(\frac{n^2}{k})$ операций. Тем самым трудоемкость алгоритма, строящего расписание $\Theta(N, y)$ составит $O\left(\frac{n^4}{k}\right)$ операций.

*2.4 Алгоритм построения расписания $\pi(N)$*

При построении оптимального расписания $\pi(N)$ будем действовать следующим образом. Построим расписание, в котором заказы отправляются по возрастанию моментов поступления. Тем самым мы получим расписание, удовлетворяющее условию минимума $C_i$ для каждого поезда $i$.
Рассмотрим заказ $J_{j_1}$, на котором достигается максимум целевой функции $(w_{j_1} L_{j_1})$. Пусть этот заказ в данном расписании отправляется $m_1$-ым поездом. Тогда для того, чтобы улучшить целевую функцию, необходимо отправить заказ $J_{j_1}$ одним из поездов, отправившимся до поезда с номером $m_1$, т.е. необходимо включить заказ $J_{j_1}$ во множество $S_{m_1-1}$. Это значит, что если мы построим расписание $\Theta(N, w_{j_1} L_{j_1})$, то в нем заказ $J_{j_1}$ будет отправлен одним из поездов, номер которого меньше $m_1$, а целевая функция будет меньше, чем $w_{j_1} L_{j_1}$ и достигается на заказе $J_{j_2}$, который идет поездом $m_2$.

Следующим шагом построим расписание $\Theta(N, w_{j_2} L_{j_2})$, в котором заказ $J_{j_2}$ должен ехать поездом с номером меньше, чем $m_2$, а значит, заказ $J_{j_2}$ попадет во множество $S_{m_2-1}$. Будем повторять эту процедуру до тех пор, пока не наступит такой шаг $s$, что расписания $\Theta(N, w_{j_s} L_{j_s})$ не существует. Это означает, что не существует расписания со значением целевой функции меньшем, чем $w_{j_s} L_{j_s}$, и при этом существует расписание, полученное на -м шаге со значением целевой функции $w_{j_s} L_{j_s}$, следовательно, оно и будет оптимальным.

**Алгоритм $\pi(N)$**

**Шаг 0.** Построить расписание, в котором заказы отправляются по возрастанию моментов поступления.
**Шаг 1.** Положить $y \coloneqq \max_{j \in N} w_j L_j$. Найти $j : w_j L_j = y,\ m : J_j \in T_m$ и $Add\_S(j, m-1)$.
**Шаг 2.** Запустить Алгоритм $\Theta(N, y)$ и перейти на шаг 1.

*2.5 Оценка трудоёмкости алгоритма построения расписания $\pi(N)$*

Заметим, что на каждом шаге происходит перенос заказа, на котором достигнута целевая функция, в одно из множеств $S_1, S_2, \dots, S_{q-1}$, при этом обратный переход заказа невозможен.

Пусть на некотором шаге достигнуто значение целевой функции на заказе $i$, отправленном поездом с номером $m$ в расписании $\Theta(N, \beta)$, после чего данный заказ перенесен во множество

$S_{m-1}$. На следующих шагах будут получены расписания вида $\Theta(N, y)$, где $y < \beta$, а любой поезд с номером $m$ при расписании $\Theta(N, \beta)$ отправлен не позже, чем поезд с номером $m$ при расписании $\Theta(N, y)$. Это следует из того, что все сдвиги, которые мы делаем при построении расписания $\Theta(N, \beta)$ будут необходимы и в расписании $\Theta(N, y)$. Поэтому, чтобы целевая функция не стала больше либо равной той, что достигалась в расписании $\Theta(N, \beta)$ на заказе $i$, должно выполняться условие отправки заказа $i$ одним из поездов с номером меньше $m$, а значит заказ $i$ останется во множестве $S_{m-1}$.

Суммарное количество мест во множествах $S_1, S_2, \dots, S_{q-1}$ равняется $k + 2k + \dots + (q-1)k = O(kq^2)$, а значит, количество шагов равно $O(kq^2) = O\left(\frac{n^2}{k}\right)$. На каждом шаге мы получаем расписание $\Theta$, для построения которого нам необходимо $O(\frac{n^4}{k})$ операций. Следовательно, общая трудоёмкость составляет $O(\frac{n^6}{k^2})$ операций.

### *3. Пример работы алгоритма*

Продемонстрируем работу алгоритма на примере расписания, построенного в примере 1.

**Шаг 1**. Строим для данного множества заказов расписание, удовлетворяющее условию $\min(C_{max})$, находим значение целевой функции $\max_{i=\overline{\{1,n\}}}(w_i L_i)$ и принимаем его за $y = 30$ (см рис.2).

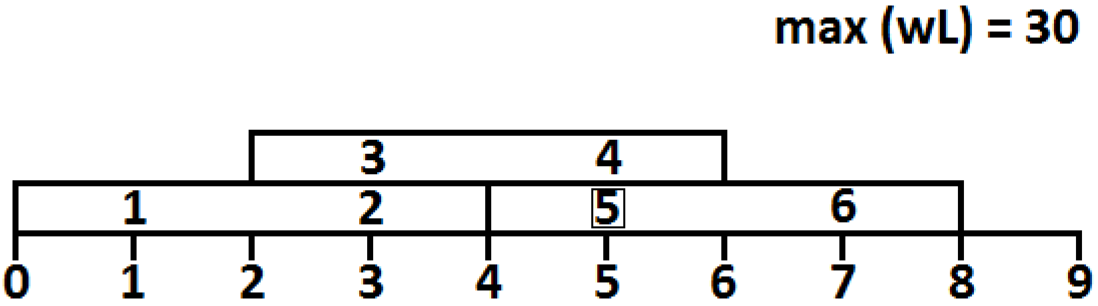


*Рис. 2. Расписание $min(C_{max})$ для множества заказов $N$.*

**Шаг 2.** Строим расписание $\Theta(N, y)$ для множества $N = \{J_1, \dots, J_6\}$ и $y = 30$. Определяем моменты $t_i' = r_i + \delta + \frac{y}{w_i} - p$, $i = \overline{1,6}$: $t_1' = 10, t_2' = 3, t_3' = t_4' = t_5' = 4, t_6' = 9$. Теперь пытаемся отправить поезда $1, 2, \dots, q$, так, чтобы это не противоречило условиям (i) – (iv). Заметим, что при отправке поезда №2 возникает противоречие свойству (iii) (см. рис. 3).

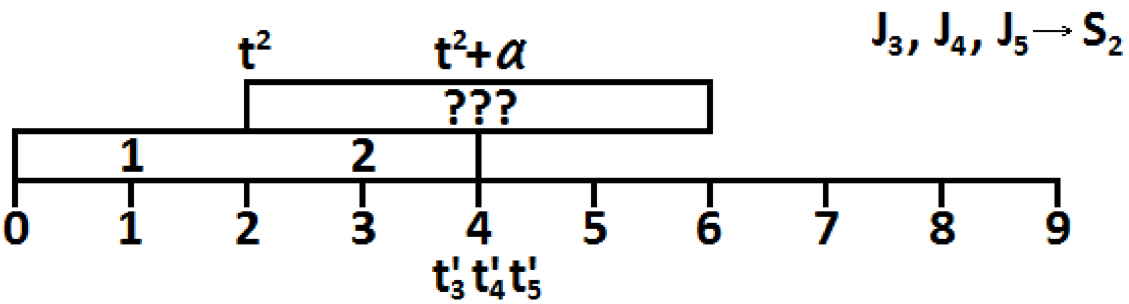


*Рис. 3.Противоречие свойству (iii) при отправке второго поезда.*

В соответствии с условием (ii) получаем $t^2 = 2$, однако есть три заказа, для которых $t^2 + \alpha \geq t'_3, t'_4, t'_5$, а значит, мы включаем заказы $J_3, J_4, J_5$ во множество $S_2$ и строим множества $X^0 = \{J_3, J_4, J_5\}$, $X' = \{J_1\}$ и $X = \{J_2, J_3, J_4, J_5\}$ и пытаемся отправить их первыми двумя поездами. Получаем расписание $\Theta(N, y)$ в котором целевая функция $max(wL) = 15$ достигается на заказе $J_1$ (см рис.4).

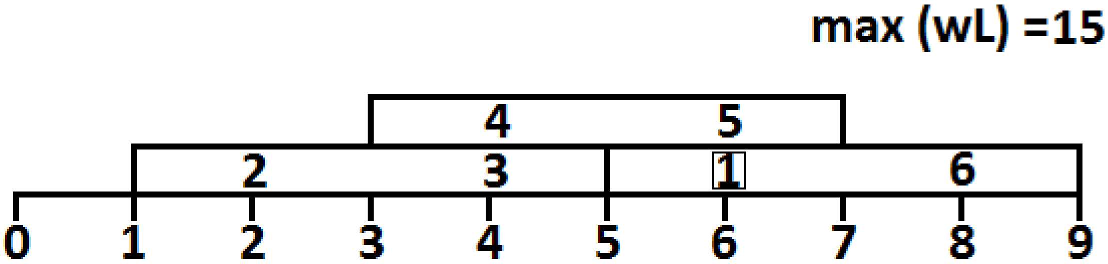


*Рис. 4. Расписание* $\Theta(N, 30)$

**Шаг 3.** Строим расписание $\Theta(N, y)$ для значения $y = 15$. Имеем $t'_1 = 5, t'_2 = 1.5, t'_3 = t'_4 = 2.5, t'_5 = 3.5, t'_6 = 6$. При отправке первого поезда имеем: $t^1 = 0, t^1 + \alpha \geq t'_2 \Rightarrow J_2 \in S_1 \subset S_2$. При попытке отправить второй поезд сталкиваемся с противоречием условию (iii), т.к. $t^2 = 2$, $t^2 + \alpha \geq t'_3, t'_4, t'_5$, следовательно, $J_3, J_4, J_5 \in S_2$. Так же как на шаге 2, строим множества $X^0 = \{J_3, J_4, J_5\}, X' = \{J_1\}$ и $X = \{J_2, J_3, J_4, J_5\}$ и пытаемся отправить заказы из множества $X$ первыми двумя поездами. Так как $r_3 = r_4 = 1$, то $t^1 = 1$, а значит, $t^2 \geq 3$. Из того, что $t^2 + \alpha \geq t'_5$, получаем противоречие условию (iii) при попытке отправить второй поезд. Отсюда следует, что $J_1 \in S_2$. Но тогда $S_2 = \{J_1, J_2, J_3, J_4, J_5\} \Rightarrow |S_2| > 4$, а значит, расписания $\Theta(N, 15)$ не существует.

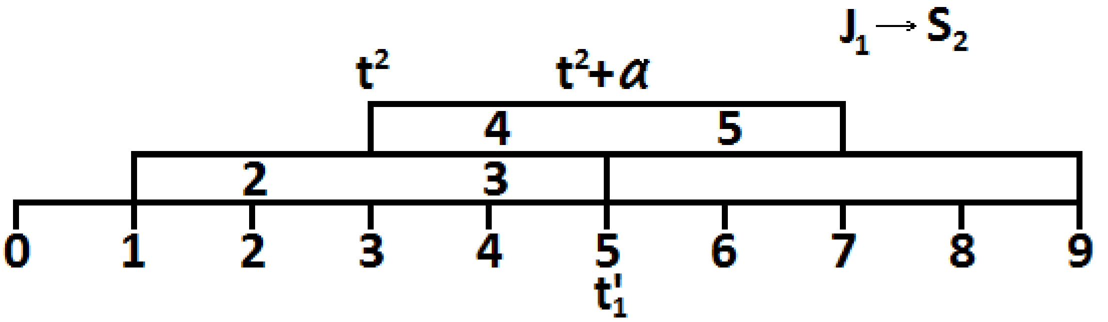


*Рис. 5.Противоречие свойству (iii) при построении расписания* $\Theta(N, 15)$.

Следовательно, расписание $\Theta(N, 30)$, полученное на шаге 2, будет оптимальным, а значение целевой функции будет равно 15.

### *4. Заключение*

Для задачи минимизации максимального взвешенного временного смещения выполнения заказа для двух станций построен полиномиальный алгоритм, трудоемкость которого составляет $O(\frac{n^6}{k})$ операций, где n – количество заказов (вагонов), k – количество вагонов в составе.

### *Литература*

1. *Лазарев А.А.* Теория расписаний. Оценка абсолютной погрешности и схема приближенного решения задач теории расписаний. М.:МФТИ, 2008 г., 222 С.