Автоматика и телемеханика, № 12, 2016

© 2016 г. Д.И. АРХИПОВ (miptrafter@gmail.com)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва),
А.А. ЛАЗАРЕВ, д-р физ.-мат. наук (jobmath@mail.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва,
Национальный исследовательский университет Высшая школа экономики, Москва,
Московский государственный университет им. М.В. Ломоносова,
Московский физико-технический институт)

МИНИМИЗАЦИЯ МАКСИМАЛЬНОГО ВЗВЕШЕННОГО ВРЕМЕННО́ГО СМЕЩЕНИЯ ДОСТАВКИ ЗАКАЗОВ МЕЖДУ ДВУМЯ ЖЕЛЕЗНОДОРОЖНЫМИ СТАНЦИЯМИ[1,2]

Рассматривается задача планирования грузоперевозок между двумя железнодорожными станциями. Требуется выполнить заказы (перевезти вагоны поездами), поступающие в произвольные моменты времени и имеющие различную ценность (вес). Скорость движения поездов между станциями может быть различной. Рассмотрены постановки задачи как с фиксированными, так и с неопределенными моментами отправления поездов. Для задачи с фиксированными моментами отправления поездов представлен алгоритм минимизации взвешенного временно́го смещения заказов трудоемкости $O(qn^2 \log n)$ операций, где q — количество поездов, а n — количество заказов. Для задачи с неопределенными моментами отправления и прибытия поездов построено Парето-множество расписаний оптимальных по критериям $wL_{\max}$ и $C_{\max}$ за $O(n^2 \max\{n \log n, q \log v\})$ операций, где v — количество временны́х окон, в которые возможно отправление поездов. Представленный алгоритм позволяет минимизировать как взвешенное временно́е смещение $wL_{\max}$, так и общее время выполнения заказов на доставку грузов $C_{\max}$.

1. Введение

Задачи оптимизации железнодорожных грузоперевозок являются объектом исследования многих ведущих специалистов и научных групп. Обзор некоторых ранее полученных результатов по данной теме представлен в [1, 2]. Некоторые оптимизационные задачи и их практические приложения для железных дорог Германии и Франции были рассмотрены в [3, 4], соответственно. В [5, 6] рассмотрена задача оптимизации грузоперевозок между двумя железнодорожными станциями для однопутной железной дороги, одинаковой скорости движения поездов и различных критериев оптимальности.

Данная статья является продолжением исследований, результаты которых представлены в [7–9]. В данной статье рассматривается постановка задачи

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты №№ 13-01-12108, 15-07-07489, 15-07-03141), Министерства образования и науки РФ (уникальный идентификатор прикладных научных исследований и экспериментальных разработок – RFMEFI58214X0003) и Deutscher Akademischer Austauschdienst (project A/1400328).

[2] Статьи, опубликованные с 3 по 36 с., являются окончанием тематического выпуска "Методы и алгоритмы решения задач транспортного типа".

с различным количеством вагонов в поездах и нефиксированным временем движения поездов между станциями.

2. Постановки основных задач

Рассматриваются следующие задачи.

З а д а ч а 1. *Минимизация максимального взвешенного временнóго смещения (ВВС) выполнения заказов для двух станций с фиксированными моментами отправления поездов.*

Имеются две станции, соединенные двухпутной железной дорогой. Необходимо выполнить множества заказов $N^1 = \{O_1^1, \ldots, O_n^1\}$ и $N^2 = \{O_1^2, \ldots, O_m^2\}$ 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. Так как железная дорога двухпутная, то расписания для каждого из направлений движения можно составлять отдельно, следовательно, для решения задачи достаточно последовательно составить расписания для множеств N^1 и N^2. Рассматриваемое множество заказов обозначим через $N = \{O_1, \ldots, O_n\}$. Для каждого заказа $O_j \in N$ определены r_j — *время поступления* заказа O_j на станцию отправления и $w_j > 0$ — *ценность* (вес, важность) заказа. Для каждого заказа определен также *директивный срок* d_j — момент времени, до которого заказ желательно доставить на станцию назначения. Без ограничения общности будем считать, что заказы пронумерованы по возрастанию моментов поступления, $r_1 \leqslant r_2 \leqslant \ldots \leqslant r_n$.

Доставка заказов с одной станции на другую осуществляется q поездами, в которых имеется $k_1, \ldots, k_q$ свободных вагонов, $\sum_{i=1}^{q} k_i = n$, и отправляющимися в фиксированные моменты времени $S_1, \ldots, S_q$. Не ограничивая общности, будем полагать что $S_1 < \ldots < S_q$. Скорость движения поездов может быть различной. *Расписанием* π будем называть последовательность пар $\pi = \{(T_1, S_1), \ldots, (T_q, S_q)\}$, где $T_i(\pi)$ — множество заказов, перевозимых поездом i, $i = 1, \ldots, q$, при расписании π, а $S_i(\pi)$ — момент времени отправления поезда i. Для удобства записи будем предполагать, что $S_0 = -\infty$. Расписание π будем называть *допустимым*, если:

1. Для любого поезда $i = 1, \ldots, q$ и любого заказа $O_j \in T_i(\pi)$ выполнено
$$r_j \leqslant S_i(\pi),$$
т.е. момент поступления заказа O_j на станцию отправления не превосходит момента отправления поезда, в который O_j включен.
2. Количество вагонов в поезде $T_i(\pi)$ равно k_i, т.е.
$$|T_i(\pi)| = k_i.$$

Множество всех допустимых расписаний обозначим через $\Pi(N)$. Таким образом, при формировании i-го поезда можно рассматривать только множество заказов $N_i \subseteq N$, поступивших к моменту времени S_i.

Момент времени F_i прибытия поезда i, $i = 1, \ldots, q$, на станцию назначения является фиксированным. Поезд не может догнать или обогнать поезд, который отправился раньше него, т.е. для всех $i = 1, \ldots, q-1$ и $l = 2, \ldots, q$ таких, что $i < l$, будет выполнено неравенство

$$F_i < F_l.$$

Пусть $C_j(\pi)$ — фактический момент времени доставки заказа $O_j \in N$ на станцию назначения при расписании π. Заметим, что $C_j(\pi)$ — момент прибытия заказа, а F_i — момент прибытия поезда на станцию назначения и если $O_j \in T_i(\pi)$, то $C_j(\pi) = F_i$.

Временнóе смещение заказа O_j при расписании π может быть вычислено по формуле

$$L_j(\pi) = C_j(\pi) - d_j.$$

Требуется найти расписание π с минимальным значением максимального взвешенного временнóго смещения (ВВС)

$$\min_{\pi \in \Pi(N)} \max_{O_j \in N} w_j L_j(\pi).$$

Данную задачу можно обозначить как $RS2|r_j, S_i|wL_{\max}$ — так принято в теории расписаний [10]. В разделе 3 для данной задачи сформулирована вспомогательная задача и предложен алгоритм ее решения. Алгоритм построения оптимального решения задачи 1 представлен в разделе 4.

З а д а ч а 2. *Минимизация максимального взвешенного временнóго смещения (ВВС) выполнения заказов для двух станций.*

Рассмотрим задачу, когда моменты отправления поездов не являются фиксированными, а моменты их прибытия на станцию назначения задаются как функции от момента отправления.

Характеристики заказов задаются параметрами, аналогичными рассмотренным ранее. Доставка заказов с одной станции на другую осуществляется q поездами, состоящими из $k_1, \ldots, k_q$ вагонов, также предполагается, что $\sum_{i=1}^{q} k_i = n$.

Расписанием π будем называть последовательность пар

$$\pi = \{(T_1(\pi), S_1(\pi)), \ldots, (T_q(\pi), S_q(\pi))\},$$

где $T_i(\pi)$ — множество заказов, перевозимых поездом i, $i = 1, \ldots, q$, а $S_i(\pi)$ — момент времени отправления поезда i при расписании π. Скорость движения поездов может быть различной. В случае когда понятно, о каком расписании идет речь, моменты отправления и множество заказов, перевозимых поездом i, будем обозначать через S_i и T_i соответственно.

Обозначим через $\widehat{u}$ время, которое должно разделять моменты отправления двух поездов. Движение пассажирских поездов и электричек, плановые ремонтные работы и другие ограничения учитываются с помощью множества *допустимых интервалов отправления* $U = \{[u_1^s, u_1^f), \ldots, [u_v^s, u_v^f)\}$, где v — общее количество интервалов, когда может быть осуществлено отправление поезда (рис. 1). Расписание π будем называть *допустимым*, если:

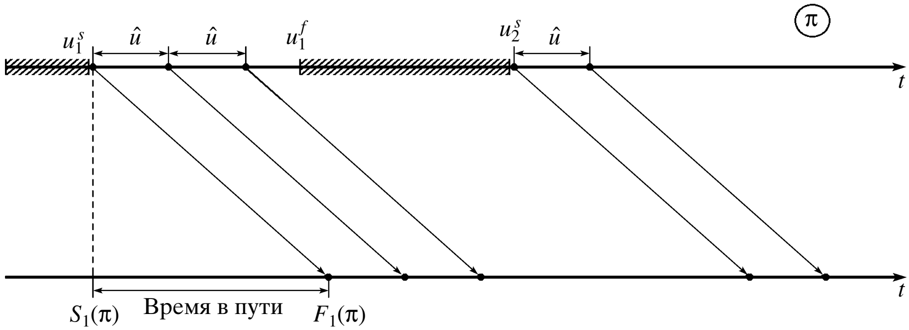

Рис. 1. Условные обозначения задачи 2.

1. момент отправления каждого поезда принадлежит одному из допустимых интервалов отправления, т.е. для любого $i = 1, \ldots, q$ выполнено
$$S_i(\pi) \in U;$$
2. моменты отправления любых двух поездов должны отличаться не менее чем на $\widehat{u}$, т.е. $\forall i,\ j = 1, \ldots, q,\ i > j$,
$$S_i(\pi) - S_j(\pi) \geqslant \widehat{u};$$
3. количество вагонов в поезде $T_i(\pi)$ равно k_i, т.е.
$$|T_i(\pi)| = k_i.$$

Множество всех допустимых расписаний обозначим через $\Pi(N, U)$. Время прибытия i-го поезда, $i = 1, \ldots, q$, на станцию назначения может быть вычислено по формуле
$$F_i(\pi) = f_i(S_i(\pi)),$$
где $f_i(t)$ — заданная монотонно неубывающая функция от времени t. При любом расписании π поезд не может догнать или обогнать поезд, который отправился раньше него, т.е. для всех $i = 1, \ldots, q-1$ и $l = 2, \ldots, q$ таких, что $i < l$, будет выполнено неравенство

$$f_i(S_i(\pi)) < f_l(S_l(\pi)). \tag{1}$$

Пусть $C_j(\pi)$ — момент времени доставки заказа $O_j \in N$ на станцию назначения при расписании π.

Временнóе смещение заказа O_j при расписании π может быть вычислено по формуле
$$L_j(\pi) = C_j(\pi) - d_j.$$

Требуется найти расписание π с минимальным значением максимального взвешенного временнóго смещения (ВВС)
$$\min_{\pi \in \Pi(N,U)} \max_{O_j \in N} w_j L_j(\pi).$$

Так как количество расписаний, удовлетворяющих данной целевой функции, может быть больше одного, то выберем такое расписание, при котором время доставки всех заказов минимально. Пусть

$$y^* = \min_{\pi \in \Pi(N,U)} \max_{O_j \in N} w_j L_j(\pi).$$

Таким образом, требуется построить расписание, оптимальное по критерию

$$\min_{\pi \in \Pi(N,U)} \max_{O_j \in N} C_j(\pi) | \max_{O_j \in N} w_j L_j(\pi) = y^*. \tag{2}$$

Данную задачу можно обозначить как $RS2|r_j|C_{\max}, wL_{\max}$. В разделе 5 статьи для этой задачи сформулирована вспомогательная задача и рассмотрены некоторые ее свойства. Алгоритм решения вспомогательной задачи представлен в разделе 6. В разделе 7 статьи предложен алгоритм построения оптимального расписания для решения задачи 2. Анализ трудоемкости алгоритма решения задачи 2 представлен в разделе 8.

3. Задача нахождения расписания с ограничением на максимальное ВВС

Сформулируем вспомогательную задачу для задачи 1. Пусть задано ограничение на ВВС y. Расписание $\pi \in \Pi(N)$ будем называть *разрешенным,* если для него выполняется неравенство

$$\max_{O_j \in N} w_j L_j(\pi) < y.$$

Множество разрешенных расписаний для заданного ограничения y будем обозначать через $\Phi(N, y)$. Требуется построить расписание $\pi \in \Phi(N, y) \subseteq \Pi(N)$ или доказать, что $\Phi(N, y) = \emptyset$.

Заметим, что заказ $O_j \in N$ не будет нарушать ограничение y тогда, когда он принадлежит составу T_i, $i = 1, \ldots, q$, и выполнено неравенство

$$F_i < d_j + \frac{y}{w_j},$$

т.е. момент прибытия поезда на станцию назначения меньше, чем момент времени $d_j + \frac{y}{w_j}$. Если бы заказ O_j прибыл на станцию назначения в момент времени t, такой что

$$t \geqslant d_j + \frac{y}{w_j},$$

то ВВС заказа O_j было бы равно

$$(t - d_j) w_j \geqslant \left(d_j + \frac{y}{w_j} - d_j \right) w_j = y.$$

Таким образом, ограничение на ВВС было бы нарушено.

Для решения данной задачи предлагается следующий алгоритм.

Алгоритм 1. На первом шаге для каждого заказа $O_j \in N$ найдем поезд с наибольшим номером $T^{\max}(j, y)$, в который может быть включен данный заказ, не нарушив ограничения на ВВС y:

$$T^{\max}(j, y) = \arg \max_{i=1,\ldots,q} F_i | F_i < d_j + \frac{y}{w_j}.$$

Далее будем последовательно формировать множества (составы) $T_1(\pi(y))$, $T_2(\pi(y)), \ldots, T_q(\pi(y))$ по следующему правилу. В состав $T_i(\pi(y))$, $i = 1, \ldots, q$, могут быть включены только заказы из N_i, где $N_i \subseteq N$ — множество заказов, поступивших к моменту отправления i-го поезда S_i. Множество $T_i(\pi(y))$, $i = 1, \ldots, q$, состоит из k_i заказов $O_{i_1}(\pi(y)), \ldots, O_{i_{k_i}}(\pi(y)) \in N_i \setminus \{T_1(\pi(y)), \ldots$ $\ldots, T_{i-1}(\pi(y))\}$, для которых выполнено

$$T^{\max}(i_1, y) \leqslant T^{\max}(i_2, y) \leqslant \ldots \leqslant T^{\max}(i_{k_i}, y)$$

и для любого $O_j \in N_i \setminus \{T_1(\pi(y)), \ldots, T_i(\pi(y))\}$ выполняется неравенство

$$T^{\max}(i_{k_i}, y) \leqslant T^{\max}(j, y).$$

Таким образом, в формируемый состав будут включены k_i заказов с наименьшими значениями $T^{\max}(j, y)$, не принадлежащие ранее сформированным составам.

В случае если при формировании i-го состава для некоторого заказа $O_j(\pi(y)) \in T_i(\pi(y))$ выполнено

$$T^{\max}(j, y) < i,$$

то заказ O_j нарушит ограничение на ВВС. Следовательно, не существует расписания $\pi(y) \in \Phi(N, y)$, удовлетворяющего ограничению y, и алгоритм завершает работу.

Если при формировании поезда i, $i = 1, \ldots, q$, количество заказов, поступивших до момента отправления, недостаточно, т.е.

$$|N_i \setminus \{T_1(\pi(y)), \ldots, T_{i-1}(\pi)\}| < k_i,$$

то из того что $|T_1(\pi(y))| = k_1$, $|T_2(\pi(y))| = k_2$, $\ldots$, $|T_{i-1}(\pi(y))| = k_{i-1}$, следует, что

$$|N_i| < \sum_{l=1}^{i} k_l,$$

т.е. невозможно сформировать состав i. Таким образом, допустимых расписаний не существует и $\Pi(N) = \emptyset$, а значит, и $\Phi(N, y) = \emptyset$. В этом случае алгоритм завершает работу. Таким образом, алгоритм заканчивает работу, когда построено допустимое расписание, либо такое расписание нельзя построить.

Формальная запись алгоритма 1 выглядит следующим образом.

0. Входные данные:
   $N, q, y, k_1, \ldots, k_q, S_1, \ldots, S_q, F_1, \ldots, F_q$.

1. Для всех заказов $O_j \in N$ вычисляются значения

$$T^{\max}(j, y) = \arg \max_{i=1,\ldots,q} F_i | F_i < d_j + \frac{y}{w_j}.$$

Также выполняется включение заказов $O_j \in N$ в множества N_i, $i = 1, \ldots$ $\ldots, q$, по правилу

$$O_j \in N_i \Leftrightarrow S_i \geqslant r_j.$$

2. Последовательно формируем составы $T_i(\pi)$ для $i = 1, \ldots, q$:
   а. Если для данного i выполнено неравенство

   $$|N_i \setminus \{T_1(\pi), \ldots, T_{i-1}(\pi)\}| < k_i,$$

   то выполняется присвоение $\pi(y) = \emptyset$ и алгоритм завершает работу. Иначе, переходим на шаг 2б.

   б. Включаем в множество $T_i(\pi)$ k_i заказов из множества $N_i \setminus \{T_1(\pi), \ldots$ $\ldots, T_{i-1}(\pi)\}$ с наименьшими значениями $T^{\max}(i_1, y), \ldots, T^{\max}(i_{k_i}, y)$. Если для некоторого заказа $O_j \in T_i(\pi)$ справедливо

   $$T^{\max}(j, y) < i,$$

   то выполняется присвоение $\pi(y) = \emptyset$ и алгоритм завершает работу. Иначе, переходим на шаг 2в.

   в. Добавляем сформированный состав к расписанию $\pi(y)$:

   $$\pi(y) := \pi(y) \cup \{T_i(\pi), S_i\};$$

   г. Если $i \neq q$, то выполняется присвоение $i := i + 1$ и переход на шаг 2а. Иначе, расписание $\pi(y)$ построено и алгоритм завершает работу.

Теорема 1. *Если в результате работы алгоритма 1 построено расписание $\pi(y) = \emptyset$, то расписаний, удовлетворяющих ограничению y для данного набора заказов, не существует, т.е. $\Phi(N, y) = \emptyset$.*

Если $\Phi(N, y) \neq \emptyset$, то в результате работы алгоритма 1 будет построено расписание $\pi(y) \in \Phi(N, y)$, удовлетворяющее неравенству

$$\max_{O_j \in N} w_j L_j(\pi(y)) < y.$$

Доказательство теоремы 1. Пусть в результате выполнения алгоритма 1 построено расписание $\pi(y) = \emptyset$. Предположим, что алгоритм остановился при формировании i-го поезда. Если алгоритм остановился на шаге 2а, то выполняется неравенство

$$|N_i \setminus \{T_1(\pi(y)), \ldots, T_{i-1}(\pi(y))\}| < k_i,$$

а значит, допустимых расписаний не существует и $\Pi(N) = \emptyset$. Следовательно, $\Phi(N, y) \subseteq \Pi(N) = \emptyset$.

Если алгоритм остановился на шаге 2б, то из алгоритма 1 следует, что при формировании поездов с номерами $l = 1, 2, \ldots, i-1$ на шаге 2б в состав $T_l(\pi(y))$ включалось максимально возможное количество заказов среди поступивших к моменту S_l, для которых выполняется неравенство $T^{\max}(j, y) < i$. Таким образом, к моменту отправления i-го поезда S_i поступило меньше, чем $\sum_{l=1}^{i} k_l$, заказов j, для которых выполнено

$$T^{\max}(j, y) \geqslant i,$$

следовательно, $\Phi(N, y) = \emptyset$.

Если же в результате работы алгорита построено расписание $\pi(y)$, то так как при нем любой заказ $O_j \in N$ отправляется поездом с номером, не большим чем $T^{\max}(j, y)$, то расписание $\pi(y) \in \Phi(N, y)$ удовлетворяет ограничению

$$\max_{O_j \in N} w_j L_j(\pi(y)) < y.$$

Теорема 1 доказана.

Лемма 1. Трудоемкость алгоритма 1 решения вспомогательной задачи $RS2|r_i, S_j|wL_{\max} < y$ составляет $O(n \log n)$ операций, где n — количество заказов.

Доказательство леммы 1. Наиболее трудоемкой частью алгоритма 1 является вычисление и сортировка n значений $T^{\max}(1, y), \ldots, T^{\max}(n, y)$ на шаге 1, для которых требуется $O(n \log q)$ и $O(n \log n)$ операций соответственно. Следовательно, так как $q \leqslant n$, то трудоемкость не превышает $O(n \log n)$ операций.

4. Алгоритм решения задачи минимизации ВВС при фиксированных моментах отправления поездов

Для нахождения оптимального расписания по критерию

$$\min_{\pi \in \Pi(N)} \max_{O_j \in N} w_j L_j(\pi).$$

будем действовать следующим образом.

Алгоритм 2. На первом шаге с помощью алгоритма 1 построим расписание $\pi_1 = \pi(y_1)$ для значения ограничения $y_1 = +\infty$ и вычислим значение $wL_{\max}(\pi_1)$. Затем будем строить расписания $\pi_2 = \pi(y_2)$ для значения ограничения $y_2 = wL_{\max}(\pi_1)$. Последовательно строим расписания $\pi_3 = \pi(y_3), \ldots, \pi_l = \pi(y_l)$, где $y_i = wL_{\max}(\pi_{i-1})$, $i = 3, \ldots, l$, до тех пор, пока алгоритм 1 не завершится с результатом $\pi(wL_{\max}(\pi_l)) = \emptyset$. В этом случае работа алгоритма 2 завершается, результатом работы является расписание $\pi^*(N) = \pi_l$, построенное на предыдущем шаге. Далее будет показано, что количество построенных расписаний $\pi_1, \ldots, \pi_l$ не превосходит $n(q-1)$.

Формальная запись алгоритма 2 выглядит следующим образом.

0. Входные данные:
   $N, q, y_1 = +\infty,\ i = 1, k_1, \ldots, k_q, S_1, \ldots, S_q, F_1, \ldots, F_q.$

1. Выполняется построение расписания $\pi_i = \pi(y_i)$ с помощью алгоритма 1.
2. Если $\pi_i \neq \emptyset$, то
   а. Присваиваем $\pi^*(N) = \pi_i$;
   б. Присваиваем $y_{i+1} = wL_{\max}(\pi_i)$, $i := i + 1$ и переходим на шаг 1.
3. Если $\pi_i = \emptyset$, то алгоритм возвращает расписание $\pi^*(N)$ и завершает работу.

Теорема 2. Расписание, построенное в результате работы алгоритма 2, является оптимальным по критерию

$$\min_{\pi \in \Pi(N)} \max_{O_j \in N} w_j L_j(\pi).$$

Если для расписания $\pi^*(N)$ выполняется

$$\pi^*(N) = \emptyset,$$

то $\Pi(N) = \emptyset$, т.е. допустимого расписания не существует.

Доказательство теоремы 2. Из работы алгоритма 2 следует, что расписания $\pi(wL_{\max}(\pi^*(N)))$ не существует, т.е. по теореме 1 имеем $\Phi(N, wL_{\max}(\pi^*(N)) = \emptyset$. Следовательно, для любого $\pi' \in \Pi(N)$ выполнено

$$\max_{O_j \in N} w_j L_j(\pi') \geqslant \max_{O_j \in N} w_j L_j(\pi^*(N)).$$

Первая часть теоремы 2 доказана.

Если

$$\pi^*(N) = \emptyset,$$

то не существует расписания $\pi(+\infty)$ и по теореме 1 имеем $\Phi(N, +\infty) = \emptyset$. Тогда, так как $\Phi(N, +\infty) = \Pi(N)$, то нет ни одного допустимого расписания, т.е. $\Pi(N) = \emptyset$. Теорема 2 доказана.

Лемма 2. Трудоемкость алгоритма 2 составляет $O(qn^2 \log n)$ операций, где n — количество заказов, q — количество составов.

Доказательство леммы 2. Без ограничения общности будем считать, что заказ $O_j \in N$, на котором было достигнуто значение целевой функции $wL_{\max}(\pi_i)$ при построении расписания π_i, был отправлен поездом a, т.е. $O_j \in T_a(\pi_i)$. Тогда при любом расписании $\pi(y)$ при значении $y < wL_{\max}(\pi_i)$ для заказа O_j будет выполнено

$$T^{\max}(O_j, y) < a.$$

Таким образом, при каждом проходе цикла 1–2 алгоритма 2 один из заказов должен быть отправлен поездом с меньшим номером, чем при расписании, построенном на предыдущем проходе цикла 1–2. Всего заказов n, для каждого заказа значение $T^{\max}(O_j, y)$ при изменении значения y может уменьшаться не более чем $q - 1$ раз. Следовательно, если $\pi^*(N) = \pi_l$, то $l \leqslant n(q - 1)$. Следовательно, алгоритм 1 решит вспомогательную задачу не более чем $n(q-1)$ раз. Трудоемкость решения вспомогательной задачи с помощью алгоритма 1 составляет $O(n \log n)$ операций. Таким образом, общая трудоемкость алгоритма 2 составляет $O(qn^2 \log n)$ операций. Лемма 2 доказана.

Для задачи 1 минимизации максимального ВВС выполнения заказов для двух станций с фиксированными моментами отправления поездов предложен алгоритм нахождения оптимального расписания трудоемкостью $O(qn^2 \log n)$ операций, где q — количество поездов, а n — количество заказов.

5. Вспомогательная задача для задачи 2

Прежде чем сформулировать вспомогательную задачу, введем дополнительные обозначения.

Введем функцию $\lceil t \rceil$, определенную на множестве моментов времени t для заданного множества допустимых интервалов отправления $U = \{[u_1^s, u_1^f), \ldots$ $\ldots, [u_v^s, u_v^f)\}$. Для любого момента времени t значение функции вычисляется по формуле

$$\lceil t \rceil = \min_{t' \in U} t' | t' \geqslant t.$$

Таким образом, $\lceil t \rceil \in U$ — это минимальный момент времени, не меньше t, принадлежащий ближайшему допустимому интервалу отправления. При этом для любого $t > u_v^f$ значение функции $\lceil t \rceil$ будет равно $+\infty$.

Введем функцию $r(N')$, определенную на семействе подмножеств N' множества заказов N, $N' \subseteq N$, и заданную формулой

$$r(N') = \max_{O_j \in N'} r_j,$$

т.е. $r(N')$ — момент времени, к которому поступят все заказы множества N'.

Определим семейство множеств $Q = \{N_0, N_1, \ldots, N_q\}$ следующим образом. Пусть N_0 — множество заказов, которые не могут быть отправлены ни одним из поездов, N_1 — заказы, которые должны быть отправлены первым поездом (вместе с тем N_0 включено в N_1), N_2 — первым или вторым поездом и т.д. Таким образом, $N_0 \subseteq N_1 \subseteq \ldots \subseteq N_q$, $N_q = N$. Для любого $i = 0, \ldots, q$ мощность множества N_i не может превышать количества заказов, которые могут быть отправлены первыми i поездами, т.е.

$$|N_i| \leqslant K_i, \quad i = 0, \ldots, q, \tag{3}$$

где $K_i = \sum_{j=1}^{i} k_j$ — суммарное количество заказов (вагонов), перевозимых поездами $1, \ldots, i$, а $K_0 = 0$. Очевидно, имеем $N_q = N$. Заметим, что определение множеств $N_0, \ldots, N_q$ отличается от того, которое было сформулировано при решении задачи 1. В данном случае множества не зависят от моментов поступления $r_1, \ldots, r_n$ или от других параметров заказов.

Для всех заказов $j = 1, \ldots, n$ обозначим через $N(O_j) = i$, $i = 1, \ldots, q$, номер множества N_i, если для данного семейства множеств Q заказ O_j принадлежит множеству N_i и не принадлежит N_{i-1}, т.е.

$$N(O_j) = \min_{i=0,\ldots,q} i | O_j \in N_i.$$

Будем говорить, что расписание π *удовлетворяет* семейству множеств $Q = \{N_0, \ldots, N_q\}$, если для любого $i = 1, \ldots, q$ выполняется $N_i \subseteq T_1(\pi) \cup \ldots \ldots \cup T_i(\pi)$, где $T_1(\pi), \ldots, T_i(\pi)$ — множества заказов, перевозимых поездами $1, \ldots, i$ при расписании π.

Сформулируем вспомогательную задачу для задачи 2. Пусть задано множество заказов N и ограничение на ВВС y. Необходимо составить допустимое расписание $\Theta(N, U, Q, y) \in \Pi(N, U)$, удовлетворяющее семейству множеств Q и критерию

$$\min_{\pi \in \Pi(N,U)} C_{\max}(\pi) | \max_{O_j \in N} w_j L_j(\pi) < y,$$

где

$$C_{\max}(\pi) = \max_{O_j \in N} C_j(\pi),$$

либо показать, что для данного значения y не существует допустимого расписания. Для решения данной задачи приме́ним подход, аналогичный методу Бенсона [11]. Для решения задач теории расписаний данный подход впервые был применен для задачи $1|r_j|L_{\max}$ в [12].

Введем дополнительные обозначения. Из ограничения на максимальное ВВС для любого $O_j \in N$ имеем

$$w_j(C_j(\pi) - d_j) < y,$$

следовательно,

$$C_j(\pi) < \frac{y}{w_j} + d_j.$$

При заданном значении ограничения y для каждого заказа $O_j \in N$ может быть определен *дедлайн* — момент времени, до которого данный заказ должен быть доставлен на станцию назначения:

$$D_j(y) = \frac{y}{w_j} + d_j.$$

Если значение времени доставки $C_j(\pi)$ не превышает $D_j(y)$, то заказ O_j не нарушает ограничения на ВВС при расписании π. Расписание π будем называть *разрешенным*, если для любого заказа $O_j \in N$, перевозимого поездом $T_i(\pi)$, момент поступления заказа r_j не превышает момента отправления поезда $S_i(\pi)$ и момент прибытия поезда $F_i(\pi)$ не превышает дедлайна заказа $D_j(y)$, т.е. выполнены неравенства:

$$S_i(\pi) \geqslant r_j,$$
$$F_i(\pi) < D_j(y).$$

Множество всех разрешенных расписаний для заданного ограничения y обозначим через $\Phi(N, U, y) \subseteq \Pi(N, U)$.

Вспомогательная задача может быть сформулирована следующим образом: имеется множество заказов $O_j \in N$, для каждого из которых определены момент поступления r_j и дедлайн $D_j(y)$. Также имеется множество интервалов допустимого отправления U и семейство множеств Q. Требуется найти расписание $\Theta(N, U, Q, y) \in \Phi(N, U, y)$, для которого выполнено равенство

$$C_{\max}(\Theta(N, U, Q, y)) = \min_{\pi \in \Phi(N,U,y)} C_{\max}(\pi) \tag{4}$$

при условии, что π удовлетворяет Q.

Рассмотрим дополнения $\overline{N_i} = N \setminus N_i$ ко всем множествам $N_i \in Q$. Из определения множества N_i следует, что любой поезд с номером больше i должен состоять только из заказов, принадлежащих множеству $\overline{N_i}$, т.е. для любого $l = i + 1, \ldots, q$ будет выполнено $T_l \subseteq \overline{N_i}$.

З а м е ч а н и е 1. При любом расписании $\pi \in \Phi(N, U, y)$, удовлетворяющем семейству множеств $Q = \{N_0, \ldots, N_q\}$, для любого $i = 1, \ldots, q$ момент отправления поезда S_i и множество заказов поезда T_i должны удовлетворять таким свойствам:

1) к моменту времени $r(T_i)$ должны поступить хотя бы $K_i = \sum_{j=1}^{i} k_j$ заказов, в том числе все заказы из N_i, т.е.

$$r(T_i) \geqslant \max\{r(N_i), r_{K_i}\};$$

2) момент времени S_i отправления поезда i удовлетворяет неравенству

$$S_i \geqslant \lceil \max\{r(T_i), S_{i-1} + \widehat{u}\} \rceil. \tag{5}$$

6. Алгоритм решения вспомогательной задачи

Алгоритм построения расписания $\Theta(N, U, Q, y)$, оптимального по критерию $C_{\max}$ для заданного множества заказов N, множества допустимых интервалов отправки U, семейства множеств $Q = \{N_0, \ldots, N_q\}$ и ограничения на ВВС y, заключается в следующем.

А л г о р и т м 3. Для всех заказов $O_j \in N$ при заданном значении ограничения y вычисляются значения дедлайнов $D_j(y)$ и выполняются присвоения $N_0^0 = N_0, \ldots, N_q^0 = N_q$, после чего алгоритм выполняется итерационно. На m-й итерации алгоритма последовательно выпоняются следующие шаги. На первом шаге рассматривается формирование последнего поезда q (рис. 2).

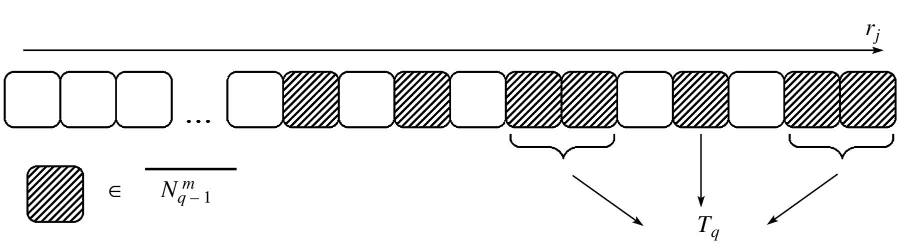

Рис. 2. Формирование поезда q.

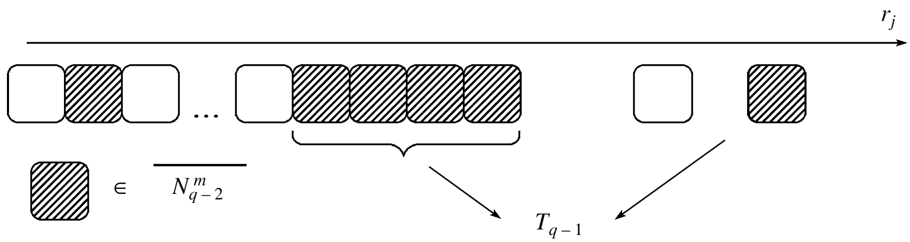

Рис. 3. Формирование поезда $q-1$.

С учетом того, что $T_q \subseteq \overline{N_{q-1}^m}$ и $|T_q| = k_q$, выбираем из множества $\overline{N_{q-1}^m}$ необходимое количество заказов k_q с наибольшими значениями моментов поступления r_j и включаем их в поезд q. После этого переходим к формированию поезда $q-1$.

Так как один заказ не может быть включен в два разных поезда, то поезд $q-1$ может перевезти только заказы из $\overline{N_{q-2}^m}$, не включенные в T_q, т.е. $T_{q-1} \subseteq (\overline{N_{q-2}^m} \setminus T_q)$ (рис. 3). Выбираем k_{q-1} заказов из множества $\overline{N_{q-2}^m} \setminus T_q$ и включаем их в T_{q-1}. На каждом следующем шаге поступаем аналогично. Таким образом, при формировании поезда i выбираем k_i заказов с наибольшими моментами поступления из множества $\overline{N_{i-1}^m} \setminus (T_{i+1} \cup T_{i+2} \cup \ldots \cup T_q)$ и включаем их в T_i. Так как для любого $l = 1, \ldots, q$ выполнено условие (3), то для любого $l = 2, \ldots, q$ будет выполнено неравенство

$$|\overline{N_{l-1}^m}| \geqslant \sum_{j=l}^{q} k_j.$$

После того как все поезда сформированы, вычисляем моменты отправления S_i поездов $i = 1, \ldots, q$ по формуле

$$S_i = \lceil \max\{r(T_i), S_{i-1} + \widehat{u}\} \rceil.$$

Последовательно находим значения $S_1 = \lceil r(T_1) \rceil$, $S_2 = \lceil \max\{r(T_2), S_1 + \widehat{u}\} \rceil$, $\ldots$, $S_q = \lceil \max\{r(T_q), S_{q-1} + \widehat{u}\} \rceil$.

Для всех заказов O_j, $j = 1, \ldots, n$, поочередно пересчитываем $N(O_j)$, последовательно проверяя неравенства

$$F_i(\pi) \geqslant D_j(y)$$

для $i = N(O_j), \ldots, 1$. Если данное неравенство верно для $i = l$, где $O_j \in T_l(\pi)$, то построенное расписание не удовлетворяет ограничению $wL_{\max}(\pi) < y$. Если неравенство выполняется, то заказ O_j включается в множество N_{i-1}^m и выполняется присвоение значения $N(O_j) = i - 1$ (рис. 4). Если же неравенство не выполняется, то так как $F_i(\pi) \geqslant F_{i-1}(\pi) \geqslant \ldots \geqslant F_1(\pi)$, то дальнейшая проверка не имеет смысла.

После того как проверка выполнена для всех O_j, $j = 1, \ldots, n$, и новые множества $N_1^m, N_2^m, \ldots N_q^m$ составлены, проверяем их мощность. В случае

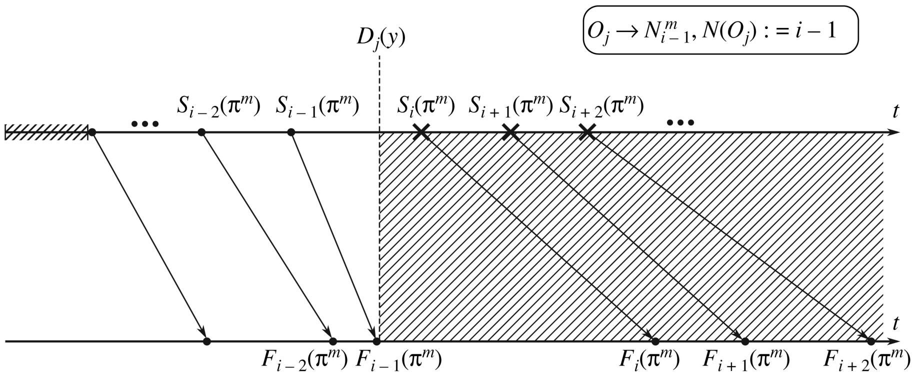

Рис. 4. Включение заказа O_j в N_{i-1}^m.

если для какого-либо множества i, $i = 1, \dots, q$, верно неравенство

$$|N_i^m| > K_i,$$

то алгоритм прерывается. Таким образом, нарушается условие (3), следовательно, разрешенного расписания $\Theta(N, U, Q, y)$ не существует. Если мощности всех множеств удовлетворяют данным ограничениям, то выполняются присвоения $m := m + 1$ и $N_i^m := N_i^{m-1}$ для всех $i = 1, \dots, q$, после чего происходит переход на следующую итерацию. Выполнение алгоритма продолжается до тех пор, пока не будет построено расписание π такое, что для любого $O_j \in T_l(\pi)$, $j = 1, \dots, n$, выполнено

$$F_l(\pi) < D_j(y)$$

или не будет нарушено условие (3).

Формальная запись алгоритма 3 выглядит следующим образом.

0. Входные данные:
   $N, U, Q, y, q, m = 0, k_1, \dots, k_q$.
1. Для всех заказов $O_j \in N$ вычисляем значения $D_j(y) := \frac{y}{w_j} + d_j$, $N(O_j) = = \min_{i=1,\dots,q} i | O_j \in N_i$. Для всех $i = 0, 1, \dots, q$ присваиваем $N_i^m := N_i$, где $N_i \in Q$.
2. Для значений $i = q, \dots, 1$ последовательно формируем составы:
   в состав $T_i(\pi^m)$ включаем k_i заказов из множества $\overline{N_{i-1}^m} \setminus (T_{i+1}(\pi^m) \cup \dots \dots \cup T_q(\pi^m))$ с наибольшими значениями моментов поступления и вычисляем значения $r(T_i(\pi))$.
3. Для значений $i = 1, \dots, q$ вычисляем моменты отправления поездов по формуле

   $$S_i(\pi^m) = \lceil \max\{r(T_i(\pi^m)), S_{i-1}(\pi^m) + \widehat{u}\} \rceil,$$

   учитывая $S_0(\pi^m) = -\infty$.
4. Для заказов с номерами $j = 1, \dots, n$ и поездов $i = N(O_j), \dots, 1$, последовательно проверяем выполнимость неравенства

   $$F_i(\pi^m) < D_j(y) :$$

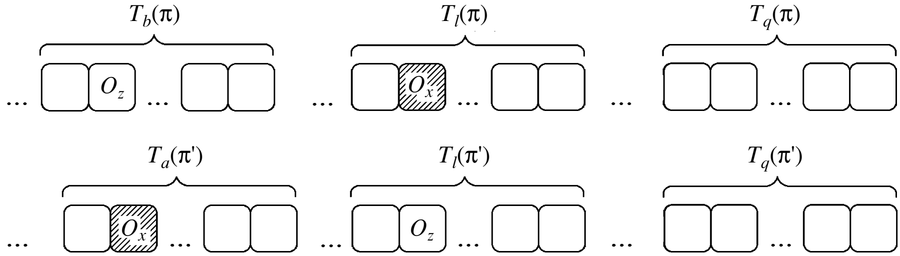

Рис. 5. Отличие в составах $T_l(\pi)$ и $T_l(\pi')$.

а. Если для какой-то пары j и i, неравенство не верно, то выполняется включение заказа O_j в множество N_{i-1}^m. Это приводит к изменению множества Q, после чего выполняется пересчет значения $N(O_j) := i - 1$. После этого проверка продолжается.

б. Если для всех пар j, i, где $j = 1, \ldots, n$, $O_j \in T_i(\pi)$, неравенство верно, то

$$\Theta(N, U, Q, y) = \pi^m$$

и алгоритм завершает работу.

5. Последовательно проверяем выполнимость условия (3)

$$|N_i^m| \leqslant K_i$$

для значений $i = 0, 1, \ldots, q$:

а. Если для всех множеств условие (3) выполнено, то присваиваем $m := m + 1$, для значений $i = 0, 1, \ldots, q$, присваиваем $N_i^m := N_i^{m-1}$ и переходим на шаг 2.

б. Если для какого-то множества условие (3) не выполнено, то алгоритм завершает работу и $\Theta(N, U, Q, y) = \emptyset$.

Лемма 3. Пусть π — расписание, построенное на некоторой итерации алгоритма 3 для набора множеств Q. Тогда для любого расписания $\pi' \in \in \Phi(N, U, y)$, удовлетворяющего множествам Q, для любого поезда $i = 1, \ldots \ldots, q$ выполняется неравенство

$$S_i(\pi) \leqslant S_i(\pi'),$$

т.е. при расписании π каждый из q поездов отправляется в минимально возможное время.

Доказательство леммы 3. Будем сравнивать составы поездов расписаний π и π', перебирая их в обратном порядке. Пусть на каком-то шаге среди множества заказов $T_l(\pi)$, перевозимых поездом l, нашелся заказ O_x, который отсутствует в $T_l(\pi')$ (рис. 5). Так как $k_l = |T_l(\pi)| = |T_l(\pi')|$, то в составе $T_l(\pi')$ должен присутствовать заказ O_z, который отсутствует в $T_l(\pi)$.

Заметим, что $T_l(\pi) \neq T_l(\pi')$ — первое из встретившихся отличий, т.е. $T_{l+1}(\pi) = T_{l+1}(\pi'), \ldots, T_q(\pi) = T_q(\pi')$. Из того что заказ $O_x \in T_l(\pi)$, но отсутствует в множествах $T_l(\pi'), T_{l+1}(\pi'), \ldots, T_q(\pi')$, следует, что при расписании π'

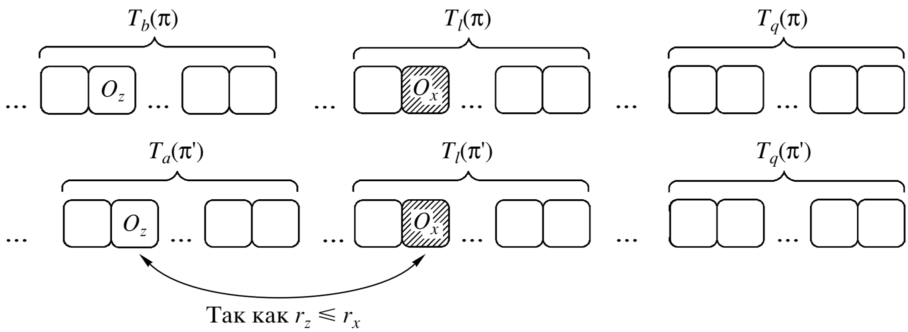

Рис. 6. Замена при расписании π'.

заказ включен в состав $T_a(\pi')$, где $a < l$. Аналогичные рассуждения проводим для заказа O_z, делая вывод, что при расписании π O_z включен в состав $T_b(\pi)$, где $b < l$ (рис. 5).

Так как при формировании состава $T_l(\pi)$ алгоритмом 3 были выбраны заказы с наибольшими значениями r_j, можно сделать вывод, что $r_x \geqslant r_z$, и так как $O_x \in T_a(\pi')$, то

$$S_a(\pi') \geqslant r_x \geqslant r_z.$$

Кроме того, $O_x, O_z \in \overline{N_{l-1}}$, так как оба расписания удовлетворяют набору множеств Q. Отсюда следует, что при расписании π' можно поменять заказы O_x и O_z местами (поездами) без сдвига моментов отправления поездов $S_1(\pi'), S_2(\pi'), \ldots, S_q(\pi')$ и расписание будет удовлетворять набору множеств Q (рис. 6).

Последовательно выполняя аналогичные операции, получаем расписание π'', удовлетворяющее множествам Q с составами $T_1(\pi), T_2(\pi), \ldots, T_q(\pi)$ и моментами отправления $S_1(\pi'), S_2(\pi'), \ldots, S_q(\pi')$. Последовательно применяя замечание 1 к моментам отправления $S_1(\pi''), \ldots, S_q(\pi'')$, получаем

$$S_1(\pi'') \geqslant r(T_1(\pi)) = S_1(\pi),$$

и для каждого значения $i = 2, \ldots, q$ выполняется

$$S_i(\pi'') \geqslant \lceil \max\{r(T_i(\pi)), S_{i-1}(\pi) + \widehat{u}\} \rceil = S_i(\pi),$$

т.е. для любого $i = 1, \ldots, q$

$$S_i(\pi') = S_i(\pi'') \geqslant S_i(\pi).$$

Лемма 3 доказана.

Лемма 4. Пусть на некоторой итерации алгоритма 3 для семейства множеств Q было построено расписание π и в результате изменений множеств на шаге 4 для данного значения y были получены множества $Q' = \{N'_0, N'_1, \ldots, N'_q\}$. Тогда любое расписание $\pi' \in \Phi(N, U, y)$, удовлетворяющее семейству множеств Q, удовлетворяет Q'.

Доказательство леммы 4. По правилу включения на шаге 4 алгоритма 3 для любого заказа O_j и множеств N_i, N'_i, $i = 0, 1, \ldots, q$, таких что $O_j \in N_i$ и $O_j \notin N'_i$, должно быть выполнено

$$F_{i+1}(\pi) \geqslant D_j(y).$$

Расписания π и π' удовлетворяют множествам Q, тогда по лемме 3 для любого $i = 0, \ldots, q-1$ выполнено

$$S_{i+1}(\pi) \leqslant S_{i+1}(\pi').$$

Так как для любого $i = 0, \ldots, q-1$ $f_{i+1}(t)$ — монотонно неубывающая функция, то

$$f_{i+1}(S_{i+1}(\pi')) \geqslant f_{i+1}(S_{i+1}(\pi)),$$
$$F_{i+1}(\pi') \geqslant F_{i+1}(\pi) \geqslant D_j(y).$$

Следовательно, при любом расписании $\pi' \in \Phi(N, U, y)$, удовлетворяющем Q, заказ O_j не может быть отправлен поездом с номером больше i. Аналогично рассматривая все заказы множеств $N'_0, N'_1, \ldots, N'_q$, получаем, что расписание π' удовлетворяет Q'. Лемма 4 доказана.

Следствие 1. *Пусть в результате построения расписания $\Theta(N, U, Q, y)$ алгоритмом 3 было получено семейство множеств $Q' = \{N'_0, N'_1, \ldots, N'_q\}$. Тогда любое расписание $\pi' \in \Phi(N, U, y)$, удовлетворяющее множествам Q, удовлетворяет Q'.*

Доказательство следствия 1. Данное утверждение следует из последовательного применения леммы 4 к каждой итерации алгоритма 3.

Теорема 3. *Алгоритм 3 строит расписание $\Theta(N, U, Q, y) \in \Phi(N, U, y)$, имеющее минимальное время обслуживания заказов $C_{\max}$ среди всех расписаний из $\Phi(N, U, y)$, удовлетворяющих Q.*

Если $\Theta(N, U, Q, y) = \emptyset$, то не существует расписания $\pi^* \in \Phi(N, U, y)$, удовлетворяющего Q.

Доказательство теоремы 3. По следствию 1 из леммы 4 любое расписание $\pi' \in \Phi(N, U, y)$, удовлетворяющее семейству множеств Q, будет удовлетворять семейству множеств Q', построенному на последней итерации алгоритма 3. По лемме 3 для любого расписания $\pi' \in \Phi(N, U, y)$, удовлетворяющего Q', для любого $i = 1, \ldots, q$ будет выполнено

$$S_i(\Theta(N, U, Q, y)) \leqslant S_i(\pi').$$

Тогда выполняется неравенство

$$S_q(\Theta(N, U, Q, y)) \leqslant S_q(\pi').$$

Так как $f_q(t)$ — монотонно неубывающая функция, то

$$f_q(S_q(\Theta(N, U, Q, y))) \geqslant f_q(S_q(\pi')),$$
$$F_q(\Theta(N, U, Q, y)) \leqslant F_q(\pi').$$

Для любого расписания π

$$C_{\max}(\pi) = F_q(\pi'),$$

следовательно,

$$C_{\max}(\Theta(N, U, Q, y)) \leqslant C_{\max}(\pi').$$

Первое утверждение теоремы 3 доказано.

Докажем второе утверждение. Пусть алгоритм 3 завершился на m-й итерации и при этом для некоторого $i = 0, 1, \ldots, q$ не было выполнено условие (3)

$$|N_i^m| > K_i.$$

Тогда по следствию 1 из леммы 4 любое расписание $\pi' \in \Phi(N, U, y)$, удовлетворяющее Q, удовлетворяет множеству N_i^m, а значит, нарушает условие (3). Противоречие.

Теорема 3 доказана.

Следствие 2. Если $\Phi(N, U, y) \neq \emptyset$, то

$$C_{\max}(\Theta(N, U, Q^0, y)) = \min_{\pi \in \Phi(N,U,y)} C_{\max}(\pi),$$

где $Q^0 = \{\emptyset, \ldots, \emptyset, N\}$.

Следствие 3. Пусть для любого $i = 0, \ldots, m$

$$wL_{\max}(\Theta(N, U, Q^i, y_i)) = y_{i+1},$$

где $Q^0 = \{\emptyset, \ldots, \emptyset, N\}$ и Q^{i+1} — семейство множеств, полученное при построении расписания $\Theta(N, U, Q^i, y_i)$.

Тогда если для любого $i = 0, \ldots, m$ $\Theta(N, U, Q^i, y_i) \neq \emptyset$, то

$$\Theta(N, U, Q^m, y_m) = \Theta(N, U, Q^0, y_m).$$

7. Алгоритм решения задачи минимизации максимального ВВС выполнения заказов для двух станций

Будем обозначать ВВС заказа O_j при расписании π через $w_j L_j(\pi)$. Максимальное ВВС заказа при расписании π обозначим через $wL_{\max}(\pi)$. При построении оптимального расписания $\pi^*(N, U)$ будем действовать следующим образом.

Алгоритм 4. На первом шаге построим расписание $\pi_1 = \Theta(N, U, Q^0, +\infty)$, где $Q^0 = \{\emptyset, \ldots, \emptyset, N\}$, при котором заказы отправляются по возрастанию моментов поступления, и включим его в множество $\Omega(N, U)$. Тем самым будет построено расписание, удовлетворяющее условию минимума F_i для каждого поезда i, $i = 1, \ldots, q$, а значит, и критерию $C_{\max}$, а в результате работы алгоритма 3 будут построены множества $Q^1 = \{N_0^1, N_1^1, \ldots, N_q^1\}$.

Рассмотрим заказ O_{j_1}, на котором достигается максимум функции ВВС $w_{j_1}L_{j_1}$ при расписании π_1. Пусть этот заказ при данном расписании отправляется поездом m_1, т.е. $O_{j_1} \in T_{m_1}(\pi_1)$. По замечанию 1 в результате работы алгоритма 3 построено расписание с минимальными моментами отправления поездов из всех возможных, удовлетворяющих ограничению $wL_{\max} < y$. Таким образом, для любого расписания π', удовлетворяющего

$$w_{j_1}L_{j_1} \leqslant wL_{\max}(\pi') < y,$$

будет выполнено

$$F_i(\pi') \geqslant F_i(\pi_1).$$

Следовательно, при любом расписании π', удовлетворяющем

$$wL_{\max}(\pi') < wL_{\max}(\pi_1),$$

заказ O_{j_1} должен прибыть на станцию назначения до момента времени $F_i(\pi')$. Это значит, что если множество $\Phi(N, U, w_{j_1}L_{j_1})$ не пустое, то при любом расписании из множества $\Phi(N, U, w_{j_1}L_{j_1})$ заказ O_{j_1} будет отправлен одним из поездов, номер которого меньше m_1.

Построим расписание $\Theta(N, U, Q^1, w_{j_1}L_{j_1})$, при котором максимальное ВВС будет достигнуто на заказе O_{j_2}, отправленном поездом m_2. Включим данное расписание в множество $\Omega(N, U)$. В результате построения расписания $\Theta(N, U, Q^1, w_{j_1}L_{j_1})$ с помощью алгоритма 3 будет построено семейство множеств Q^2.

Следующим шагом будет выполняться построение расписания $\Theta(N, U, Q^2, w_{j_2}L_{j_2})$, при котором заказ O_{j_2} должен быть отправлен поездом с номером меньше m_2, а значит, заказ O_{j_2} попадет в множество N_{m_2-1}. Данное расписание также включим в $\Omega(N, U)$.

Будем повторять эту процедуру до тех пор, пока не наступит такой шаг $s+1$, что расписания $\Theta(N, U, Q^s, w_{j_s}L_{j_s})$ не существует. В этом случае алгоритм 4 возвращает множество $\Omega(N, U)$ и расписание $\pi^*(N, U) = \pi_s$ и завершает работу.

Формальная запись алгоритма 4 выглядит следующим образом.

0. Входные данные:
   $N, U, k_1, \ldots, k_q,\ y = +\infty,\ s = 0.$
1. Для всех $i = 1, \ldots, q-1$ выполняется присвоение $N_i^s := \emptyset$. Также присваиваем $N_q^s := N$.
2. Выполняется построение расписания $\pi_{s+1} = \Theta(N, U, Q^s, y)$ с помощью алгоритма 3. В результате работы алгоритма 3 получено семейство множеств Q^{s+1}.
3. Если $\pi_{s+1} \neq \emptyset$, то:
   а. Включаем π_{s+1} в $\Omega(N, U)$;
   б. Присваиваем $\pi^*(N, U) := \pi_{s+1}$;
   в. Присваиваем $y := wL_{\max}(\pi_{s+1})$, $s := s+1$ и переходим на шаг 2.
4. Если $\pi_{s+1} = \emptyset$, то алгоритм возвращает множество $\Omega(N, U)$, расписание $\pi^*(N, U)$ и завершает работу.

Теорема 4. Множество расписаний $\Omega(N,U)$, полученное в результате выполнения алгоритма 4, является оптимальным по Парето для критериев $wL_{\max}$ и $C_{\max}$. Расписание $\pi^*(N,U)$, полученное в результате выполнения алгоритма 4, оптимально по критерию

$$\min_{\pi\in\Pi(N,U)} \max_{O_j\in N} C_j(\pi) | \min_{\pi\in\Pi(N,U)} \max_{O_j\in N} w_j L_j(\pi).$$

Доказательство теоремы 4. Заметим, что $\Pi(N,U)=\Phi(N,U,+\infty)$. Тогда по следствию 2 из теоремы 3 получаем, что расписание $\Theta(N,U,\emptyset,\ldots$ $\ldots,\emptyset,N,+\infty)$ является оптимальным по критерию $C_{\max}$. На каждом проходе цикла 2–3 алгоритма 4 выполняется построение расписания по алгоритму 3. По теореме 3 имеем:

$$C_{\max}(\pi_1) < C_{\max}(\pi_2) < \ldots < C_{\max}(\pi_s),$$
$$wL_{\max}(\pi_1) > wL_{\max}(\pi_2) > \ldots > wL_{\max}(\pi_s).$$

По следствию 3 из теоремы 3 получаем, что

$$\pi_1 = \Theta(N,U,Q^0,+\infty)$$

и для любого $i=2,\ldots,s$

$$\pi_i = \Theta(N,U,Q^{i-1},wL_{\max}(\pi_{i-1})) = \Theta(N,U,Q^0,wL_{\max}(\pi_{i-1})).$$

Тогда по следствию 2 из теоремы 3

$$C_{\max}(\pi_1) = \min_{\pi\in\Phi(N,U,+\infty)} C_{\max}(\pi)$$

и для любого $i=2,\ldots,s$

$$C_{\max}(\pi_i) = \min_{\pi\in\Phi(N,U,wL_{\max}(\pi_{i-1}))} C_{\max}(\pi).$$

Следовательно, для любого $i=2,\ldots,s$ не существует расписания $\pi'\in$ $\in\Pi(N,U)$, такого что

$$\begin{cases} C_{\max}(\pi_{s-1}) < C_{\max}(\pi') < C_{\max}(\pi_s), \\ wL_{\max}(\pi_{s-1}) > wL_{\max}(\pi') > wL_{\max}(\pi_s). \end{cases}$$

Таким образом, если будет доказано, что расписание $\pi^*(N,U)$ является оптимальным по критерию $wL_{\max}$, то из этого будет следовать, что множество расписаний $\Omega(N,U)$ оптимально по Парето. Пусть расписание $\pi^*(N,U)$ имеет значение целевой функции, равное y^*. Из несуществования расписания $\Theta(N,U,Q^s,y^*)$ по теореме 3 следует, что расписаний, удовлетворяющих построенному семейству множеств Q^s и неравенству $(wL)_{\max} < y^*$, не существует. Последовательно применяя следствие 1 из леммы 4 к построению расписаний $\pi_1,\ldots,\pi_s$, получаем, что не существует расписания $\pi\in\Phi(N,U,y)$, удовлетворяющего Q^0, т.е $\Phi(N, U, y) = \emptyset$. Получаем, что не существует расписания $\pi \in \Pi(N, U)$, для которого выполнено неравенство

$$\max_{O_j \in N} w_j L_j(\pi) < y^*.$$

Следовательно, расписание $\pi^*(N, U)$ оптимально по критерию $wL_{\max}$. Из теоремы 3 также следует, что расписание $\pi^*(N, U)$ имеет наименьшее общее время доставки заказов. Теорема 4 доказана.

8. Оценка трудоемкости алгоритма решения основной задачи

Теорема 5. Трудоемкость алгоритма 4 составляет

$$O(n^2 \max\{n \log n, q \log v\})$$

операций, где n — количество заказов, q — количество поездов, v — количество временных окон, в которые возможно отправление поездов.

Доказательство теоремы 5. Для начала рассмотрим трудоемкость алгоритма 3. На шаге 2 выполняется подбор $k_q, \ldots, k_1$ заказов для формирования составов $T_q, \ldots, T_1$ и вычисление значений $r(T_q(\pi)), \ldots, r(T_1(\pi))$. Следовательно, трудоемкость шага 2 с учетом сортировки заказов по моментам поступления составляет $O(n \log n)$ операций. Вычисление моментов отправления на шаге 3 для q поездов имеет трудоемкость $O(q \log v)$ операций, где v — количество допустимых интервалов отправки в множестве U. Трудоемкость проверки $O(n)$ неравенств на шаге 5 составляет $O(n)$ операций. Вклад в трудоемкость проверки неравенств на шаге 4 будет оценен позже. Таким образом, одна итерация алгоритма 3 (без учета шага 4) имеет трудоемкость $O(\max\{n \log n, q \log v\})$ операций.

На каждой итерации алгоритма 3 происходит добавление заказа в одно или несколько множеств $N_1, N_2, \ldots, N_{q-1}$. Так как алгоритм 3 строит расписание с минимальными моментами отправления из всех, удовлетворяющих ограничению $wL_{\max} < y$, то при уменьшении значения y для существования расписания необходимо, чтобы заказ, на котором было достигнуто максимальное значение ВВС, был отправлен поездом с меньшим порядковым номером. Следовательно, на каждом проходе цикла шагов 2–3 алгоритма 4 происходит включение заказа, на котором достигнута целевая функция, в одно из множеств $N_1, N_2, \ldots, N_{q-1}$. Следовательно, суммарное число итераций алгоритма 3 при проходе цикла 2–3 алгоритма 4 не превышает суммы мощностей множеств $N_1, N_2, \ldots, N_{q-1}$:

$$\sum_{i=1}^{q} |N_i| \leqslant \sum_{i=1}^{q-1} (q-i) k_i < n^2.$$

Из приведенных рассуждений следует, что общая трудоемкость выполнения шагов 1, 2, 3 и 5 алгоритма 3 при выполнении алгоритма 4 составляет $O(n^2 \max\{n \log n, q \log v\})$ операций.

Теперь рассмотрим вклад выполнения шага 4 алгоритма 3 в трудоемкость алгоритма 4. Заметим, что для каждого заказа O_j, $j = 1, \ldots, n$, количество выполнения шагов 4а алгоритма 3 равно $n - N(O_j)$, где $N(O_j)$ — наименьший индекс множества $N_1, \ldots, N_n$, которому принадлежит заказ O_j после завершения работы алгоритма 4. Таким образом, количество шагов 4а алгоритма 3 не превышает суммы мощностей множеств $N_1, N_2, \ldots, N_{q-1}$:

$$\sum_{i=1}^{q} |N_i| \leqslant \sum_{i=1}^{q-1} (q-i)k_i < n^2.$$

Количество обращений к шагу 4а равно $O(1)$, следовательно, общая трудоемкость шагов 4а алгоритма 3 при выполнении алгоритма 4 составляет $O(n^2)$ операций. Количество шагов 4б алгоритма 3 равно количеству расписаний в множестве $\Omega(N, U)$, что не превышает количества итераций алгоритма 3 при проходе цикла 2–3 алгоритма 4. Трудоемкость шага 4б равна $O(1)$. Таким образом вклад шага 4б алгоритма 3 в трудоемкость алгоритма 4 составляет $O(n^2)$ операций.

Таким образом, трудоемкость алгоритма 4 составляет

$$O(n^2 \max\{n \log n, q \log v\})$$

операций. Теорема 5 доказана.

Следствие 4. Мощность множества $\Omega(N, U)$ *не превышает* $\sum\limits_{i=1}^{q-1} (q-i)k_i$.

9. Заключение

В статье рассмотрены две постановки задачи минимизации максимального взвешенного временно́го смещения доставки заказов между двумя станциями на железнодорожном транспорте. Для постановки с фиксированными моментами отправления и прибытия поездов предложен алгоритм трудоемкости $O(qn^2 \log n)$ операций.

Для постановки с неопределенными моментами отправления и прибытия поездов рассмотрена бикритериальная задача минимизации максимального взвешенного временно́го смещения и общего времени доставки заказов между двумя железнодорожными станциями. Представлен алгоритм трудоемкости $O(n^2 \max\{n \log n, q \log v\})$ операций, позволяющий построить множество Парето-оптимальных расписаний. Общее количество расписаний, входящих в Парето-множество, не превышает

$$\sum_{i=1}^{q-1} (q-i)k_i,$$

что меньше n^2. Доказано, что предложенный метод «наполнения множеств» позволяет минимизировать общее время доставки заказов и время отправления каждого из поездов.

СПИСОК ЛИТЕРАТУРЫ

1. *Cordeau J.-F., Toth O., Vigo D.* A Survey of Optimization Models for Train Routing and Scheduling // Transp. Sci. 2000. V. 32 (4). P. 380–396.
2. *de Oliveira E.S.* Solving Single Track Railway Scheduling Problem Using Constraint Programming // Univer. of Leeds. School of Computing. PhD Thesis. 2001.
3. *Brucker P., Heitmann S., Knust S.* Scheduling Railway Traffic at a Construction Site // OR Spectrum 2002. 24 (1). P. 19–30.
4. *Sourd F.* Application of Scheduling Theory to Solve Real-life Railway Problems // Proc. Conf. on Optimization and Practices in Industry (COPI'11). Paris. 2011.
5. *Gafarov E., Dolgui A., Lazarev A.* Two-Station Single-Track Railway Scheduling Problem With Trains of Equal Speed // Comput. Ind. Eng. 2015. V. 85. P. 260–267.
6. *Disser Y., Klimm M., Lubbecke E.* Bidirectional Scheduling on a Path // Matheon Preprint 1060, Res. Cent. Matheon, Tech. Univ. Berlin. 2014. 32 p.
7. *Лазарев А.А., Мусатова Е.Г., Архипов Д.И.* Задача минимизации максимального взвешенного временного смещения выполнения заказа для двух станций // Тр. 3-й Всеросс. конф. с междунар. участием «Технические и программные средства систем управления, контроля и измерени» (УКИ-2012, Москва). М.: ИПУ РАН, 2012. С. 1962–1967.
8. *Arkhipov D., Lazarev A.* The Problem of Minimization Maximum Weighted Lateness of Orders for Two Railway Stations // EURO 2012 – Vilnius, 2012. P. 151.
9. *Архипов Д.И., Лазарев А.А.* Минимизация максимального взвешенного временного смещения для заказов на доставку грузов между двумя станциями в условиях ограниченного движения составов // Тр. 3-й науч.-техн. конф. с междунар. участием «Интеллектуальные системы управления на железнодорожном транспорте – ИСУЖТ-2014» (Москва, 2014). М.: ОАО «НИИАС», 2014. С. 7–10.
10. *Graham R.L., Lawler E.L., Lenstra J.K., et al.* Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey // Ann. Discret. Math. 1979. V. 5. P. 287–326.
11. *Benson H.P.* An Outer Approximation Algorithm for Generating all Efficient Extreme Points in the Outcome Set of a Multiple Objective Linear Programming Problem // J. Glob. Optim. 1998. V. 13. P. 1–24.
12. *Лазарев А.А.* Теория расписаний. Оценки абсолютной погрешности и схема приближенного решения задач теории расписаний. М.: Изд-во МФТИ, 2008.

Статья представлена к публикации членом редколлегии А.И. Кибзуном.

Поступила в редакцию 04.02.2016