

УДК 519.854.2

Рецензенты: д.т.н., профессор В.Н. Бурков;  
д.т.н., профессор Д.И. Коган

Под редакцией академика РАН С.Н. Васильева

Данное учебное пособие посвящено задачам теории расписаний, возникающим на транспорте. Представлены основы теории расписаний, а также способы построения моделей и методы решения задач управления транспортными системами.

Изложенный материал предназначен для студентов и преподавателей вузов математических специальностей, специалистов в области управления и практиков, занимающихся решением задач планирования грузовых перевозок.

© Физический факультет МГУ, 2012 г.

© Лазарев А.А., Мусатова Е.Г., Кварацхелия А.Г., Гафаров Е.Р.

# Оглавление

<b>Предисловие редактора</b>	<b>5</b>
Введение . . . . .	8
<b>1 Общие сведения о теории расписаний и задачах транспортного планирования</b>	<b>10</b>
1.1 Предмет теории расписаний . . . . .	10
1.2 Классификация задач <b>ТР</b> . . . . .	22
1.3 Задачи управления транспортными системами . . . . .	37
1.4 Классификация задач календарного планирования, возникающих на железнодорожном транспорте . . . . .	41
1.5 Обзор публикаций по задачам железнодорожного планирования . . . . .	44
<b>2 Модели задач железнодорожного планирования</b>	<b>57</b>
2.1 Задача формирования составов и расписания движения грузовых поездов . . . . .	57
2.2 Целочисленная постановка задачи формирования составов и расписания их движения . . . . .	66
2.3 Модели задач железнодорожного планирования, использующие пространственно-временные графы . . . . .	76
2.4 Задача формирования грузовых потоков . . . . .	89

2.5	Задача оперативного управления движением составов . . .	92
<b>3</b>	<b>Алгоритмы решения задач железнодорожного планирования</b>	<b>97</b>
3.1	Решение задачи минимизации среднего времени выполнения заказов для двух станций с одним локомотивом . . . .	97
3.2	Задача составления расписания движения поездов между двумя станциями, соединенными однопутной железной дорогой . . . . .	104
3.3	Задача минимизации максимального взвешенного временного смещения выполнения заказа для двух станций . . . .	120
3.4	Полиномиальные алгоритмы задач о станциях на основе сведения к задачам теории расписания для одного прибора	130
3.5	Приближённые методы решения задач железнодорожного планирования . . . . .	136
	Список литературы . . . . .	152

# Предисловие редактора

Уважаемый читатель!

Учебное пособие, которое Вы держите в руках, написано на кафедре физико-математических методов управления МГУ, созданной в 2009 году на физическом факультете университета. На кафедре преподаются методы решения разнообразных актуальных задач управления системами физико-технической, организационно-экономической и другой природы.

Решение задач планирования и управления стало особенно актуальным в 20-м веке. Именно в это время формируются новая область математики — исследование операций, а также смежные с ней дисциплины — теория массового обслуживания, теория расписаний, теория автоматического управления, теория оптимального управления, теория многокритериального принятия решений.

Разработка расписаний — одна из самых популярных с теоретической и практической точек зрения область исследования операций. Задачи теории расписаний связаны с упорядочиванием некоторых работ (операций) по времени и/или по исполнителям (приборам).

Эти задачи широко востребованы в технических, организационных и других системах, в том числе и в физике (например, при планировании экспериментов), а этот учебник посвящен задачам теории расписаний, возникающим на транспорте. В случае транспортных систем возникает необходимость составить расписание движения транспортных средств по участкам некоторого пути. В данном случае в роли т.н. “приборов” выступают пути с ограниченной пропускной способностью, а “операци-

ями” являются транспортные средства, перевозящие конкретный груз. При этом необходимо учитывать ограничения на очередность движения транспорта, ограничения, связанные с подвижным составом, и т.п. Цель решения таких задач — построение допустимых расписаний, при котором все ограничения соблюдены или, что является более сложным, — нахождение оптимального допустимого расписания по тому или иному критерию оптимальности, например, по критериям быстродействия (т.е. с минимизацией общего времени выполнения всех работ), минимума финансовых потерь от запаздывания и т.п.

Решение задач теории расписаний усложняется тем фактом, что большинство из них являются  $NP$ -трудными, т.е. алгоритмы их решения, реализованные на ЭВМ, могут требовать неприемлемо большое время работы для решения практических задач “большой размерности”.

Очевидно, что задачи управления транспортными системами могут сильно различаться между собой в зависимости от вида транспорта. В данном пособии акцент сделан на задачи с жесткими ограничениями на пропускную способность путей и транспортные средства. Типичным представителем систем с подобными характеристиками является железнодорожный транспорт, хотя подобные ограничения могут возникать и применительно к другому транспорту.

В этом учебнике излагаются предмет исследования теории расписаний и история ее становления как отдельной дисциплины. Приводятся основная терминология и классификация задач, применяемые в рамках теории расписаний, а также классификация задач, возникающих на транспорте. В качестве типовых задач построения расписаний рассматриваются: задача о маршрутизации и календарном планировании перевозки заказов, задачи с ограничениями на подвижной состав, задачи построения расписания (очередности) движения составов на участках с низкой пропускной способностью и некоторые другие.

На примере приведенных транспортных задач можно представить,

как знания из математической дисциплины "Теория расписаний" используются на практике.

Книга рекомендуется как дополнительный учебник к читаемому на кафедре курсу "Теория расписаний".

Заведующий кафедрой  
физико-математических методов управления МГУ,  
академик С.Н. Васильев.  
Июнь 2012

# Введение

Данное пособие посвящено моделям и методам решения задач управления транспортными системами и связи транспортных задач с теорией расписаний. Основной акцент сделан на задачах железнодорожного планирования и построения расписаний движения грузовых поездов.

Первая глава содержит необходимые сведения из теории расписаний, а также обзор подходов к решению задач на железнодорожной сети.

Во второй главе представлены различные модели и задачи железнодорожного планирования: задачи формирования составов, маршрутов следования поездов, построения графиков их движения, формирования грузовых потоков через сортировочные станции, а также задачи оперативного управления движением составов на узле железной дороги. Рассмотрены различные варианты и постановки задач в зависимости от целевых функций (минимизация издержек, суммарного запаздывания, среднего времени заказа в пути, времени окончания работ, максимизация пропускной способности узла и т.д.), а также частные случаи этих задач с фиксированными маршрутами или расписаниями движения поездов.

Третья глава посвящена методам решения задач железнодорожного планирования. На частных случаях моделей, представленных во второй главе, демонстрируется применение как специальных алгоритмов решения транспортных задач, так и известных алгоритмов теории расписаний, адаптированных к специфике железнодорожной тематики. Кроме того, представлен подход к решению задач транспортной логистики на основе введения специальной метрики, который позволяет находить приближенное решение задачи с заранее известной погрешностью.

А.А. Лазарев, Е.Г. Мусатова, А.Г. Кварацхелия, Е.Р. Гафаров.

Июнь 2012 г.

## **Благодарности**

Считаем своим долгом выразить благодарность: С.Н. Васильеву, В.Н. Буркову, Д.И. Когану, Р.Р. Садыкову, П.С. Кореневу, Д.И. Архипову, А.В. Баранову, А.А. Карпычеву, М.В. Ласковой, Н.Ф. Хуснуллину, А.В. Лобову, Ф. Вернеру (Германия), Ф. Баптисте (Франция) за помощь в работе и обсуждение полученных результатов.

Авторы выражают благодарность Е.Г. Лазаревой и Т.С. Ефимовой за большую помощь в подготовке и оформлении материалов книги и постоянную поддержку.

Работа выполнена при частичной финансовой поддержке научных программ 15 и 29 РАН, а также фондов: РФФИ (Россия), DAAD (Германия), CNRS (Франция).



# Глава 1

## Общие сведения о теории расписаний и задачах транспортного планирования

### 1.1 Предмет теории расписаний

Люди на протяжении всей своей жизни сталкиваются с необходимостью составления расписаний. Каждый из нас занимается планированием личного времени. Мы знаем **что** и к **какому сроку** нужно выполнить, а также имеем представление, сколько времени потребуется на каждое дело. Исходя из этих данных, мы составляем расписание, упорядочивая наши дела по времени. Чаще всего составление личного расписания не вызывает сложности. Практически все люди при этом руководствуются “похожими алгоритмами”, стремясь сделать **все** дела и **вовремя**.

Часто мы планируем наши действия в порядке возрастания крайних сроков исполнения работ (дел). Например, студенты во время экзаменационной сессии учат предмет с наименьшим директивным сроком (ближайший по дате сдачи экзамен). Если первый экзамен необходимо сдать 12-го января, второй 15-го, а третий 19-го, при этом на каждый экзамен необходимо 3 дня, то большинство студентов составят такое расписание: “с 9-го по 11-ое я готовлюсь к первому экзамену, с 12-го по 14-ое ко вто-

рому и с 16-го по 18-ое к третьему”.

Сложности при составлении расписаний появляются тогда, когда работ становится много, нужно учесть множество дополнительных условий и/или составить расписание не для одного человека, а для целого коллектива. Представьте себе сотни работ и десятки исполнителей, для которых необходимо составить расписание.

В процессе решения таких задач, были выработаны общие рекомендации, принципы и методики составления расписаний. В последствии подобные задачи стали исследоваться в рамках специального раздела науки — **Теории Расписаний** (далее **ТР**).

Эта наука появилась не на пустом месте, а возникла на стыке других областей научного знания. Прежде чем дать определение, что такое **Теория Расписаний**, опишем одну важную область математики.

---

**Исследование операций (ИО)** – научный метод выработки количественно обоснованных рекомендаций по принятию решений. Важность количественного фактора в ИО и целенаправленность сформулированных рекомендаций позволяют определить ИО как теорию принятия оптимальных решений. ИО способствует превращению искусства принятия решений в математическую дисциплину. Термин “ИО” возник в результате буквального перевода выражения “operation research”, введенного в конце 30-х годов 20-го века как условное наименование одного из подразделений британских ВВС, занимающегося вопросами использования радиолокационных установок в общей системе обороны. Первоначально ИО было связано с решением задач военного содержания, но уже с конца 40-х годов прошлого века оно используется для решения технических, технико-экономических задач, а также задач управления на различных уровнях.

---

Теперь мы можем дать определение **Теории Расписаний**.

---

**Теория расписаний** – это раздел исследования операций, в котором строятся и анализируются математические модели<sup>1</sup> календарного планирования (т.е. упорядочивания во времени) различных целенаправленных действий с учетом целевой функции и различных ограничений.

---

Задачи составления расписаний возникают в частности:

- на производстве, когда нужно упорядочить отдельные операции по исполнителям (цеха, станки) и по времени;
- на транспорте при составлении расписания движения поездов, самолетов, общественного городского транспорта;
- при планировании занятий в учебных заведениях;
- при планировании занятости персонала, например, дежурства врачей;
- при выполнении сложных продолжительных проектов строительства зданий, кораблей и т.п.;
- при планировании проведения спортивных мероприятий;
- в компьютерных сетях при планировании очередности передачи пакетов информации и т.д.

Содержательно многие задачи **ТР** являются оптимизационными, т.е. состоят в выборе (нахождении) среди множества допустимых расписаний (расписаний, допускаемых условиями задачи) тех решений, на которых достигается “оптимальное” значение целевой функции. Обычно под “оптимальностью” понимается минимальное или максимальное значение некоторой целевой функции. *Допустимость* расписания понимается в

---

<sup>1</sup>Построить математическую модель – значит описать с использованием математического аппарата изучаемые процесс или явление.

смысле его осуществимости, а *оптимальность* — в смысле его целесообразности.

---

**Пример.** *Необходимо построить дом как можно быстрее, при этом последовательность работ должна быть соблюдена. Для данной задачи допустимое расписание то, при котором будет соблюдена последовательность работ, а оптимальное расписание – это допустимое расписание, при котором дом будет построен в минимальные сроки и (или) с минимальными затратами.*

---

Другой тип задач заключается в поиске допустимого расписания, удовлетворяющего всем условиям. Приведем примеры обоих типов задач.

### Задача нахождения допустимого расписания

На одном процессоре требуется выполнить множество  $N = \{1, 2, \dots, n\}$  заданий. Для каждого задания  $j \in N$  определены длительность выполнения  $p_j > 0$ , время поступления задания на процессор  $r_j \geq 0$  и крайний директивный срок  $D_j > 0$ , к которому задание должно быть выполнено. Процессор готов к выполнению заданий с момента времени 0 и может выполнять одновременно только одно задание. Прерывания при выполнении любого задания запрещены. Необходимо построить допустимое расписание выполнения заданий, при котором все условия задачи соблюдены. То есть необходимо для каждого требования  $j \in N$  определить момент начала выполнения  $S_j$  такой, что  $S_j \geq r_j$  и момент окончания выполнения  $C_j = S_j + p_j \leq D_j$ . Причем если  $S_j < S_i$ , то  $S_j + p_j \leq S_i$ , где  $S_i$  – момент начала выполнения другого задания  $i \in N, i \neq j$ , т.к. процессор не может выполнять более одного задания в каждый момент времени.

В задачах **ТР** время задается в условных единицах. Параметры  $p_j, r_j, D_j, C_j, S_j$  могут измеряться в минутах, часах, днях и т.п. С точки зрения вычислительного процесса два примера –  $p_1 = 2$  минуты,  $p_2 = 3$  минуты,  $r_1 = 0$  – нулевая минута,  $r_2 = 1$  – первая минута и, пример, где  $p_1 = 2$  часа,  $p_2 = 3$  часа,  $r_1 = 0$  – нулевой час,  $r_2 = 1$  – первый

час, – являются идентичными, поэтому единицу измерения времени при определении задач опускают.

### Задача нахождения оптимального расписания

Модифицируем предыдущую задачу следующим образом. Пусть крайние сроки  $D_j$  не заданы (т.е.  $D_j = +\infty$ ,  $j = 1, 2, \dots, n$ ). Все остальные условия остаются теми же. Обозначим  $C_j$  – момент окончания выполнения задания  $j$ , т.е.  $C_j = S_j + p_j$ , где  $S_j$  – момент начала выполнения задания  $j$ . Необходимо построить допустимое расписание, при котором значение функции  $\sum_{j=1}^n C_j$  будет минимальным.

Теперь с оглядкой на две приведенные задачи можно дать следующие два определения.

**Определение 1.** Задача, в которой все входные данные полностью определены, называется **индивидуальной задачей**.

**Определение 2.** **Массовая задача** – бесконечное множество индивидуальных задач.

В дальнейшем мы будем называть массовую задачу просто *задачей* (например, “Задача коммивояжера”), а индивидуальную задачу – *примером*.

В таблице 1.1 представлены допустимое и оптимальное расписания для обеих задач на одном и том же примере. Схематично полученные расписания изображены на рисунке 1.1. На этом рисунке расписания представлены в виде набора прямоугольников, расположенных вдоль виртуальной временной оси  $t$ . Каждый прямоугольник соответствует некоторому заданию, а длина прямоугольников соответствует продолжительности выполнения заданий, а их расположение на оси указывает время начала и окончания выполнения каждого задания.

Для первой задачи существует единственное допустимое расписание, при котором порядок выполнения заданий определен следующим образом: (1, 3, 2, 4). Сначала выполняется первое задание, потом третье,

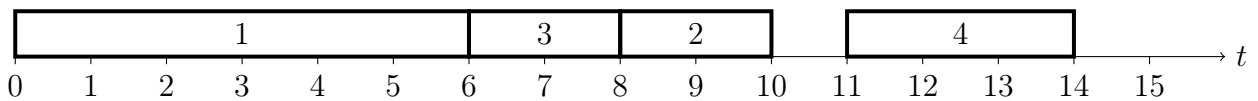
**Таблица 1.1. Пример. Исходные данные, допустимое и оптимальное расписания**

$j$	1	2	3	4
$p_j$	6	2	2	3
$r_j$	0	1	2	11
$D_j$	7	10	9	15
Допустимое расписание. Время начала выполнения $S_j$	0	8	6	11
Допустимое расписание. Время окончания выполнения $C_j$	6	10	8	14
Оптимальное расписание. Время начала выполнения $S_j$	5	1	3	11
Оптимальное расписание. Время окончания выполнения $C_j$	11	3	5	14

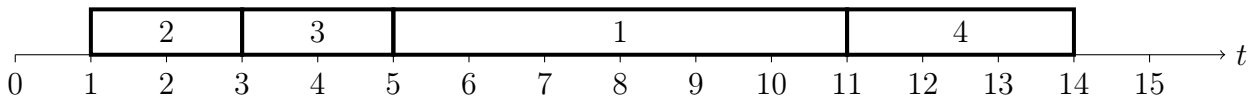
второе и четвертое. Очевидно, что для данного примера любой другой порядок выполнения заданий недопустим. Например, при порядке обслуживания  $(2, 1, 3, 4)$  будет нарушено условие  $C_1 \leq D_1$ , где  $D_1 = 7$  и  $C_1 = 9$ .

Для второй задачи схематично представлено оптимальное расписание. На этом же рисунке для второй задачи изображено допустимое (в терминах второй задачи), но неоптимальное расписание, заданное порядком  $(1, 2, 3, 4)$ . Легко убедиться, что при оптимальном расписании, заданном порядком обслуживания  $(2, 3, 1, 4)$  имеем  $\sum_{j=1}^4 C_j = 11 + 3 + 5 + 14 = 33$ , а при расписании, заданном как  $(1, 2, 3, 4)$ , имеем  $\sum_{j=1}^4 C_j = 6 + 8 + 10 + 14 = 38$ .

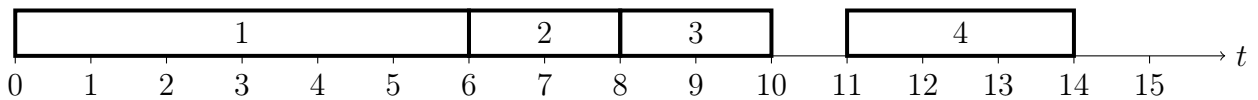
Допустимое расписание для первой задачи:



Оптимальное расписание для второй задачи:



Неоптимальное расписание для второй задачи:



**Рис. 1.1. Пример. Графическое представление расписаний.**

Задачи **ТР** (как задачи раздела Исследования Операций) обладают рядом черт, обуславливающих методику их составления и решения. Во-первых, даже для простых параметрических задач не удастся представить решения в виде аналитического выражения от соответствующих параметров (в виде формулы). Поэтому задачи **ТР**, в подавляющем большинстве, не поддаются аналитическому решению и должны решаться численно. Во-вторых, большинство задач **ТР** содержит в своих формулировках большое количество числового материала, не сводящегося к аналитическим выражениям. Поэтому численное решение этих задач, за немногими исключениями, возможно лишь с помощью компьютера.

Для решения задач **ТР** необходимо разработать *алгоритм решения*. То есть последовательность действий, выполняемых компьютером, с помощью которых можно построить искомое расписание (допустимое или оптимальное).

---

**Алгоритм решения задачи – это последовательность действий, с помощью которых можно построить искомое решение для любого примера задачи.**

---

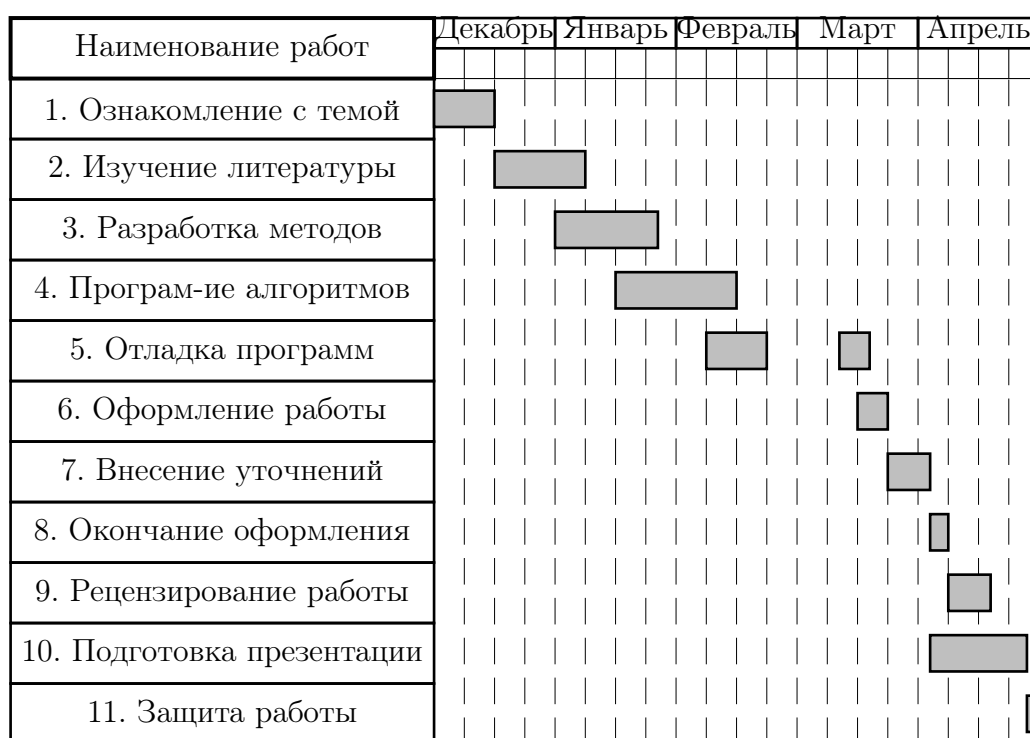
Встречающиеся на практике задачи составления расписаний содержат тысячи, а порой и миллионы заданий. Поэтому основная цель при исследовании моделей (задач) **ТР** – это **построение эффективных, т.е. быстрых, алгоритмов решения**. Решение примера должно быть получено за “разумное” время.

### **1.1.1 Возникновение и этапы развития теории расписаний**

До 20-го века решение **задач ТР** не требовало много времени или большого числа вычислений. В начале прошлого века развитие науки и техники ускорилося. Увеличился темп и обыденной жизни. Задачи составления расписаний оперировали все большим количеством взаимосвязанных работ, и упорядочить их во времени становилось все труднее.

К началу 20-го века относятся два показательных примера. В период

с 1903 по 1919 гг. американский ученый Генри Гантт публикует ряд научных работ и предлагает новый способ представления расписаний, получивший название “Диаграмма Гантта”. Гантт занимался исследованием и улучшением руководства (менеджмента) на промышленных предприятиях (например, в компании по производству хлопчатобумажных тканей и на предприятиях по постройке кораблей). Диаграмма Гантта – это схематичное изображение календарного плана. В ней работы представлены в виде прямоугольников, размещенных вдоль оси времени. Длина прямоугольника соответствует времени, необходимому на выполнение соответствующей работы. На диаграмме указаны как продолжительность работы так и время ее начала и окончания. На рис. 1.2 представлен пример такой диаграммы<sup>2</sup>.



**Рис. 1.2. Диаграмма Гантта**

Другой показательный пример – изобретение конвейера Генри Фордом. В 1908-м году он организует поточное производство на своем автомобильном заводе по “конвейерному типу”. В данной интерпретации

<sup>2</sup>На диаграмме представлена хронология подготовки студентом дипломной работы.



конвейер – это способ организации производства каких-либо изделий, при которой:

- процесс производства разделяется на отдельные операции (стадии);
- одновременно в производстве находится несколько изделий, находящиеся на различных стадиях.

Операции при таком способе производства упорядочены. То есть существует очередность операций и нарушать ее нельзя. Например, не имеет смысла прикручивать колеса или вставлять стекла, пока не собрана рама. Еще одна важная особенность такого производства – выполнение независимых операций параллельно. То есть пока собирается кузов, параллельно на другой линии можно собирать двигатель.

Конвейерное производство позволило Форду в 1,5 раза сократить время выпуска автомобилей. Теперь конвейер Форда можно встретить на многих массовых производствах.

Еще один пример уже из советской истории. В 60–70-е годы 20-го века на Новочеркасском электровозостроительном заводе, а также при создании автоматизированной системы управления на Львовском телевизорном заводе, была использована следующая методика управления. В основе ее лежит простая идея, известная в отечественной литературе как система “управления работой на склад”: восстановление запаса деталей, созданного на складе, по мере расходования деталей на сборке – разновидность системы регулирования запасов. “Обнуление” запаса интерпретируется как “точка заказа”, а “партия заказа”, т.е. решение, сколько деталей следует произвести для восстановления запаса, фиксировалось в специальной карточке, которую брал как сменное задание рабочий из ячейки, соответствующей дате рабочего дня. Так создавался необходимый запас деталей к необходимому сроку (т.е. “Вовремя”) и упрощалось оперативное управление в массовом производстве дискретного типа на предприятиях, взявших за образец новочеркасскую систему. Картотека заданий расставлялась по ячейкам в расчете на начальный запас,

соответствующий потребности сборки в определенный стандартный интервал времени, например месяц. Впоследствии идеи этих двух систем были полностью повторены на японских автомобильных заводах. Теперь всему миру известны “японские” методы “Канбан” (яп. “карточка”) и “Just-in-Time” (“Вовремя”).

К середине 20-го века выделились еще две области знаний, некоторые задачи которых впоследствии стали частью **ТР**. Это *сетевое планирование* и *теория массового обслуживания*. Чтобы описать эти две теории, дадим несколько определений.

---

**Сетевая модель – информационно-аналитическая модель реализации некоторого комплекса взаимосвязанных работ, рассматриваемая как ориентированный граф без контуров, отображающий естественный порядок выполнения этих работ во времени.**

---

Сетевая модель может содержать некоторые другие характеристики (например, время, стоимость, ресурсы), относящиеся к отдельным работам или комплексу в целом.

---

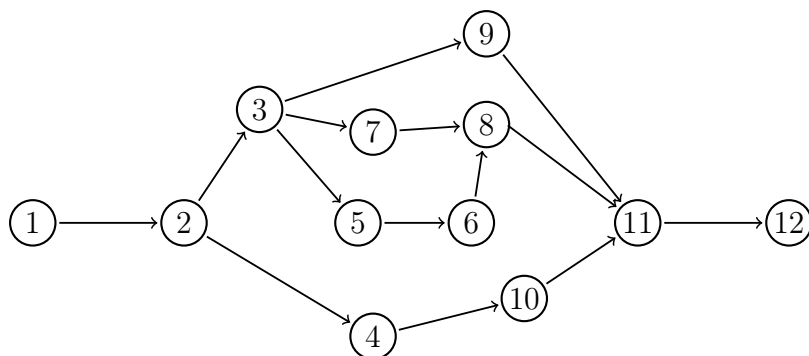
**Сетевое планирование – совокупность методов анализа, планирования и управления, использующих сетевую модель как основную форму представления информации об управляемом комплексе работ.**

---

Использование сетевого планирования позволяет повысить качество планирования и управления при реализации комплекса работ. Например, дает возможность четко координировать деятельность всех сторон (исполнителей), участвующих в реализации, выделять наиболее важные задачи, определять сроки реализации и т.д.

Методы сетевого планирования являются основой специальных компьютерных программ – систем сетевого планирования и управления, таких как Microsoft Project, Primavera и т.д.

В 1958-м году по заказу Министерства Обороны США для проек-



**Рис. 1.3. Сетевой график**

та создания ракетной системы “Поларис” был разработана специальная техника анализа и оценки комплексов взаимосвязанных работ. Проект “Поларис” был ответом на кризис, наступивший после запуска Советским Союзом первого космического спутника. Созданная техника получила название PERT (Project Evaluation and Review Technique). Двумя ее основными элементами было – наглядное представление комплекса работ на бумаге в виде т.н. сетевого графика (см. сетевая модель) и метод расчета критического пути по этому графику. Позднее мы подробнее расскажем о сетевом графике как о способе представления комплекса работ на бумаге (плоскости). Пример сетевого графика представлен на рис. 1.3. Этот сетевой график соответствует проекту подготовки дипломной работы, изображенному на Диаграмме Гантта на рис. 1.2.

В настоящее время сетевое планирование – это раздел **ТР**, в котором рассматриваются задачи составления расписаний для комплексов взаимосвязанных работ, например, при строительстве здания, сборке корабля или самолета, при выполнении крупных проектов (строительство БАМ) и т.п.

Теперь несколько слов об одной смежной области **ТР**. Создание систем телефонной связи и необходимость расчета их пропускной способности послужили в 20-х годах 20-го века стимулом развития *Теории массового обслуживания* (ТМО). В ТМО изучаются потоки требований на обслуживание (заданий), поступающих в системы обслуживания и вы-

ходящих из них, длительность ожидания, длины очередей и т.д. Основная цель исследований в ТМО – рационализация системы обслуживания. Яркий пример задачи ТМО – планирование расписания работы кассиров в продуктовом супермаркете. Если нам известно, как меняется количество покупателей в зависимости от времени суток (вечером покупателей обычно больше), и мы хотим добиться того, чтобы покупатели не стояли в очереди слишком долго, но при этом и кассиры не “простаивали”, то мы должны спланировать количество кассиров в каждое время оптимальным образом. На стыке двух областей знаний – ТМО и **ТР** – появились новые задачи, получившие названия “стохастические задачи” **ТР**.

Итак, в первой половине 2-го века был сформулирован ряд практических и теоретических задач составления расписаний. В 1956-м году Ричард Беллман [17] предложил термин “Теория расписаний” для обозначения совокупности данных задач и относящихся к ним научных знаний. В 1967-м году публикуется монография Конвея, Максвелла и Миллера “Теория расписаний”. В 1975-м году перевод этой книги на русский язык синхронно выходит с книгой советских авторов В.С. Танаева и В.В. Шкурбы “Введение в теорию расписаний”. С этих пор можно считать **ТР** сформировавшейся теорией.

В 70-х годах 20-го века (после выхода работ о теории сложности решения оптимизационных задач) акцент в исследовании задач **ТР** сместился. Теперь при изучении задач **ТР** исследователь не только строит эффективные алгоритмы решения, но и ищет ответ на вопрос – насколько сложна задача в терминах  $NP$ -трудности и полиномиальной разрешимости, и какой быстроты алгоритм “в лучшем случае” можно для этой задачи построить.

### 1.1.2 Способы представления расписаний

Ранее мы уже познакомились с некоторыми способами представления расписаний. Обозначим главные из них.

- **Табличное представление.** В таблице представлены промежутки времени, в которые выполняются задания, а также их исполнители (номер станка, процессор и т.п.). Пример расписания в виде таблицы представлен в первой главе;
- **Графическое представление.** Например, с помощью Диаграммы Ганта.
- Для некоторых задач **ТР** возможно **векторное (перестановочное) представление** расписания. При этом указывается лишь порядок выполнения заданий, например (2, 3, 4, 1). Пример такого порядка приводится в первой главе.

## 1.2 Классификация задач ТР

Приведем некоторые способы классификации задач **ТР**, а затем подробнее расскажем о некоторых из этих задач.

**Способы классификации задач ТР :**

- **По типу искомого решения:**
  - *Задачи упорядочивания.* В этих задачах уже задано распределение работ по исполнителям, а также определены все параметры работ (продолжительность выполнения, время поступления и т.д.). Необходимо составить расписание (или порядок) выполнения работ каждым исполнителем;
  - *Задачи согласования.* Основное внимание в этих задачах уделяется выбору продолжительности выполнения работ, времени поступления и другим параметрам;

- *Задачи распределения* подразумевают поиск оптимального распределения работ по исполнителям.

• По типу целевой функции:

- *Задачи с суммарными критериями оптимизации.* В предыдущей главе мы привели пример такой задачи, в которой необходимо было минимизировать *суммарное* значение моментов окончания обслуживания работ  $\sum_{j=1}^n C_j$ ;
- *Задачи с  $\min\max$  (минимаксными) критериями оптимизации.* Отличие этих задач от задач с суммарными критериями заключается в том, что нужно минимизировать не сумму некоторых значений, а лишь максимальное из них. Например, если в упомянутой задаче необходимо *минимизировать максимальное* значение  $C_{\max}$ , где  $C_{\max} = \max_{j \in N} C_j$ , то мы получим одну из тривиальных задач этого класса;
- *Многокритериальные задачи оптимизации.* Если в исследуемых задачах необходимо построить оптимальное решение с точки зрения нескольких целевых установок (функций), то такие задачи называются многокритериальными. Например, если в упомянутой задаче необходимо не только минимизировать значение  $\sum_{j=1}^n C_j$ , но минимизировать и время простоя процессора (прибора), то это многокритериальная задача<sup>3</sup>;
- *Задачи на построение допустимого расписания.* В предыдущем разделе был дан пример такой задачи. Необходимо отметить, что данный класс задач можно свести к оптимизационным задачам, введя специальную функцию штрафа, который нужно минимизировать. Тем не менее, принято выделять такие задачи в отдельный класс.

---

<sup>3</sup>Некоторые многокритериальные задачи могут быть преобразованы (сведены) к однокритериальным. Другие же могут подразумевать нахождение “Парето-оптимального решения” или нахождение оптимума иерархической целевой функции.

- По способу задания входной информации:

- *Детерминированные задачи (off-line)*. Для таких задач характерно, что все входные данные задачи точно известны, т.е. даны значения всех параметров до начала ее решения;
- *Динамические задачи (on-line)*. Для данных задач расписания строятся в режиме реального времени, т.е. перед началом решения задачи мы не знаем значения всех параметров. Расписание строится по частям по мере поступления новой информации. При этом в любой момент может быть понадобится ответ о качестве построенного “частичного” расписания.

- По разделу ТР. В рамках ТР принято выделять следующие разделы:

- *Сетевое планирование или построение расписания для проекта, Project scheduling (PS)*;
- *Календарное планирование или построение расписания для приборов, Machine scheduling (MS)*;
- *Составление временных таблиц (Time Tabling)*;
- *Доставка товаров в магазины (Shop-Floor Scheduling)*;
- *Составление расписаний движения транспортных средств (Transport Scheduling), Циклические расписания для транспортных средств (Vehicle Routing)*;
- *Составление расписаний спортивных мероприятий (Sports scheduling)*.

Приведенные классификации задач ТР условны и лишь указывают на некоторые характерные особенности решаемых задач.

### 1.2.1 Дополнительные условия в задачах ТР

---

В первой главе мы использовали слово “задание” для обозначение тех действий, которые нужно упорядочить во времени. В дальнейшем мы будем использовать другие, общепринятые в ТР, слова обозначающие задания: *Требование* или *Работа*.

---

В задачах ТР могут быть заданы дополнительные ограничения на параметры требований, порядок обслуживания требований, на приборы. Перечислим некоторые обозначения, которые потребуются нам в дальнейшем. Параметры требований обозначаются следующим образом:

$r_j$  – *момент поступления требования на обслуживание*. Данный параметр определяет момент времени, начиная с которого требование может быть поставлено на обслуживание, но не обязательно его обслуживание начнется в этот момент;

$p_j$  – *продолжительность обслуживания требования*. Параметр определяет время, которое необходимо для обслуживания требования;

$d_j$  – *директивный срок завершения обслуживания*. Данный параметр определяет момент времени, к которому желательно завершить обслуживание требования. Необходимо различать желательный и предельный моменты завершения обслуживания (англ. – due date  $d_j$  и deadline  $D_j$ , соответственно). Желательный момент завершения обслуживания можно нарушать, хотя при этом накладывается штраф, который влияет на значение целевой функции задачи;

$D_j$  – *предельный срок завершения обслуживания*. Предельный срок завершения нарушать нельзя, и любое расписание, в котором есть завершающееся после своего предельного момента требование, является недопустимым. Примером директивных сроков  $d_j$  (due date) и  $D_j$  (deadline) могут служить: момент окончания ужина (который можно и нарушить) и день проведения экзамена (который нарушать крайне нежелательно);



$w_j$  – вес требования. Данный параметр характеризует “важность”, “значимость” требования и учитывается в целевой функции задачи при подсчете штрафа, который накладывается на обслуживание требования.

Существует специальная система кодирования задач **ТР**, с которой мы познакомимся позднее. В этой системе можно встретить следующие обозначения:

*pmtn* – данная запись означает, что допустимы прерывания в обслуживании требований. То есть можно прервать обслуживание требования, обслужить другое требование и после продолжить обслуживание прерванного;

*prec* – означает, что между требованиями заданы отношения предшествования. Эта же запись может выглядеть как *tree*, *out – tree*, *in – tree*, *chain*, которые означают, что граф отношений предшествования имеет вид дерева или цепочки;

*batch* – свидетельствует о том, что рассматривается задача batching, когда требования объединены в группы.

### 1.2.2 Целевые функции в задачах ТР

Обычно задача теории расписаний характеризуется целевой функцией (критерием оптимальности), которую необходимо минимизировать (реже, максимизировать) на множестве допустимых расписаний. Целевая функция в задачах **ТР** вычисляется на основе некоторого набора штрафов (штрафных функций), которые возникают при фиксации порядка обслуживания требований в расписании.

В теории расписаний различают следующие основные типы штрафных функций:

- $C_j$  – *момент завершения*, равный моменту окончания обслуживания требования  $j$ ;
- $L_j$  – *временное смещение*, равное величине  $C_j - d_j$ ;
- $T_j$  – *запаздывание*, равное величине  $\max\{0, C_j - d_j\}$ ;
- $E_j$  – *опережение*, равное величине  $\max\{0, d_j - C_j\}$ ;
- $U_j$  – *требование запаздывает*, равно 0, если  $C_j \leq d_j$ , и 1, – в противном случае.

В задачах, когда задан *вес* требования  $w_j$ , указанные выше критерии называются *взвешенными*, а их значение вычисляется путем умножения исходного значения на коэффициент  $w_j$ . Например, *взвешенное временное запаздывание*  $w_j T_j$  вычисляется как  $w_j \max\{0, C_j - d_j\}$ .

Ранее мы приводили классификацию задач **ТР** в зависимости от типа целевой функции. Теперь мы можем привести конкретные примеры. Можно выделить следующие критерии оптимальности:

1. *минимаксные критерии* – в задачах с такими критериями целевая функция представляет собой функцию максимума от значений штрафов требований. Примеры, минимаксных критериев:

- $C_{\max} \rightarrow \min$  – критерий минимизации максимального момента завершения требований,  $C_{\max} = \max_{j \in N} C_j$ . Задачи с такой целевой функцией называют задачами на *быстродействие*, (makespan – в англоязычной литературе);
- $L_{\max} \rightarrow \min$  – критерий минимизации максимального временного смещения  $L_{\max} = \max_{j \in N} L_j$ .

2. *суммарные критерии* – в задачах с такими критериями целевая функция представляет собой сумму значений штрафов требований. Примеры суммарных критериев:

- $\sum_{j \in N} C_j \rightarrow \min$  – критерий минимизации суммарного времени окончания обслуживания требований;
- $\sum_{j \in N} T_j \rightarrow \min$  – критерий минимизации суммарного запаздывания требований;
- $\sum_{j \in N} U_j \rightarrow \min$  – критерий минимизации количества запаздывающих требований.

В **ТР** также исследуются задачи на максимизацию аналогичных целевых функций, например,  $\sum_{j \in N} T_j \rightarrow \max$ .

### 1.2.3 Построение расписания для проекта. Project scheduling (*PS*)

В данном разделе принято выделять одну базовую задачу. Остальные задачи раздела, как правило, либо являются ее частными случаями либо же модификациями. Это задача *построения расписания выполнения работ проекта с учетом отношений предшествования и ограничения на ресурсы* (Resource-Constrained Project Scheduling Problem. *RCPSP*).

**Проект** – совокупность взаимосвязанных действий, направленных на достижение конкретных целей. К примеру, проект “строительство дома” может состоять из работ “выемка грунта”, “укладка фундамента”, “возведение стен первого этажа” и т.д. Очевидно, что по технологии “укладка фундамента” производится после “выемки грунта”, и т.д. В этом случае говорят, что между работами существует отношение предшествования.

В задаче *RCPSP* необходимо построить оптимальное расписание проекта (выполнения работ проекта) с учетом сетевого графика (отношений предшествования между работами) и с учетом необходимых/доступных ресурсов, при котором будет оптимизирована некоторая целевая функция. Самая популярная целевая функция – общее время выполнения проекта (*makespan* или  $C_{\max}$ ).

Данные задачи часто возникают на практике. Например, при строительстве того или иного объекта на разных стадиях строительства необходимо разное количество трудовых ресурсов, строительной техники, материалов и т.п. Между отдельными стадиями строительства существуют отношения предшествования обусловленные технологией. Требуется составить расписание выполнения работ, при котором не нарушаются отношения предшествования, ресурсные ограничения, и при этом срок окончания строительства был бы минимален. Алгоритмы решения этой задачи используются в известных программных продуктах Microsoft Project, Spyder Project, Primavera, 1С: Управление Строительной Организацией и т.д.

**Постановка задачи *RCPSP*** звучит следующим образом.

Дано множество требований  $N = \{1, \dots, n\}$  и  $K$  возобновляемых ресурсов  $k = 1, \dots, K$ . В каждый момент времени  $t$  доступно  $Q_k$  единиц ресурса  $k$ . Заданы продолжительности обслуживания  $p_i \geq 0$  для каждого требования  $i = 1, \dots, n$ . Во время обслуживания требования  $i$  требуется  $q_{ik} \leq Q_k$  единиц ресурса  $k = 1, \dots, K$ . После завершения обслуживания требования, освобожденные ресурсы в полном объеме могут быть мгновенно назначены на обслуживание других требований.

Между некоторыми парами требований заданы ограничения предшествования:  $i \rightarrow j$  означает, что обслуживание требования  $j$  начинается не раньше окончания обслуживания требования  $i$ .

Обслуживание требований начинается в момент времени  $t = 0$ . Прерывание при обслуживании требований запрещены.

Необходимо определить моменты времени начала обслуживания требований  $S_i$ ,  $i = 1, \dots, n$ , так, чтобы минимизировать время выполнения всего проекта, т.е. минимизировать значение

$$C_{\max} = \max_{i=1, \dots, n} \{C_i\},$$

где  $C_i = S_i + p_i$ . При этом должны быть соблюдены следующие ограни-

чения:

- 1) в каждый момент времени  $t \in [0, C_{\max})$  должно выполняться  $\sum_{i=1}^n q_{ik} \varphi_i(t) \leq Q_k$ ,  $k = 1, \dots, K$ , где  $\varphi_i(t) = 1$ , если требование  $i$  обслуживается в момент времени  $t$  и  $\varphi_i(t) = 0$ , в противном случае. То есть требования в процессе своего обслуживания должны быть полностью обеспечены ресурсами;
- 2) не нарушаются отношения предшествования между требованиями, т.е.  $S_i + p_i \leq S_j$ , если  $i \rightarrow j$  для  $i, j \in N$ .

Значение  $C_{\max}$  в англоязычной литературе называется *makespan*. К сожалению, нам неизвестен адекватный перевод этого термина на русский язык. На рисунке 1.4 представлен пример задачи с сетевым графиком и двумя расписаниями – оптимальным и неоптимальным.

Требования (работы) в такой задаче могут быть, например, такими: “выемка грунта”, “укладка фундамента” и т.п. В качестве необходимых ресурсов могут выступать: экскаваторы, разнорабочие, каменщики и т.д. Поэтому для выполнения работы “выемка грунта” может потребоваться одновременное участие 3-х экскаваторов, прораба и 7-ми разнорабочих.

Необходимо заметить, что существует класс задач *RCPSP* с невозобновимыми ресурсами, например, деньги, горюче-смазочные материалы и т.п.

Необходимо заметить, что существует класс задач *RCPSP* с невозобновимыми ресурсами, например, деньги, горюче-смазочные материалы и т.п.

#### 1.2.4 Построение расписания для приборов.

##### Machine scheduling (*MS*)

В отличие от Project Scheduling, где для выполнения одной работы требуется одновременное участие нескольких исполнителей, в задачах для

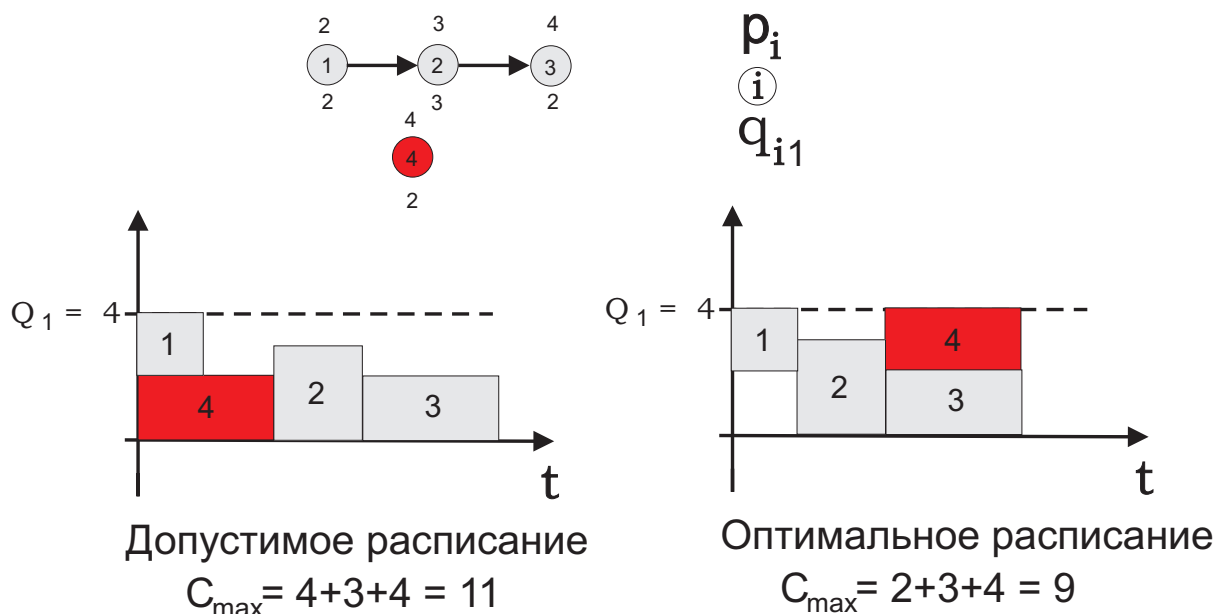


Рис. 1.4. Пример задачи *RCPSP*.

приборов каждое требование обычно выполняется (обслуживается) одновременно только на одном приборе (машине).

---

Для задач *MS* исполнителями являются *Приборы*, *Машины* или *Процессоры*. Если не приводятся уточнения, эти три термина считаются эквивалентными.

---

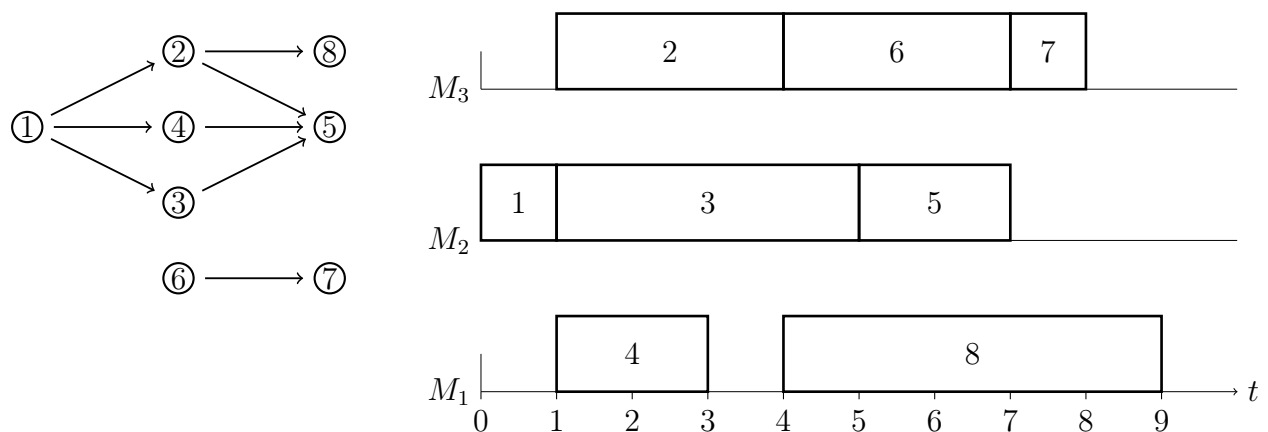
#### Задачи для одного прибора

С примерами задач для одного прибора мы познакомились в предыдущих разделах. Их характерной особенностью является то, что одновременно прибор может обслуживать (выполнять) только одно требование (задание). Несмотря на кажущуюся простоту в определении, решение одноприборных задач представляется трудной, но важной вычислительной проблемой. Алгоритмы их решения используются для решения более сложных задач, поэтому одноприборные задачи называются фундаментальными.

## Задачи для параллельных приборов

Для этих задач вместо одного прибора доступно  $m$  приборов  $M_1, M_2, \dots, M_m$ . Между требованиями могут быть заданы отношения предшествования. Каждое требование может выполняться на любом приборе. Если приборы идентичны, то время обслуживания  $p_j$  требования  $j$  не зависит от выбора машины, на которой требование будет обслужено. Эта задача соответствует частному случаю задачи *RCPSP*, где  $K = 1$ ,  $Q_1 = m$  и необходимое количество ресурса  $q_{j1} = 1$ , для всех требований  $j \in N$ .

**Пример.** Рассмотрим пример на рис. 1.5. Дано  $n = 8$  требований и  $m = 3$  приборов, времена обслуживания  $p_1 = 1, p_2 = 3, p_3 = 4, p_4 = 2, p_5 = 2, p_6 = 3, p_7 = 1, p_8 = 5$ . Допустимое расписание, для которого  $C_{\max} = 9$ , представлено на том же рисунке.



**Рис. 1.5. Расписание для идентичных машин**

Помимо идентичных приборов, могут рассматриваться приборы с разной производительностью. Для каждой работы  $j$  и прибора  $k$  может быть задано свое время  $p_{jk}$  обслуживания требования  $j$  на приборе  $k$ .

## Задачи Цеха (Shop Scheduling)

В этих задачах каждое требование состоит из операций, выполнение которых может назначаться только на определенные приборы (машины).

В общем случае дано  $m$  приборов  $M_1, M_2, \dots, M_m$  и каждое требование  $j$  содержит операции  $O_{1j}, \dots, O_{n_jj}$ . Между операциями могут быть заданы отношения предшествования (маршрут обработки детали). Две операции одного и того же требования не могут выполняться одновременно, и каждый прибор может выполнять одновременно только одну операцию. Время выполнения операции  $O_{ij}$  равно  $p_{ij}$ , и она может выполняться на машине  $\mu_{ij} \in \{M_1, M_2, \dots, M_m\}$ . Данная задача также может быть преобразована (сведена) в задачу *RCPSP*.

Далее мы опишем важные частные случаи задачи цеха.

### **Job-shop**

Для данного случая заданы отношения предшествования между операциями вида  $O_{1j} \rightarrow O_{2j} \rightarrow \dots \rightarrow O_{n_jj}$ . При этом нет отношений предшествования между отдельными требованиями. Количество операций у разных требований может быть различным.

### **Flow-shop**

Для данного частного случая задачи цеха каждая работа состоит из одних и тех же операций, т.е.  $n_j = m, \forall j \in N$ , а также задана машина, на которой обслуживаются операции, т.е.  $\mu_{ij} = M_i, i = 1, \dots, m, j = 1, \dots, n$ . Тогда расписание для каждого прибора задается вектором – порядком обслуживания операций, относящихся к разным работам. В русскоязычной литературе данные задачи порой называют задачами конвейерного типа.

### **Open-shop**

Задачи этого типа имеют такую же постановку, как и *Flow-shop* задачи. Единственное отличие – отсутствие отношений предшествования между операциями. То есть количество операций у каждого требования равно



$m$ , но порядок их выполнения может отличаться для разных требований.

### Прочие задачи Machine scheduling

Особняком стоят задачи, в которых на приборы налагаются специальные ограничения.

Например, в задачах batching (“группирования” или “партий”) один прибор может обслуживать одновременно несколько требований. При этом все требования из одной и той же “партии” имеют одно и то же время начала обслуживания и одно и то же время окончания.

В мультипроцессорных задачах обслуживание требования (операции) может производиться одновременно несколькими процессорами (приборами).

#### 1.2.5 Система обозначений для задач Machine Scheduling

В разделе Machine Scheduling для краткого обозначения задач принята специальная система обозначений. Эта система позволяет использовать не громоздкие названия задач (например, “Минимизация взвешенного суммарного запаздывания для одного прибора с разным временем поступления и одинаковой продолжительностью обслуживания требований”), а их краткое обозначение (соответственно,  $1|p_j = p, r_j| \sum w_j T_j$ ). Подробнее расскажем об этой системе обозначений, т.к. она повсеместно используется в публикациях по **ТР**.

Грэхем, Лаулер, Ленстра и Ринной Кан (Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G) в 1979-м году [35] для задач Machine Scheduling предложили трехпозиционную систему обозначений вида  $\alpha|\beta|\gamma$ , общепринятую на данный момент. Поле  $\alpha$  описывает характеристики задачи, связанные с приборами, и содержит всего одну запись. Значения в поле  $\beta$  уточняют характеристики обслуживания и ограничения, накладываемые на процесс обслуживания требований. Количество

записей в данном поле может быть произвольным (пустое значение, одна запись, несколько записей). Поле  $\gamma$  описывает целевую функцию задачи, значение которой необходимо минимизировать (или максимизировать), и обычно содержит всего одну запись.

В поле  $\alpha$  допустимы следующие значения:

- 1 – задача для одного прибора;
- $Pm$  – идентичные параллельные приборы. Количество идентичных приборов равно  $m$ , а  $P$  расшифровывается как *parallel*, то есть параллельные или идентичные приборы;
- $Qm$  – параллельные приборы с различной производительностью;
- $Fm$  – системы типа *Flow-Shop*;
- $Om$  – системы типа *Open-Shop*;
- $Jm$  – системы типа *Job-Shop*.

В этом поле могут быть указаны также другие записи, например (*sa*) или (*nd*), поясняющие характер обслуживания требований, о которых будет рассказано позже.

Ограничения и условия, накладываемые на обслуживание требований, перечисляются в поле  $\beta$  с помощью перечисления одной или нескольких записей:

- $r_j$  – моменты поступления (*release dates*). Если данное значение указано в поле  $\beta$ , то обслуживание требования  $j$  не может быть начато ранее его момента поступления  $r_j$ . Если  $r_j$  отсутствует в поле  $\beta$ , то предполагается, что все требования поступают на обслуживание одновременно в момент времени  $t = 0$ ;
- $D_j$  – предельные сроки завершения обслуживания требований;

- *pmnt* – допустимы прерывания (*preemption*). Если этот параметр опущен, то прерывания обслуживания требований запрещены;
- *prec* – отношения предшествования (*precedence relations*). Вместо этой записи в обозначениях задач можно встретить записи *tree*, *in – tree*, *out – tree* или *chain*. То есть отношения предшествования заданы в виде: дерева; входящего, выходящего дерева; цепочек;
- *batch(b)* – эта запись означает, что рассматривается задача, где требования обслуживаются партиями. То есть речь идет о задачах типа *batching*.

В поле  $\beta$  могут быть указаны и другие понятные значения. Например, запись  $p_j = p$  означает, что для всех требований задана одна и та же продолжительность обслуживания (константа  $p$ ).

В поле  $\gamma$  указывается целевая функция. Классические целевые функции были перечислены выше.

Согласно этой системе обозначений запись  $F2|r_j|C_{\max}$ , например, означает задачу “Минимизация общего времени обслуживания требований для системы Flow-shop с двумя приборами при неодновременном поступлении требований”. Думаем, читателю не составит труда разобраться и в записях вида:  $1|p_j = p, r_j| \sum w_j T_j$  или  $Pm|r_j, pmtn| \sum C_j$ .

### 1.2.6 Составление временных таблиц (Time Tabling)

Каждый из нас сталкивался с учебным расписанием в школе или в ВУЗе. Чаще всего учебное расписание для группы студентов представляется в виде таблицы (поэтому этот раздел **ТР** называется “*Time Tabling*”), в которой на пересечении строк (дни недели, время лекций) и столбцов (номер группы) указан предмет и номер аудитории, в которой состоится занятие по этому предмету. Общее расписание ВУЗа – совокупность расписаний для каждой группы. Фактически в таком расписании согласованы между собой во времени аудитории, группы учащихся и

преподаватели. Составление таких расписаний, порой, нелегкая задача, особенно когда существует дефицит помещений, а количество занятий и групп студентов большое. При составлении расписания нужно учесть разнообразные требования к аудитории, времени и т.д. Приведем некоторые примеры таких условий:

- Условия, связанные с аудиториями. Понятно, что аудитория должна вмещать всех учеников, и в ней должно быть соответствующее оборудование. Единоновременно в аудитории может проходить только одно занятие;
- Условия, связанные со студентами. Желательно, чтобы между лекциями не было больших перерывов. Необходимо чтобы студент успел перейти в другой учебный корпус, если занятия проходят в разных зданиях;
- Условия, связанные с преподавателями. У преподавателей также есть свои личные предпочтения, например, в какие дни и время проводить занятия;
- Условия, предъявляемые к учебному процессу. Желательно, чтобы после занятий по физкультуре не было лекционных занятий.

Задачи *Time Tabling* возникают при планировании занятости персонала, при согласовании времени различных встреч и т.д. Зачастую задачи *Time Tabling* можно свести к задачам *Project Scheduling*.

### 1.3 Задачи управления транспортными системами

Важный класс задач управления транспортными системами можно описать в терминах сети, состоящей из точек (узлов), соединенных путями (дугами), по которым осуществляются различные виды перевозок (потoki). Задачи планирования транспортных перевозок восходят своими корнями к работам французского математика Гаспара Монжа [51].

Существенное продвижению в этом направлении было получено советским математиком и экономистом Леонидом Витальевичем Канторовичем [42], вследствие чего непрерывная классическая постановка транспортной задачи получила название задачи Монжа–Канторовича.

Рассмотрим дискретный вариант этой задачи — транспортную задачу линейного программирования.

Имеется  $m$  пунктов отправления  $A_1, \dots, A_m$ , в которых сосредоточены запасы однородных продуктов в количестве  $a_1, \dots, a_m$  единиц.

Имеется  $n$  пунктов назначения  $B_1, \dots, B_n$ , потребность которых в указанных продуктах составляет  $b_1, \dots, b_n$  единиц.

Известны  $c_{ij}$ ,  $i = 1, 2, \dots, m$ ;  $j = 1, 2, \dots, n$  — стоимости перевозки единиц груза от каждого  $i$ -го поставщика каждому  $j$ -му потребителю. Требуется составить план перевозок, при котором все продукты доставлены в пункты назначения, и суммарные затраты на перевозку всех грузов являются минимальными.

В случае, если  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ , т.е. суммарные запасы поставщиков равны суммарным запросам потребителей, формулировка транспортной задачи называется **замкнутой** транспортной моделью (**открытой** — в противном случае).

Выберем в качестве переменных  $x_{ij}$ ,  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ , объемы перевозок от  $i$ -го поставщика каждому  $j$ -му потребителю.

По условию задачи требуется обеспечить минимум суммарных затрат на перевозки. Следовательно, целевая функция задачи имеет вид:

$$(1.1) \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min.$$

Суммарное количество груза, направляемого из каждого пункта отправления во все пункты назначения, должно быть равно запасу груза в дан-

ном пункте:

$$(1.2) \quad \sum_{j=1}^n x_{ij} = a_i \quad \forall i = 1, \dots, m.$$

Вторая группа уравнений выражает требование удовлетворить запросы всех  $n$  потребителей полностью и имеет вид:

$$(1.3) \quad \sum_{i=1}^m x_{ij} = b_j \quad \forall j = 1, \dots, n.$$

Объемы перевозок — неотрицательные числа:

$$(1.4) \quad x_{ij} \geq 0 \quad \forall i = 1, \dots, m, \forall j = 1, \dots, n.$$

Как одна из задач линейного программирования, транспортная задача (1.1)–(1.4) может быть решена некоторым стандартным методом решения задач линейного программирования, кроме того, разработаны специальные методы, учитывающие специфику ограничений задачи [2].

Необходимо отметить, что задача в такой постановке не учитывает многих факторов, возникающих на практике. Перечислим некоторые из них.

- **Неоднородность доставляемых грузов.** Как правило, из разных пунктов необходимо доставить различные виды товаров. Часто груз заранее определяется пунктом назначения.
- **Ограничения на пропускную способность путей.** Данные ограничения определяют возможность перемещения транспорта и зависят от вида путей (железнодорожные, водные, автомобильные пути).
- **Ограничения на транспортные средства.** Здесь могут быть учтены как количество транспортных единиц, имеющихся в наличии, так и ограничения на их эксплуатационные свойства (грузоёмкость, скорость движения и т.д.).

- **Транспортные расходы** могут включать в себя расход на горючее, ремонтные, эксплуатационные работы, привлечение обслуживающего персонала.
- **Учет времени.** Зачастую необходимо не только минимизировать издержки, но и время выполнения заказов. Также необходимо учесть директивные сроки, к которым должны быть доставлены товары.

В силу перечисленных ограничений необходимым оказывается не только определение, из какого пункта в какой будет доставляться груз, но и построение маршрутов перемещения товаров, а также расписания движения транспорта. Последнее из перечисленных ограничений наиболее сильно указывает на связь транспортных задач с теорией расписаний.

Очевидно, что задачи транспортного планирования могут сильно различаться между собой в зависимости от вида транспортных систем. В данном пособии акцент сделан на задачах с жёсткими ограничениями на пропускную способность путей и транспортные средства. Типичным представителем систем с подобными характеристиками является железнодорожный транспорт. Однако подобные ограничения могут возникать и применительно к другому транспорту.

Рассмотрим пример задачи, исследуемой далее в главе 3. Имеются две железнодорожные станции и один локомотив. Необходимо построить расписание движения поезда, доставляющего заказы между станциями. Данная задача может быть проинтерпретирована как задача паромов, перевозящего грузы с одного берега на другой, или задача перевозки грузов между двумя городами посредством одной грузовой машины, т.е. существенным моментом здесь является жёсткое ограничение на количество транспортных средств.

Задачи с однопутными железными дорогами соответствуют транспортным задачам с ограничением на пропускную способность пути. Примерами таких путей могут также выступать узкие речные каналы и т.д.

Поэтому далее будем рассматривать задачи управления транспортными системами на примере железных дорог.

## 1.4 Классификация задач календарного планирования, возникающих на железнодорожном транспорте

На транспорте, как и в других областях хозяйственной деятельности, возникают различные задачи дискретной и непрерывной оптимизации. В рамках этих задач ставятся следующие вопросы:

- какой подвижной состав и в каком количестве приобретать?
- где и какие транспортные магистрали прокладывать?
- где и какую инфраструктуру (ремонтные, пассажирские и грузовые станции) разворачивать?
- какой персонал и в каком количестве нанимать?
- по каким маршрутам и по какому расписанию будет двигаться подвижной состав?

В общем, все задачи хозяйствующего субъекта можно разбить на три блока:

- **задачи стратегического планирования.** Полученные в рамках данных задач решения считаются условно-постоянными и используются без значительных изменений в течении длительного периода времени. Например, вопрос о том, где и какие дороги/станции строить, относится к данному блоку задач;
- **задачи тактического планирования.** Жизненный цикл решений этих задач, как правило, короче, и сами решения подразумевают более значительные корректировки. К этому блоку можно отнести



задачи о покупке локомотивов или построения годового расписания движения.

- **задачи оперативного уровня.** Данные задачи возникают в режиме реального времени, т.к. их возникновение связано с различными видами неопределенностей. В какой состав включать грузовой вагон, на какой путь подать опоздавший поезд, как изменить расписание движения транспорта на участке с учетом некоторой поломки? Все это вопросы оперативного уровня.

Так как в этой книге речь идет о задачах календарного планирования, то приведенная ниже классификация относится к проблемам, так или иначе связанным с расписаниями. Или же расписания используются как входные данные, или же они строятся в результате решения задачи. Приведенные ниже классификации дополняют классические классификации задач Теории Расписаний, приведенные в предыдущей главе.

#### **Классификация задач по уровню и горизонту планирования:**

- задачи стратегического планирования;
- задачи тактического планирования;
- задачи оперативного уровня.

#### **Классификация задач по характеру искомого решения:**

- **задачи маршрутизации** (например, задача выбора маршрута движения поездов);

$RWR|...|...$  — railway routing

Например,  $RWR, 1|2|\sum T_j$  — задача с одним локомотивом, двумя станциями и критерием минимизации суммарного запаздывания.

$RWR, m|k, chain|w_j L_j$  — задача с  $m$  локомотивами,  $k$  станциями, расположенными в виде цепочки, и критерием минимизации взвешенного временного смещения.

- **задачи календарного планирования** (например, задача построения расписания движения поездов по выбранным маршрутам);

$RWRF|...|...$  — railway routing fixed.

Например,  $RWRF, m|k, prec|\sum C_j$  — задача с  $m$  локомотивами,  $k$  станциями, расположенными в виде произвольного графа, с критерием минимизации среднего времени выполнения всех заказов.

- **задачи объемного планирования** (например, задача о выборе уже сформированных составов, в которые включать тот или иной заказ на грузовую перевозку);
- **задачи объемно-календарного планирования** (задача о формировании составов и построении расписания движения для этих составов);
- **задачи объемно-календарного планирования и маршрутизации** (задача о формировании составов и построении расписания и маршрутов движения для этих составов);
- **прочие оптимизационные задачи**, использующие уже построенные расписания. Например, задача о покупке локомотивов может использовать в качестве исходных данных уже построенное годовое расписание движения, из которого можно извлечь информацию о потребностях в локомотивах.

## Классификация задач по характеру управляемых ресурсов

- **задачи подвижного состава** (например, задача о выборе последовательности перевозки заказов ограниченным числом локомотивов, где нет ограничений на используемые транспортные магистрали, т.е. пропускную способность графа дорог. В данной задаче ресурс — локомотивы, требования — заказы. Решением для данных задач является расписание движения локомотивов);

- **задачи управления участком пути** (например, задача о последовательности прохождения составами заданной станции или участка пути. В данной задаче ресурс - участок пути или станция, требования - локомотивы. Решением задачи является последовательность прохождения локомотивами участка);
- **задачи управления людскими ресурсами** (например, задача о построении индивидуальных рабочих графиков для поездных или ремонтных бригад);
- **смешанные задачи** (например, задача станционного диспетчера, который выбирает на какой перон какой поезд подать. Здесь решением будут маршрут движения каждого состава и расписание их движения по станции);

## 1.5 Обзор публикаций по задачам железнодорожного планирования

Публикации, посвященные железнодорожной тематике, можно разбить по типу исследуемых задач на следующие группы:

- стратегические задачи проектирования инфраструктуры железнодорожной сети (проектирование сети, расположения обгонных линий, средств эксплуатации железнодорожных объектов);
- стратегические задачи управления парком локомотивов и вагонов;
- задачи планирования расписаний и маршрутов грузоперевозок.

### 1.5.1 Стратегические задачи проектирования инфраструктуры железнодорожной сети

С первым классом задач связаны следующие работы. В 1994 г. Хиггинс [36] исследовал проблему расположения обгонных путей на однопутной

ных линиях для решения задачи минимизации запаздывания движения составов. В 1990 г. Краай [43] решал задачу планирования расширения бразильской железнодорожной сети. В модели минимизировались средние затраты во времени на перемещение гружёных и порожних составов для имеющейся железнодорожной сети. Руководствуясь соображениями о том, что улучшенное содержание путей уменьшает время следования, которое в свою очередь повышает производительность труда и качество обслуживания клиентов, а также уменьшает вероятность аварий, в 1976 г. Леблан [46] разработал модель нелинейной оптимизации, в которой инвестиции в усовершенствование путей для увеличения скорости движения составов и снижения эксплуатационных затрат снижают предельный доход. В 1989 г. Ачарья [12] решал задачу оптимизации размещения старых рельсов на участках с менее интенсивным движением взамен их утилизации. В 1997 Феррейра и Мюррей [32] в целях максимизации чистых доходов субъектов деятельности железнодорожной отрасли исследовали нормы проектирования и возможности сегмента железной дороги справляться с растущими нагрузками и скоростями железнодорожных составов.

### **1.5.2 Стратегические задачи управления парком локомотивов и вагонов**

Для фиксированного расписания задача управления парком железнодорожных составов заключается в присоединении порожних и гружёных вагонов и локомотивов к поездам для удовлетворения требований по срокам отправления и спросу на порожние вагоны. Многие исследователи берут перемещение гружёных вагонов как исходную информацию и оптимизируют перемещение порожних. В 1989 Тёрнквист и Марковиц [65] исследовали проблему минимизации затрат на перемещение, хранение и просрочку заказов, решая задачу маршрутизации порожних вагонов. Их модель допускала разные типы вагонов, их взаимозаменяемость и фиксированное расписание. Сформулирована задача минимизации за-

трат при однопродуктовом потоке в сети. Учитывались пространственные и временные ограничения на предложение и спрос вагонов. В 1998 г. Шерали и Сухарко [62] занимались проблемами оптимизации перемещения вагонов-автовозов. При таких допущениях, как возможность наличия просроченных заказов, фиксированное расписание и неопределённое время в пути, решалась задача минимизации максимального взвешенного запаздывания при ограничениях на соотношение между постоянным совокупным объёмом вагоно-дней и общим объёмом вагонов. В 1997 г. Зиарати [67] решал задачу оптимизации составления недельного расписания перемещения локомотивов между точками перецепки с учётом заездов на пункты дозаправки и планового обслуживания. Минимизировались затраты на перемещение локомотивов при ограничениях на требуемую мощность подвижного состава, плановое обслуживание, пропускную способность ремонтных цехов и спрос на локомотивы. Для решения использовался метод ветвей и границ с декомпозицией Данцига-Вольфа. Узким местом модели является наличие различных типов локомотивов. В 1998 Пауэлл и Карвальо [57] рассматривали задачу маршрутизации и обработки порожних и гружёных вагонов для максимизации дохода. Предполагалось, что расписание задано и фиксировано. Максимизировалась прибыль при ограничении на то, что только перемещения гружёных вагонов в допустимые временные окна приносят прибыль. Был разработан приближённый метод декомпозиции с использованием “жадного” алгоритма. Позже данный подход был применен к решению задачи маршрутизации и обработки разных типов вагонов.

Рассмотрим далее работы, посвященные оптимизации масштабирования и размещения парка вагонов и локомотивов. В 1991 г. Божон и Тюрнквист [16] попытались одновременно определять размер парка и маршруты движения вагонов для максимизации ожидаемых доходов от транспортировки грузов за вычетом издержек на приобретение и эксплуатацию оборудования, аренду и лизинг. В данной модели рассматривался стохастически изменяющийся спрос на перевозку. Время перевозки

описывается вероятностными функциями. В 1997 г. Шерали и Тунчбилек [63] исследовали задачу минимизации количества дополнительных вагонов, необходимых для удовлетворения спроса на всех сегментах железнодорожной сети во времени. Они предложили модель потоков в сети с изменяющимся во времени спросом.

### **1.5.3 Задачи планирования расписаний и маршрутизации грузоперевозок**

Работы, связанные с планированием и оптимизацией расписаний и маршрутизации, разбиты на три класса:

- формирование расписания движения поездов;
- планирование маршрутизации;
- комбинированные модели расписаний и маршрутизации.

#### **Формирование расписания движения поездов**

Одной из самых ранних работ была работа Шпигеля [64], опубликованная в 1973 г. В ней рассматривалась однопутная железная дорога с возможностью обгона на станциях. Шпигель первым заметил большое сходство между задачей составления расписания поездов и хорошо известной задачей теории расписаний для нескольких приборов. В условиях железнодорожной задачи поездам (работам) требуется несколько участков пути (приборов) для того чтобы пройти свой маршрут. Операции ограничены по времени и пространству. Временные ограничения состоят в том, что нельзя пройти данный участок пути, не пройдя перед этим предшествующий ему. Пространственные ограничения запрещают находиться на одном участке пути более, чем одному поезду. Для предотвращения нахождения более одного состава на участке Шпигель определил так называемые порядковые ограничения. Поставленная задача

может быть решена с помощью метода ветвей и границ. Релаксированная задача линейного программирования не включает в себя порядковые ограничения. Ветвление необходимо только при конфликте поездов. В итоге Шпигель рассмотрел случаи, включающие 5 участков пути и 10 поездов.

В 1998 г. Брэннлунд [21] рассматривал модель с однокольной дорогой, движение поездов по которой происходит с заданной, не меняющейся скоростью. Максимизировалась прибыль для выбранных маршрутов при ограничении на пропускную способность каждого сегмента железнодорожной сети. В 1983 Содер и Вестерман [60] разработали систему планирования прохождения поездами перекрёстков. При ограничениях на скорость поездов и длины линий обгона минимизируется суммарное взвешенное запаздывание с учётом приоритетов поездов. Оптимальное решение отыскивается пересмотром всех пересечений. В 1991 г. Йованович и Харкер [41] разработали модель SCAN-I расписаний на множестве взаимосвязанных полос движения. Чтобы определить, насколько расписание достижимо, используется модифицированный метод ветвей и границ, который выполняется во времени, разрешая конфликты последовательно, избегая тупиковых ситуаций. Перебор с возвратом возникает, когда реалистичный вариант расписания не может быть сгенерирован. В том же 1991 г. Краай [43] предложил подход к построению расписаний, допускающий переменную скорость поездов. При выполнении ограничения на максимальную скорость поездов минимизируются расход топлива и отклонения от плана.

Другой подход использовали Кэри и Локвуд [29] в 1995 г. Авторы поставили задачу целочисленного программирования с бинарными переменными. Каждая бинарная переменная регулирует порядок каждой пары поездов на данном участке. В отличие от модели Шпигеля, задача включает в себя все порядковые ограничения. Учитывая размер задачи, авторы использовали эвристический подход, который последовательно упорядочивает поезда. В 1996 г. Хиггинс [37] предложил метод ветвей и

границ для однопутной дороги, имеющий сходства с методом Шпигеля. В начальную постановку задачи не входят взаимодействия поездов. В каждой вершине вычисляется не только текущее значение задержки поезда, но и нижняя граница ожидающейся задержки из-за нерешенных конфликтов. Эта сумма определяет цену вершины. В работе рассмотрены случаи с 30 поездами и 12 путями. Кай и Го [24] в 1994 г. предложили простой эвристический жадный алгоритм для той же задачи. Они сделали предположение, что все поезда, идущие в одном направлении, имеют не только одинаковую скорость, но и один и тот же путь. В 1998 г. Кай [25] представил двухфазовый подход. На первой фазе происходило обновление текущего времени и координат. Во второй фазе применялась усовершенствованная версия жадного эвристического алгоритма. В 1999 г. Сахин [59] предложил эвристический алгоритм, определяющий конфликты в порядке их появления и последовательно разрешающий их. В 2000 г. Оливейра и Смит [54] предложили постановку задачи для нескольких приборов применительно к железнодорожной тематике. Конфликты разрешались в хронологическом порядке, целевая функция минимизировала суммарное запаздывание.

## **Задачи маршрутизации перевозок**

Рассмотрим работы по маршрутизации перевозок. Ньютон [52] в 1998 г. разработал модель минимизации суммарного пробега, затрат на обработку и издержки простоя вагонов с ограничениями на сроки доставки и обработку составов и вагонов на сортировочных станциях. Учитывались приоритеты заказов на перевозки при фиксированном спросе. Для решения задачи был использован модифицированный метод ветвей и границ. В 1998 г. Квон [45] исследовал задачу многопродуктового потока в сети, чтобы определить маршруты вагонов с учётом предположений о том, что расписание поездов, планы формирования блоков и привязки блоков к поездам известны. Задача состоит в минимизации штрафа за запаздывание поставки с учётом того, что спрос удовлетворён, вагоны нужным



образом объединены в блоки, блоки распределены по поездам, ограничение на вместимость поездов выполнено. В 1997 г. Нозик и Морлок [53] рассматривали дискретную во времени задачу минимизации общих переменных затрат на перемещение гружёных и порожних вагонов, если фиксировано расписание поездов и известен спрос на вагоны. Задача решается при ограничениях на удовлетворение спроса, на размер парка и пропускные способности сортировочных станций. Была разработана итеративная процедура линейного программирования с округлением получаемых дробных величин до тех пор, пока не будет достигнуто приемлемое целочисленное решение. Показано, что эвристические методы дают хорошие результаты.

Наиболее общим подходом к моделированию задачи маршрутизации поездов является использование методики графа противоречий. С помощью этой методики для моделирования задачи используются два довольно разных подхода: задача упаковки и задача раскрашивания графа. В 1996 г. Цванвельд и др. [68] рассмотрели постановку задачи упаковки с одинаковой стоимостью. Другими словами, все железнодорожные пути равномерно вносят вклад в одну из целевых функций. В 2001 г. постановка задачи была расширена [69]. Авторы применили похожий метод моделирования. Однако было сделано разумное предположение, что поезда предпочитают определенные пути остальным. Таким образом, были сформулированы задачи взвешенной упаковки. В отличие от Цванвельда в 2003 г. Делорме [31] определяет несколько целевых функций, которые рассматриваются в лексикографическом порядке. Задача заключается в определении максимального числа поездов, которые можно направить через данный железнодорожный узел. Для решения предложен метаэвристический метод, использующий процедуру жадного случайного адаптивного поиска. Согласно результатам проведенных численных экспериментов, погрешность полученных решений для рассмотренных примеров не превышает 3%. В 2005 г. Гандибло [33] предложил как альтернативу метаэвристический алгоритм "муравьиных колоний".

В 2007 г. Капрара [27] изучил задачу подготовки платформ, состоящую в назначении прибывающих на станцию поездов к доступным платформам. Он представил формулировку задачи целочисленного программирования, основанную на графе несовместимости путей. Ограничение модели — нахождение не более одного поезда на одном пути. Задача модели заключается в минимизации суммы весов, определенных с использованием платформы. Были рассмотрены несколько случаев, включающих в себя до 237 поездов за 24-часовой период. Метод показал лучшие результаты, чем эвристика, используемая в железнодорожной компании. Де Лука Кардильо и Мионе [28] (1998 г.), Биллионе [18] (2003 г.) и Корнельсен и Ди Стефано [30] (2007 г.) предложили метод раскраски графа для задачи подготовки платформ. В 2007 г. Родригез [58] определил, что время ожидания состоит из двух типов задержки: замедления и ожидания. Кроме того, существует задержка при наборе скорости поезда. Целевой функцией задачи является минимизация суммы задержек. Автор протестировал и сравнил два варианта формулировок задачи условной оптимизации: первая рассматривает все три типа задержек, а вторая только первые два. Тестирование задачи со второй постановкой показало недооценку времени задержки по сравнению с актуальной. Тем не менее, метод потребовал намного меньше вычислительного времени и показал неплохие результаты. Тестовые данные включали в себя от 6 до 24 поездов и были основаны на железнодорожном узле Пьеррефит-Гонесс в Париже.

## **Комбинированные модели расписаний и маршрутизации**

В 1970 г. Морлок и Петерсон [50] представили одну из первых моделей совместного поиска маршрутов и расписаний. Для удовлетворения плана доставки скоропортящихся продуктов минимизировались суммарные постоянные издержки содержания поездов, переменные издержки на транспортировку, обработку и хранение груза, и альтернативные издержки использования железнодорожного оборудования. Вводятся огра-

ничения на время прибытия поезда, маршрут, набор остановок и допустимое количество прицепляемых вагонов. Решалось, какой поезд отправлять и какой груз в него загружать. Использовался метод ветвей и границ. В 1998 г. Горман [34] изучал дискретную задачу одновременного поиска возможных маршрутов без остановок и распределения грузов. При ограничениях на сроки доставки, вместимость поездов, обслуживание на сортировочных станциях и пропускные способности путей минимизировались затраты на эксплуатацию обслуживающего персонала и локомотивов, горючее, обработку грузов и другие затраты. Для решения задачи использовался модифицированный генетический алгоритм. В 1995 Хантли [38], работая над задачей крупномасштабных перевозок зерна, предложил процедуру формирования расписания поездов и маршрутизации вагонов. Каждый поезд определяется безостановочным маршрутом, пунктами отправления и назначения и временем прибытия. Путь подбирается для набора вагонов одного типа, пункта назначения и времени прибытия при фиксированном спросе. Задача заключается в минимизации нелинейной функции затрат на обслуживающий персонал, топливо, эксплуатацию вагонов и локомотивов при ограничениях, что все грузы должны быть отправлены и вместимость поезда не превышает допустимую. Для решения использовался метод имитации отжига (simulating annealing). В 1996 г. Марин и Салмерон [49] предложили агрегированную стационарную модель распределения грузов, в которой определяются маршруты поездов и количество вагонов. Минимизируются постоянные затраты на эксплуатацию поездов, затраты на обработку грузов, штрафы за задержку и затраты на добавочный поезд. Ограничения накладываются на число вагонов, допустимых для передвижения по каждому сегменту железной дороги и для обработки на каждой сортировочной станции и на число поездов. Использовались методы имитации отжига и поиска с запретом (tabu search) и др.

В 2002 г. Капрара [26] рассмотрел задачу составления расписаний для поездов через главные коридоры для Итальянских железных дорог. Ко-

ридор соединяет две главные станции, промежуточные станции и состоит из параллельных отрезков однопутных путей в каждом направлении. Авторы доказали, что задача является NP-трудной. Была смоделирована задача для ациклического ориентированного мультиграфа, содержащего множество вершин прибытия и отправления, а также стоковых и истоковых вершин. Позднее, в 2008 г., Каччиани и др. [23] описали метод генерации колонок для составления расписаний поездов в коридорах. В 2005 г. Борндорфер [19] предложил постановку, похожую на постановку Капрары. Однако, в отличие от Капрары авторы не рассматривали коридоры, соединяющие главные станции. Борндорфер предложил использовать метод аукциона для выделения оптимальной пропускной способности железнодорожной сети. В последующей работе (2007 г.) Борндорфер и Шлехте [20] рассмотрели задачу оптимального распределения путей с другой стороны. Они показали, что постановка задачи имеет постоянное число строк и поддается решению методом генерации колонок. В 1989 г. Серафини и Укович [61] сформулировали задачу планирования периодических событий, которая может быть использована для составления периодических расписаний в железнодорожной отрасли. Эта задача состоит в планировании множества периодических событий в заданные временные окна. Различные ограничения данной задачи описаны Питерсом [56] в 2003 г. Либхен и Моринг [47] в 2004 г. рассмотрели более сложные модели с использованием периодической задачи.

В 2009 Лиу и Козан [48] поставили задачу составления расписания движения поездов как задачу конвейерного типа с параллельными приборами и блокировками. При этом участки дороги рассматривались как приборы, а железнодорожные составы как работы. Это позволило рассматривать железнодорожную сеть, состоящую как из однопутных, так и многопутных участков. Авторами решалась задача минимизации времени выполнения всех работ (*makespan*). В основу метода решения была взята идея смещения узкого места (*shifting bottleneck*), к которой добавили возможность установки пауз в расписание для устра-

нения ситуаций блокировки, когда на одном одноколейном участке одновременно находятся два состава, движущихся в противоположенных направлениях. Впервые была взята в рассмотрение длина состава, т.е. брался во внимание тот факт, что при переходе состава с одного участка дороги на другой поезд некоторое время, зависящее от длины состава, находится на двух участках пути — в случае высокой загрузки железнодорожной сети это влияет на расписание. Эксперименты были проведены в Австралии на данных Квинслендской железной дороги. Была показана эффективность предложенных подходов. Применение данного подхода позволяет дать рекомендации при определении участков дороги, на которых расширение или увеличение скорости движения позволит существенно улучшить расписание движения.

#### **1.5.4 Обзор публикаций по железнодорожной тематике в России за последние годы**

Следует отметить исследования данной проблематики в некоторых статьях журнала Вестник ВНИИЖТ. Так в статье [1] представлена дорожно-сетевая модель динамики вагонных парков на сети железных дорог, обеспечивающая решение задач, связанных с изучением влияния неравномерности на перевозочный процесс, нестандартных ситуаций, возникающих на дорогах, и подготовкой рекомендаций по восстановлению ритмичной работы сети. Решение задачи было выполнено с использованием инструментов теории марковских процессов с конечным числом состояний. Применение этих процессов дает возможность составить линейные дифференциальные уравнения для вероятностей состояний, а также линейные алгебраические уравнения предельных вероятностей состояний, отражающих относительное время пребывания системы (в данном случае — вагонного парка) в каждом из этих состояний стационарного режима. Для выбора маршрутов продвижения вагонопотоков задача решается отдельно для груженых и порожних потоков по стоимостному критерию путем построения кратчайших путей в графе.

В данной работе применен итерационный алгоритм Дейкстры. Расчёты порожних вагонопотоков выполнены с помощью типового алгоритма транспортной задачи методом линейного программирования.

Была разработана дорожно-сетевая модель оценки влияния неравномерности грузовых перевозок и затруднений в использовании вагонного парка отдельных железных дорог на работу других дорог сети.

Транспортные системы, такие, как железнодорожные станции и транспортные узлы, имеют сильную структурную и функциональную связность. Расчётные технические и технологические параметры чаще всего не соответствуют реальным условиям работы, и возможный разброс в оценке эффективности бывает недопустимо велик.

В работе [3] предлагается применять имитационные модели при выборе решений по расчёту транспортных систем и управлении потоками. Описываются виды рекомендуемых моделей, приводятся примеры.

Опыт исследования транспортных систем позволил выработать четыре основных подхода к их моделированию, которые рассматриваются в данной статье:

- аналитический (расчет по формулам);
- графический (построение суточного плана-графика);
- теории массового обслуживания (вероятностная модель);
- имитационного моделирования (построение подробной компьютерной модели).

Анализ расчётов проектируемых транспортных систем показывает, что ошибки могут быть весьма значительными. Для повышения качества при построении моделей необходимо использование систем имитационного моделирования. В этой связи интересен опыт применения имитационной системы ИСТРА, а также подсистемы САПР. Подсистема САПР значительно облегчает процесс построения модели. При задании началь-

ных и конечных точек модель сама строит все возможные маршруты передвижения. Технологию остается только запретить ненужные. Имеется большой набор сервисных средств и механизмов контроля правильности исходной информации.

В настоящее время информационные системы предоставляют оперативным и административным руководителям огромные потоки слабо упорядоченных сведений. Системы автоматизированного анализа должны выявлять “узкие места” инфраструктуры, указывать на “болевые точки” современной технологии, плохую стыковку отдельных операций, причины возникновения межоперационных простоев. В аналитических системах должен проводиться и оперативный анализ эффективности принимаемых управленческих решений. В работе [4] при выборе решений по расчёту транспортных систем и управлении потоками предлагается применять потоковые модели на основе динамической транспортной задачи с задержками (ДТЗЗ). Описываются виды рекомендуемых моделей, приводятся примеры.

Появление разных собственников вагонов осложнило управление вагонопотоками, возрос оборот вагона и увеличился порожний пробег. Собственники вагонов предъявляют различные требования к маршрутам и режимам пропуска, что усложняет управление железнодорожными перевозками. Это резко усложняет задачу, и диспетчеру уже невозможно без специальных моделей рассчитать рациональные режимы в динамике на несколько суток вперед. Одним из возможных вариантов решения проблемы могут быть динамические потоковые модели.

Предложенные модели получили дальнейшее развитие в статье [5], где предлагается динамическая потоковая модель, которая позволяет рассчитать оптимальный вариант обращения вагонов разных собственников с рядом индивидуальных требований — ограничений по возможным маршрутам, времени оборота и т.п. Особенностью модели является отображение переходов вагонов из порожнего состояния в груженое и наоборот.

## Глава 2

# Модели задач железнодорожного планирования

Данная глава посвящена способам моделирования задач железнодорожного планирования. Рассматриваются различные методы построения математических моделей в зависимости от ограничений, уровня планирования (оперативный, стратегический) и т.д.

### 2.1 Задача формирования составов и расписания движения грузовых поездов

Проблема заключается в организации доставки грузов по Российским железным дорогам (РЖД). Необходимо сформировать грузовые составы и составить расписание их движения с целью минимизации запаздывания доставки грузов.

Сеть РЖД включает в себя около 76 (см. рис.2.1) сортировочных станций, между которыми осуществляются перевозки грузовых вагонов. Имеется множество заказов на перевозку грузов. Каждый заказ характеризуется станцией отправления, станцией назначения и имеет определенные директивный срок и вес (относительная “важность” заказа). Если заказ состоит из нескольких вагонов  $k > 1$ , то на каждый вагон оформляется отдельный заказ, т.е. в множестве заказов будет  $k$  идентич-



ных заказов. В дальнейшем будем предполагать, что множество заказов состоит из одновагонных заказов.



**Рис. 2.1. Сортировочные станции РЖД**

Запаздывание заказа определяется разницей между фактическим временем доставки последнего вагона с грузом на станцию назначения и директивным сроком. Запаздывание считается равным нулю, если доставка произошла раньше директивного срока.

Для перевозки грузов используются грузовые поезда (или составы, что является эквивалентным определением). Под железнодорожным составом будем понимать локомотив, локомотивную бригаду и некоторое множество вагонов. При этом должны удовлетворяться некоторые технические ограничения (максимальный общий вес вагонов, максимальная длина состава и т.п.). Каждый состав движется по определенному маршруту, состоящему из последовательности сортировочных станций. На сортировочных станциях к составу могут быть прицеплены или отцеплены вагоны, а также может быть сменен локомотив и локомотивная бригада. Расписание движения каждого состава состоит из последовательности времён прибытия на каждую станцию и времён отбытия с каждой станции на маршруте поезда.

### 2.1.1 Общая постановка задачи

Рассмотрим описанную выше проблему формирования составов и графиков движения грузовых поездов. Пусть  $N$  — множество станций, определяющих вершины графа  $G$  железнодорожной сети. Дорога от одной станции до другой разбита семафорами на блок-участки. Даны множества заказов  $O_{ij}, i, j \in N$ , для перевоза груза со станции  $i$  на станцию  $j$ ,  $O = \bigcup_{i,j \in N} O_{ij}$  — множество всех заказов. Каждый заказ представляет собой один вагон, для которого известны пункт отправления и пункт назначения. Заказ  $J_{ijk} \in N$  поступает на станцию  $i$  в момент времени  $r_{ijk}$  и должен быть перевезен в пункт назначения  $j$  не позже директивного срока  $d_{ijk} = r_{ijk} + p_{ijk} + \Delta$ , где  $p_{ijk}$  — продолжительность обслуживания заказа, которая определяется нормативно,  $\Delta$  — известная константа. В случае невыполнения заказа начисляется штраф  $z_{ijk}$ . Каждый заказ характеризуется своей значимостью  $w_{ijk}$  и весом (массой)  $m_{ijk}$ .

Пусть  $L_{oc}$  — множество имеющихся в наличии локомотивов, а  $B$  — множество локомотивных бригад. Под составом будем понимать объект, состоящий из трех элементов: вагонов, локомотива и локомотивной бригады. Для функционирования железнодорожной сети необходимо выполнение следующих ограничений.

**1. Ограничения по составу.** Количество вагонов в составе не должно превышать 71 вагон или быть меньше 30 вагонов. Вес состава ограничен 6000 т.

**2. Ограничения по горкам.** На горке не может формироваться более некоторого фиксированного числа составов, зависящего от параметров горки (как правило, не более 10).

**3. Ограничения по путям.** На одном участке (от семафора до семафора) не может находиться более одного состава в одном направлении. Кроме того, грузовой состав может идти по участку только в заданные интервалы времени  $[t_1, t_2], [t_3, t_4], \dots$  (остальное время может быть выделено под движение пассажирских составов, электричек или ремонт

путей).

**4. Ограничения по станциям.** На сортировочной станции  $i \in N$  не может одновременно находиться более  $sort_i$  составов.

**5. Ограничения по локомотивам и локомотивным бригадам.** Каждый локомотив  $l$  “приписан” к некоторому подмножеству станций  $G_l$ , причем  $G = \bigcup_l G_l$ . Локомотивная бригада  $b_i$  закреплена за некоторой станцией и может “обслуживать” подграф  $G_i^b$ .

Необходимо так сформировать составы и графики их движения, чтобы не были нарушены ограничения.

Пусть  $C_{ijk}$  — время завершения доставки  $k$ -го вагона со станции  $i$  на станцию  $j$ . Можно рассмотреть несколько вариантов целевой функции.

1. Минимизация суммарного запаздывания:

$$\min \sum_{J_{ijk} \in O} w_{ijk} \max\{0, C_{ijk} - d_{ijk}\}.$$

2. Минимизация среднего времени нахождения состава в пути:

$$\min \sum_{J_{ijk} \in O} C_{ijk}.$$

3. Минимизации максимального временного смещения:

$$\min \max_{J_{ijk} \in O} \{C_{ijk} - d_{ijk}\}.$$

4. Выбор подмножества заказов  $\bar{O} \subseteq O$ , которые можно выполнить без нарушения директивных сроков, дающего максимальную прибыль:

$$\max \sum_{J_{ijk} \in \bar{O}} w_{ijk} - \sum_{J_{ijk} \in O \setminus \bar{O}} z_{ijk}.$$

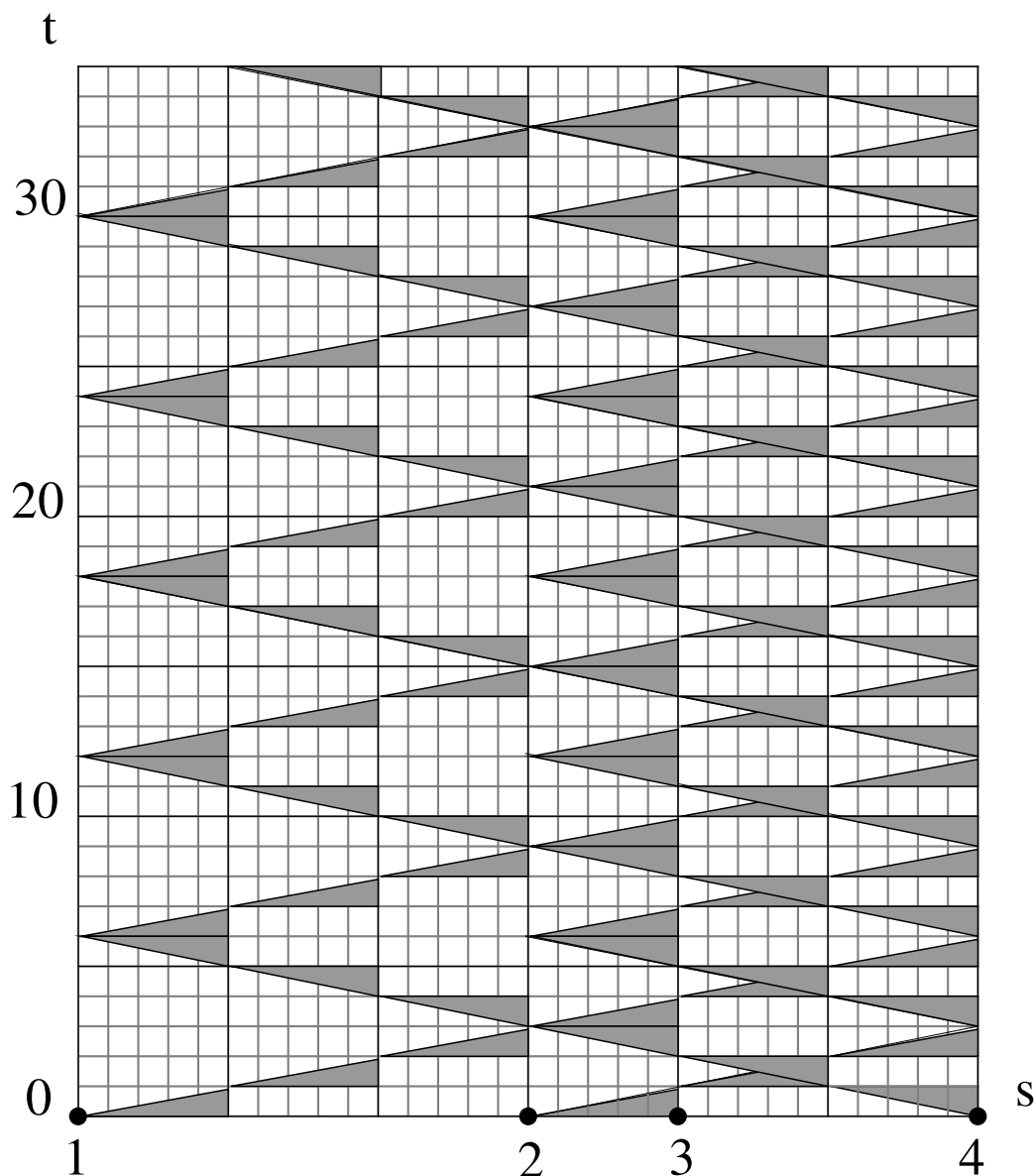
**Пример 2.1.1.1.** Рассмотрим следующую задачу формирования составов и расписания поездов. Имеется 4 железнодорожные станции, расположенные на одной прямой. Между станциями курсируют 4 локомотива.

В начальный момент времени на каждой станции находится по одному локомотиву. Необходимо перевезти между станциями 3 вида заказов, различающихся коэффициентом значимости  $w$ . Ниже в табл. 2.1 приведены параметры заказов. Здесь  $n$  — количество заказов (вагонов),  $r$  — время поступления заказа на станцию отправления,  $w$  — коэффициент значимости товара,  $d$  — директивный срок.

Станция отправления	Станция назначения	Товар	$n$	$r$	$w$	$d$
1	2	1	10	0	1	5
1	2	2	5	0	10	5
1	3	2	7	12	10	18
1	3	1	4	14	1	20
1	4	1	6	14	1	22
1	4	2	8	22	10	30
2	1	2	2	4	10	9
2	3	2	8	15	10	18
2	4	2	10	24	10	29
3	2	3	5	14	20	18
3	4	3	5	16	20	20
3	1	3	5	17	20	23
3	1	2	5	18	10	24
4	1	2	6	7	10	15
4	2	2	6	18	10	23
4	3	2	8	30	10	34

**Таблица 2.1. Параметры заказов**

Вместимость состава составляет 10 вагонов. Время, затрачиваемое на сортировку (погрузку-разгрузку заказов), равно одному дню. Расстояние между станциями 1 и 2 поезд преодолевает за 3 дня, между станциями 2 и 3 — за 1 день, между станциями 3 и 4 — за 2 дня. Необходимо составить расписание движения грузовых поездов, которое бы не препятствовало движению пассажирского транспорта. Графики движения пассажирских поездов приведены на рис. 2.2. В закрашенные временные интервалы движение грузового транспорта на соответствующих блок-участках и в соответствующем направлении запрещено. Например, левый нижний треугольник означает, что движение в направлении с первой на вторую станцию запрещено в первый день на первом блок-участке.



**Рис. 2.2.** Интервалы недоступности движения грузовых поездов

На рис. 2.3 приведено одно из возможных допустимых расписаний для данного примера.

В общем случае нахождение оптимального расписания, минимизирующего одну из перечисленных выше целевых функций, — нетривиальная задача, требующая индивидуального подхода для каждого случая. В частности, в последующих параграфах данной главы будут представлены некоторые математические модели для подобного рода задач.

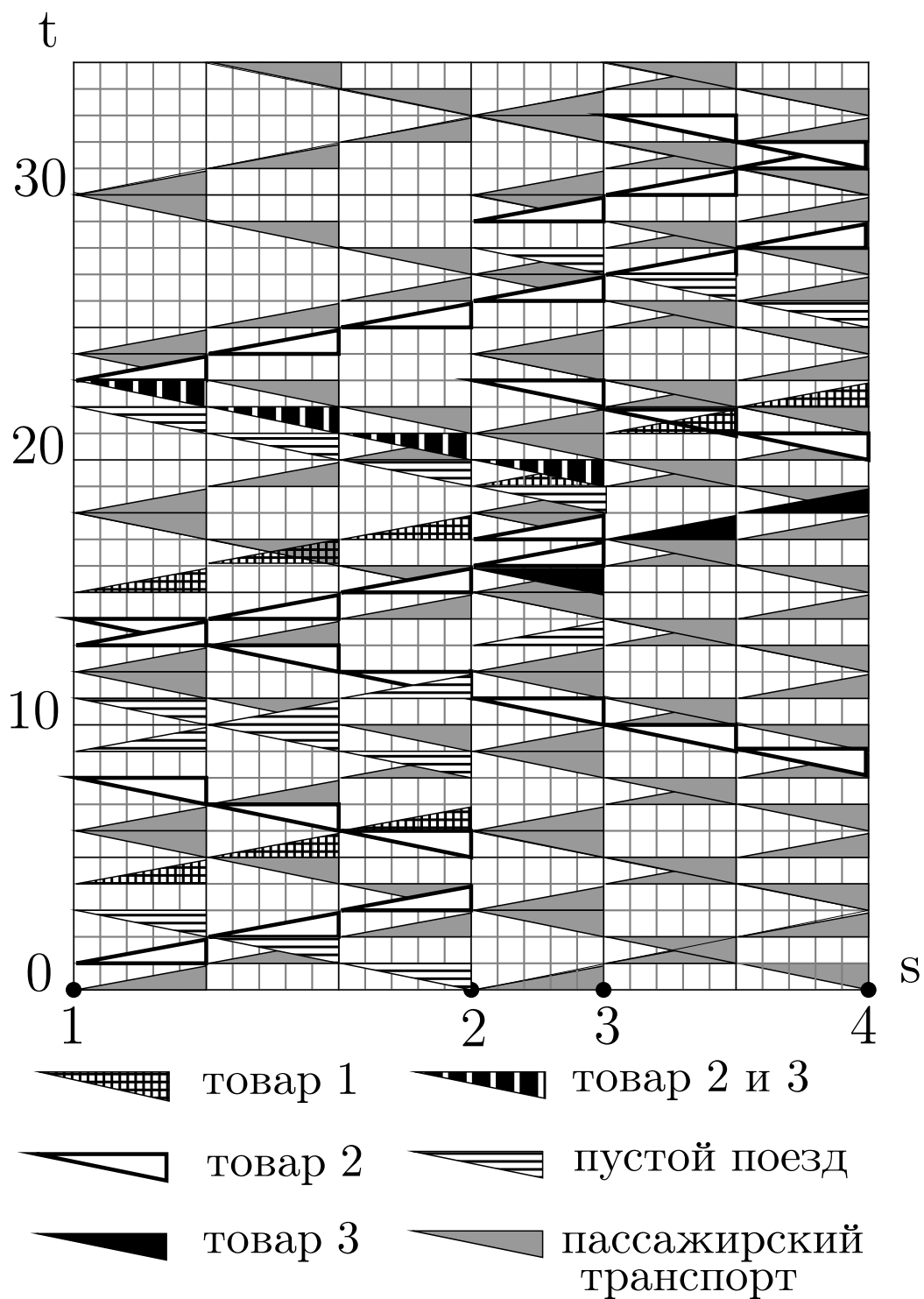


Рис. 2.3. Допустимое расписание для примера 2.1.1.1

Помимо общих моделей будем рассматривать некоторые частные случаи данной задачи с небольшим количеством станций. Такой подход поз-

воляет сосредоточиться на отдельных особенностях задачи. Кроме того, как будет показано в гл. 3, некоторые варианты задач имеют тесную связь с классическими задачами теории расписаний.

### 2.1.2 Задачи с двумя станциями

Будем рассматривать случай, когда необходимо обслужить всего две станции, соединенные двухпутной железной дорогой. Для каждой из четырех представленных выше целевых функций будем рассматривать следующие ограничения.

1. Ограничения на длину и вес состава.
2. На одном участке не может находиться более одного состава в одном направлении.
3. Ограничение на количество локомотивов. (Начальный вариант: 1-2 локомотива).
4. Грузовой состав может идти по участку только в заданные интервалы времени  $[t_1, t_2], [t_3, t_4], \dots$
5. На участке, соединяющем станции:
  - нет семафоров;
  - 1 семафор;
  - $k$  семафоров.

### 2.1.3 Задачи с цепочкой станций

В данном случае все станции расположены последовательно. Начальный вариант — 3 станции. Для каждой из четырех представленных выше целевых функций будем рассматривать следующие ограничения.

1. Ограничения на длину и вес состава.
2. На одном участке не может находиться более одного состава в одном направлении.

3. Ограничение на количество локомотивов. (Начальный вариант: 1-2 локомотива).

4. Грузовой состав может идти по участку только в заданные интервалы времени  $[t_1, t_2], [t_3, t_4], \dots$

5. “Привязка” локомотивов к подмножеству станций.

6. Ограничение на количество составов на одной станции.

#### **2.1.4 Задачи с замкнутой цепочкой станций**

В отличие от предыдущего раздела, цепочка последовательно расположенных станций замкнута. Начальный вариант — 3 станции, расположенные в форме треугольника. Для каждой из четырех представленных выше целевых функций будем рассматривать следующие ограничения.

1. Ограничения на длину и вес состава.

2. На одном участке не может находиться более одного состава в одном направлении.

3. Ограничение на количество локомотивов. (Начальный вариант: 1-2 локомотива).

4. Грузовой состав может идти по участку только в заданные интервалы времени  $[t_1, t_2], [t_3, t_4], \dots$

5. “Привязка” локомотивов к подмножеству станций.

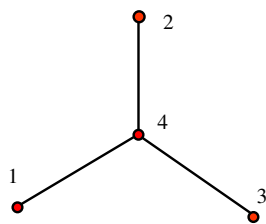
6. Ограничение на количество составов на одной станции.

#### **2.1.5 Задачи со станциями, расположенными в форме звезды**

Для начала будем рассматривать четыре станции с расположением в форме звезды (см. рис. 2.4).

Для каждой из четырех представленных выше целевых функций будем рассматривать следующие ограничения.





**Рис. 2.4. Расположение станций в задаче 2.1.5**

1. Ограничения на длину и вес состава.
2. На одном участке не может находиться более одного состава в одном направлении.
3. Ограничение на количество локомотивов. (Начальный вариант: 1-2 локомотива).
4. Грузовой состав может идти по участку только в заданные интервалы времени  $[t_1, t_2], [t_3, t_4], \dots$

В гл. 3 предлагаются методы решения некоторых из приведенных задач.

## **2.2 Целочисленная постановка задачи формирования составов и расписания их движения**

В данном параграфе мы приведем одну из возможных целочисленных постановок задачи формирования составов и расписания их движения, а также рассмотрим модификации этой постановки в зависимости от различных дополнительных ограничений.

### **2.2.1 Задачи целочисленного программирования**

Вначале остановимся на некоторых моментах теории целочисленного программирования.

Рассмотрим задачу линейного программирования (ЛП) в стан-

дартной форме:

$$(2.1) \quad \begin{aligned} \min cx, \\ Ax \leq b, \quad x \geq 0, \end{aligned}$$

где  $A$  —  $m \times n$ -матрица,  $c, x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$ .

В случае, если некоторые переменные задачи должны быть целыми (например, переменные, отвечающие за количество вагонов), получаем задачу **смешанного целочисленного программирования**:

$$(2.2) \quad \begin{aligned} \min cx + hy, \\ Ax + By \leq b, \\ x \geq 0, \quad y \geq 0, \quad y \in Z. \end{aligned}$$

Если же все переменные задачи целые, и, кроме того, могут принимать только значения 0 или 1, то получаем задачу **ЛП с булевыми переменными**:

$$(2.3) \quad \begin{aligned} \min cx, \\ Ax \leq b, \quad x \geq 0, \quad x \in \{0, 1\}. \end{aligned}$$

Наличие булевых переменных может быть обусловлено непосредственными условиями задачи (например, переменная равна 1, если поезд отправляется по некоторому пути, и 0 в противном случае). Часто введение дополнительных булевых переменных необходимо для формулировки альтернативных условий. Например, в некоторой задаче оптимизации решение  $x$  должно удовлетворять дополнительному условию

$$(2.4) \quad g(x) \geq 0 \text{ или } h(x) \geq 0.$$

Пусть  $g^{min}$  и  $h^{min}$  — известные нижние оценки для  $g(x)$  и  $h(x)$  на множестве допустимых решений. Тогда ограничение (2.4) эквивалентно следующим условиям:

$$g(x) \geq yg^{min}, \quad h(x) \geq (1 - y)h^{min}, \quad y \in \{0, 1\}.$$

Действительно, если  $y = 0$ , то  $g(x) \geq 0$ ,  $h(x) \geq h^{min}$ , а в случае, когда  $y = 1$ , получаем  $g(x) \geq g^{min}$ ,  $h(x) \geq 0$ .

Аналогично, если необходимо выполнение любых  $k$  из  $m$  альтернативных условий

$$g_i(x) \geq 0, \quad i \in \{1, 2, \dots, m\},$$

то с помощью введения  $m$  булевых переменных  $y_i$  получаем эквивалентную систему ограничений:

$$\begin{aligned} g_i(x) &\geq y_i g_i^{min}, \quad i \in \{1, 2, \dots, m\} \\ \sum_{i=1}^m y_i &\leq m - k, \quad y_i \in \{0, 1\}, \quad i \in \{1, 2, \dots, m\}. \end{aligned}$$

Также часто в задачах оптимизации возникают ограничения вида

$$(2.5) \quad x_i \geq a_i \text{ либо } x_i = 0.$$

Например, заказ  $i$  либо не выполняется, либо обрабатывается целая партия, содержащая не менее  $a_i$  однотипных заказов. Пусть известно, что в оптимальном решении количество заказов заведомо меньше  $x_i^{max}$ . Тогда ограничение (2.5) запишется как

$$x_i \leq y_i x_i^{max}, \quad x_i \geq y_i a_i, \quad y_i \in \{0, 1\}.$$

Таким образом, введение булевых переменных позволяет свести задачи оптимизации с логическими ограничениями к задачам дискретной оптимизации.

## 2.2.2 Формулировка задачи формирования составов и расписания их движения

Пусть имеется граф железнодорожной сети  $G = (N, E)$ , где  $N$  — множество сортировочных станций,  $E$  — множество ребер (дорог). Станции соединены двухпутными железными дорогами, так что ребра  $(u, v)$  и  $(v, u)$  будем считать различными.

Каждая станция  $i \in N$  содержит заказы (вагоны) для доставки на другие станции. Введём следующие обозначения:

- $n_{ij}$  — количество заказов для перевозки со станции  $i$  в  $j$ ;
- $sort_i$  — максимальное количество составов, которые могут быть одновременно обслужены на станции  $i$ .
- $J_{ijk}$  —  $k$ -й заказ, направляемый из  $i$  в  $j$ ;
- $r_{ijk}$  — момент поступления заказа  $J_{ijk}$  на станцию отправления;
- $d_{ijk}$  — директивный срок заказа  $J_{ijk}$ ;
- $w_{ijk}$  — вес (значимость) заказа  $J_{ijk}$ ;
- $m_{ijk}$  — масса заказа  $J_{ijk}$ ;
- $M$  — грузоподъемность поезда (максимальная суммарная масса вагонов);
- $L$  — максимальная длина состава.

Для постановки задачи дискретизируем шкалу времени с некоторым шагом. Примем длительность шага равной единице, например, одному часу. Обозначим за  $T$  множество моментов времени  $\{0, 1, \dots, H\}$ , где  $H$  — достаточно большое число (верхняя оценка момента времени, к которому могут быть доставлены все заказы). Пусть время движения по ребру  $(i, j)$  составляет  $p_{ij}$  единиц. Кроме того, задан интервал движения поездов на каждом ребре  $(i, j)$ , равный  $\delta_{ij}$  (иными словами, после выхода поезда со станции  $i$  следующий поезд может отправиться только через  $\delta_{ij}$  единиц времени).

Также заданы множества временных интервалов недоступности  $\tau_{ij}$ , когда пути  $(i, j)$  заняты движением пассажирского транспорта, ремонтными работами и т.д.

Требуется определить расписание и маршрут движения каждого заказа со станции отправления на станцию назначения так, чтобы минимизировать суммарное взвешенное время выполнения всех заказов.

### 2.2.3 Задача формирования составов без ограничений на количество локомотивов

#### Переменные задачи

Введём следующие переменные. Переменная  $x_{ijk(u,v)t}$  принимает значение 1, если заказ  $J_{ijk}$  начинает движение по ребру  $(u, v)$ ,  $u \neq v$ , в момент времени  $t$ , и 0 в противном случае.

Бинарная переменная  $y_{(u,v)t}$  равна единице, если в момент времени  $t$  из  $u$  в  $v$  отправляется поезд, и нулю в противном случае.

Связь переменных  $x$  и  $y$  задаётся следующим образом:

$$(2.6) \quad y_{(u,v)t} \geq \frac{1}{L} \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} x_{ijk(u,v)t} \quad \forall t \in T, \forall (u, v) \in E.$$

Опишем основные ограничения задачи.

#### Ограничения задачи

Первая группа ограничений обеспечивает движение заказов по железнодорожной сети и представляет собой стандартные требования для задач маршрутизации.

#### Ограничения на передвижение по графу дорог

Заказ не может быть отправлен раньше времени его поступления на станцию:

$$(2.7) \quad \sum_{t \in T} \sum_{(i,u) \in E} x_{ijk(i,u)t} \cdot t \geq r_{ijk} \quad \forall i, j \in N, \forall k \in \{1, 2, \dots, n_{ij}\}.$$

Заказ может войти в вершину не более одного раза:

$$(2.8) \quad \sum_{t \in T} \sum_{(u,v) \in E} x_{ijk(u,v)t} \leq 1 \quad \forall i, j \in N, \quad \forall k \in \{1, 2, \dots, n_{ij}\}, \quad \forall v \in N.$$

Заказ может выйти из вершины не более одного раза:

$$(2.9) \quad \sum_{t \in T} \sum_{(u,v) \in E} x_{ijk(u,v)t} \leq 1 \quad \forall i, j \in N, \quad \forall k \in \{1, 2, \dots, n_{ij}\}, \quad \forall u \in N.$$

Все заказы должны быть доставлены:

$$(2.10) \quad \sum_{t \in T} \sum_{(u,j) \in E} \sum_{k \in \{1, 2, \dots, n_{ij}\}} x_{ijk(u,j)t} = n_{ij} \quad \forall i, j \in N.$$

Если заказ пришел на промежуточную станцию, он должен из неё выйти:

$$(2.11) \quad \sum_{t \in T} \sum_{(u,v) \in E} x_{ijk(u,v)t} = \sum_{t \in T} \sum_{(v,w) \in E} x_{ijk(v,w)t}$$

$$\forall i, j \in N, \quad \forall k \in \{1, 2, \dots, n_{ij}\}, \quad \forall v \in N, \quad v \neq i, v \neq j,$$

при этом заказ начинает прохождение следующего ребра не раньше чем через время, требуемое на прохождение предыдущего:

$$(2.12) \quad \sum_{t \in T} \sum_{(u,v) \in E} x_{ijk(u,v)t} \cdot (t + p_{uv}) \leq \sum_{t \in T} \sum_{(v,w) \in E} x_{ijk(v,w)t} \cdot t$$

$$\forall i, j \in N, \quad \forall k \in \{1, 2, \dots, n_{ij}\}, \quad \forall v \in N, \quad v \neq i, v \neq j.$$

Далее идут требования, учитывающие специфику железнодорожного транспорта.

### Ограничения по составам

Ограничение по длине состава (одновременно по ребру может быть отправлено не более  $L$  вагонов):

$$(2.13) \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} x_{ijk(u,v)t} \leq L \quad \forall t \in T, \forall (u, v) \in E.$$

Ограничение по массе состава:

$$(2.14) \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} m_{ijk} \cdot x_{ijk(u,v)t} \leq M \quad \forall t \in T, \forall (u, v) \in E.$$

### Ограничения по сортировочным станциям

На сортировочной станции  $v$  не может одновременно обслуживаться более  $sort_v$  составов:

$$(2.15) \quad \sum_{i \in N} \sum_{j \in N} \sum_{(u,v) \in E} y_{(u,v)t} + \sum_{i \in N} \sum_{j \in N} \sum_{(v,w) \in E} y_{(v,w)t} \leq sort_v \quad \forall t \in T, \forall v \in N.$$

Величина  $sort_v$  зависит от продуктивности станции.

### Ограничения по путям

Поезда отправляются по ребру  $(u, v)$  с интервалом  $\delta_{uv}$ :

$$(2.16) \quad y_{(u,v)t_1} + y_{(u,v)t_2} \leq 1 \quad \forall (u, v) \in E, \forall t_1, t_2 \in T, t_1 < t_2 < t_1 + \delta_{uv}.$$

Данное ограничение моделирует ситуацию движения по так называемым блок-участкам. Пути между двумя соседними сортировочными станциями разделены семафорами на участки, на которых одновременно может находиться не более одного поезда. В этом случае  $\delta_{uv}$  — это время прохождения самого длинного блок-участка между станциями  $u$  и  $v$ .

Движение в запрещённые моменты времени невозможно:

$$(2.17) \quad y_{(u,v)t} = 0 \quad \forall (u, v) \in E, \forall t \in \tau_{uv}.$$

На практике данные ограничения возникают в случае временного закрытия тех или иных путей на ремонт, выделения пути для движения

скорых поездов и т.д.

### Целевая функция

Несложно видеть, что в этом случае время прибытия заказа  $J_{ijk}$  на станцию назначения определяется как

$$(2.18) \quad C_{ijk} = \sum_{t \in T} \sum_{(u,j) \in E} x_{ijk(u,j)t} \cdot (t + p_{uj}).$$

Если предположить, что затраты пропорциональны суммарному расстоянию, пройденному всеми поездами, осуществляющими перевозки вагонов, то суммарные издержки составят

$$(2.19) \quad \sum_{t \in T} \sum_{(u,v) \in E} y_{(u,v)t} \cdot p_{uv}.$$

В таком случае целевой функцией задачи может выступить

$$(2.20) \quad \min F_1(x, y) = \varphi(C) + \sum_{t \in T} \sum_{(u,v) \in E} y_{(u,v)t} \cdot p_{uv},$$

где  $\varphi(C)$  — функция, характеризующая сроки доставки заказов. В случае минимизации взвешенного суммарного времени окончания работ

$$(2.21) \quad \varphi(C) = \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} w_{ijk} C_{ijk},$$

которое характеризует среднее время нахождения заказа в пути, мы получаем линейную целевую функцию, а значит, задачу целочисленного линейного программирования (ЦЛП).

### Возможные модификации задачи

Отметим, что решение в представленной задаче с ограничениями (2.6)–(2.17) существует только при выборе достаточно большого интервала планирования  $T$ . В противном случае условие доставки всех заказов



(2.10) может быть не выполнено. В этом случае целесообразно ставить задачу ЦЛП без жёстких ограничений (2.10), (2.11) с целевой функцией (2.22)

$$\max F_2(x, y) = \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} \sum_{t \in T} \sum_{(u, j) \in E} w_{ijk} x_{ijk(u, j)t} - \sum_{t \in T} \sum_{(u, v) \in E} y_{(u, v)t} \cdot p_{uv},$$

максимизирующей взвешенное число доставленных заказов с учетом транспортных издержек.

В случае фиксированного расписания движения поездов можно заранее задать значения переменных  $y_{(u, v)t}$ . В этом случае задача (2.6)–(2.17), (2.20) будет содержать только переменные  $x_{ijk(u, v)t}$  и заключаться в определении, по какому маршруту (какими поездами) должен быть отправлен заказ. В целевой функции при этом не будут учитываться издержки (2.19), а ограничения (2.13) могут быть модифицированы следующим образом:

$$(2.23) \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in \{1, 2, \dots, n_{ij}\}} x_{ijk(u, v)t} \leq L_{(u, v)t} \quad \forall t \in T, \forall (u, v) \in E.$$

Это целесообразно для случая, когда к поездам, идущим по расписанию, уже прикреплены какие-либо другие заказы.

#### 2.2.4 Случай ограниченного числа локомотивов

В описанной выше постановке предполагалось наличие неограниченного количества локомотивов. Однако на практике часто возникает ситуация невозможности сформировать состав в некоторый момент времени ввиду нехватки локомотивов.

Далее будем рассматривать случаи, когда имеются ограничения на количество или расположение вагонов.

### Случай заранее заданного начального расположения локомотивов

Введем целочисленную переменную  $z_{it}$ ,  $i \in N$ ,  $t \in T$ , равную количеству локомотивов на станции  $i$  в момент времени  $t$ ,  $z \in Z_+$ . Пусть величины  $z_{i0}$ ,  $i \in N$ , заранее известны (начальное расположение локомотивов на сети железных дорог). Тогда количество локомотивов в произвольный момент времени  $t$  на станции  $v$  определяется следующим образом:

$$(2.24) \quad z_{vt} = z_{v0} - \sum_{0 \leq \tau < t} \sum_{(v,u) \in E} y_{(v,u)\tau} + \sum_{(v,u) \in E} \sum_{0 \leq \tau \leq t - p_{uv}} y_{(u,v)\tau} \quad \forall t \in T, \quad \forall v \in N.$$

В результате получаем ограничение на количество формируемых на станции составов в зависимости от имеющихся в наличии локомотивов:

$$(2.25) \quad \sum_{(u,v) \in E} y_{(u,v)t} \leq z_{ut} \quad \forall t \in T.$$

### Случай фиксированного количества локомотивов

Если величины  $z_{i0}$ ,  $i \in N$ , заранее не известны, а имеется лишь ограничение на общее количество локомотивов, то к ограничениям (2.24)–(2.25) добавляется неравенство

$$(2.26) \quad \sum_{i \in N} z_{i0} \leq Loc,$$

где  $Loc$  — количество имеющихся в наличии локомотивов.

Таким образом, в зависимости от возникающей на практике ситуации, задача формирования составов и расписания их движения может формулироваться в разных вариантах: можно фиксировать или оставлять произвольными число локомотивов ( $z_{it}$ ), маршруты и расписания движения поездов ( $y_{(u,v)t}$ ), изменять целевые функции и т.д. Во всех случаях задача представляет собой задачу целочисленного линейного программирования, для решения которой существует ряд точных и аппроксимационных методов [2, 66]. Эффективность решения данной задачи зависит

от анализа структуры множества ограничений и выбора подходящего метода решения.

## 2.3 Модели задач железнодорожного планирования, использующие пространственно-временные графы

В данном варианте проблемы предполагается, что маршруты движения грузовых составов, а также расписание их движения фиксированы. Необходимо назначить вагоны с грузом на поезда, тем самым определив маршрут их движения. В ходе своего движения вагоны могут отцепляться от состава и прицепляться к другому составу на сортировочных станциях. Железнодорожная сеть задана произвольным графом  $G$  со множеством вершин  $N$ , соответствующим сортировочным станциям, и множеством рёбер  $E$ , соответствующим железнодорожным путям. Рассмотрим случай, когда имеется не более одного заказа на перевозку груза от станции  $i \in N$  до станции  $j \in N$ , и что весь груз доступен для транспортировки в нулевой момент времени, т.е. каждый заказ однозначно определяется парой  $(i, j)$ . Множество заказов будем обозначать как  $O$ . Пусть  $n_{ij}$  — количество вагонов, направляемых из  $i$  в  $j$ , а  $m_{ij}$  и  $l_{ij}$  — масса и длина каждого вагона используемого для выполнения заказа  $(i, j)$ . Пусть также  $w_{ij}$  — вес (важность) заказа  $(i, j)$ , и  $d_{ij}$  — его директивный срок.

Пусть  $Q$  — множество поездов, циркулирующих по сети  $G$ . Пусть  $Q_i^{\text{in}}$  и  $Q_i^{\text{out}}$  — множество поездов, прибывающих и отбывающих со станции  $i \in N$ . Множества  $Q_i^{\text{in}}$  и  $Q_i^{\text{out}}$  могут быть различными, так как станция  $i$  может быть начальной или конечной для некоторых составов. Пусть  $t_{qi}^{\text{in}}$ ,  $q \in Q_i^{\text{in}}$ , и  $t_{qi}^{\text{out}}$ ,  $q \in Q_i^{\text{out}}$ , — время прибытия и отбытия со станции  $i$  поезда  $q$ . Предположим, что все поезда могут перевозить любые вагоны. Обозначим через  $\Delta$  максимальное время, необходимое, чтобы прицепить

или отцепить вагоны от поезда.  $M$  и  $L$  задают, соответственно, максимальный вес (массу) и длину одного состава.

Для постановки задачи мы будем использовать так называемый **пространственно-временной направленный граф** (*space-time network*, [39])  $G' = (V', A')$ . В таком графе каждая вершина определяется временем, местом и поездом.

Вершины данного графа делятся на три типа:

- **вершины отбытия:** каждая вершина  $v_{it}^{\text{out}}$  соответствует отбытию со станции  $i \in N$  некоторого поезда  $q \in Q$  в момент  $t = t_{qi}^{\text{out}}$ ;
- **вершины прибытия:** каждая вершина  $v_{it}^{\text{in}}$  соответствует прибытию на станцию  $i \in N$  некоторого поезда  $q \in Q$  в момент  $t = t_{qi}^{\text{in}}$ ;
- **вершины сортировки:** каждая вершина  $v_{it}^{\text{srt}}$  соответствует окончанию отцепки вагонов от пришедшего поезда ( $t = t_{qi}^{\text{in}} + \Delta$ ,  $q \in Q$ ) или началу прицепки вагонов к уходящему поезду ( $t = t_{qi}^{\text{out}} - \Delta$ ,  $q \in Q$ ) на станции  $i \in N$ .

Рёбра графа  $G'$  делятся на пять типов:

- **рёбра движения поезда** ( $A^{\text{mov}}$ ): каждое ребро  $a_{qij}^{\text{mov}} = (v_{it'}^{\text{out}}, v_{jt''}^{\text{in}})$  соответствует движению поезда  $q \in Q$  со станции  $i$  на следующую на его маршруте станцию  $j$ , здесь  $t' = t_{qi}^{\text{out}}$  и  $t'' = t_{qj}^{\text{in}}$ ;
- **рёбра стоянки поезда** ( $A^{\text{tr}}$ ): каждое ребро  $a_{qi}^{\text{tr}} = (v_{it'}^{\text{in}}, v_{it''}^{\text{out}})$  соответствует стоянке поезда  $q \in Q$  на промежуточной на его маршруте станции  $i$ , здесь  $t' = t_{qi}^{\text{in}}$  и  $t'' = t_{qi}^{\text{out}}$ ;
- **рёбра отцепки вагонов** ( $A^{\text{dwn}}$ ): каждое ребро  $a_{qi}^{\text{dwn}} = (v_{it}^{\text{in}}, v_{i,t+\Delta}^{\text{srt}})$  соответствует отцепке вагонов от поезда  $q \in Q$  на находящейся на его маршруте станции  $i$ , здесь  $t = t_{qi}^{\text{in}}$ ;
- **рёбра прицепки вагонов** ( $A^{\text{up}}$ ): каждое ребро  $a_{qi}^{\text{up}} = (v_{i,t-\Delta}^{\text{srt}}, v_{it}^{\text{out}})$  соответствует прицепке вагонов к поезду  $q \in Q$  на находящейся на его маршруте станции  $i$ , здесь  $t = t_{qi}^{\text{out}}$ ;

- **рёбра сортировки** ( $A^{\text{srt}}$ ): каждое ребро  $a_{it't''}^{\text{srt}} = (v_{it'}^{\text{srt}}, v_{it''}^{\text{srt}})$  между соседними вершинами сортировки станции  $i$  (вершины сортировки  $v_{it}^{\text{srt}}$  упорядочены по возрастанию моментов времени  $t$ ).

Для иллюстрации на рис. 2.5 представлен пример графа железнодорожной сети  $G$ . В данном примере станция  $i$  является конечной для поезда  $q_1$ , промежуточной для поезда  $q_2$  и начальной для поезда  $q_3$ . На рис. 2.6 показана часть пространственно-временного графа  $G'$ , соответствующая станции  $i$  графа  $G$ .

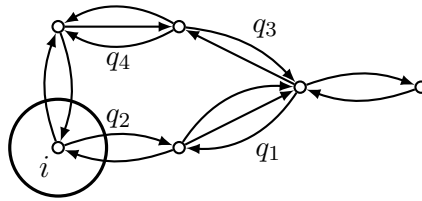


Рис. 2.5. Пример графа железнодорожной сети

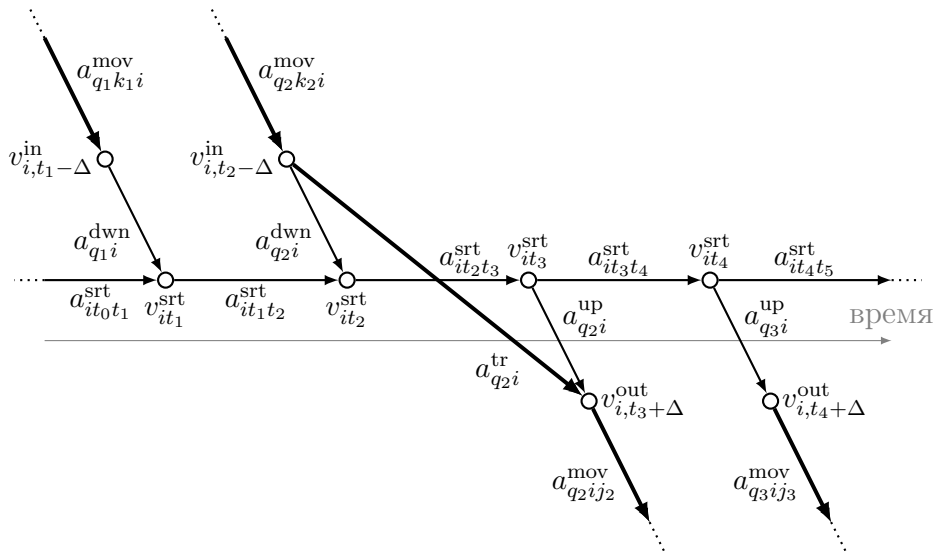


Рис. 2.6. Пример части пространственно-временного графа (станция  $i$ )

Маршрут движения каждого вагона из заказа однозначно определяется соответствующим путём в пространственно-временном графе. Например, частичный путь

$$(v_{i,t_1-\Delta}^{\text{in}}, v_{it_1}^{\text{srt}}, v_{it_2}^{\text{srt}}, v_{it_3}^{\text{srt}}, v_{i,t_3+\Delta}^{\text{out}})$$

на рис. 2.6 соответствует маршруту, по которому вагон прибывает на станцию  $i$  в составе поезда  $q_1$ , перецепляется и отбывает со станции в составе поезда  $q_2$ . Путь вагона из заказа  $(i, j) \in O$  начинается в вершине  $v_{i0}^{\text{srt}}$  и заканчивается в вершине  $v_{jH}^{\text{srt}}$ , где  $H$  — длина горизонта планирования (максимальное время прибытия поезда на его конечную станцию). Движение всех вагонов заказа  $(i, j) \in O$  соответствует потоку величины  $n_{ij}$  от источника  $v_{i0}^{\text{srt}}$  до стока  $v_{jH}^{\text{srt}}$  в пространственно-временном графе  $G'$ . Очевидно, что поток должен быть целочисленным, т.е. величина потока на каждом ребре должна быть некоторым положительным целым числом.

Пусть целочисленная переменная  $f_{ij}^a$  определяет величину потока  $(i, j) \in O$  по ребру  $a \in A'$ , т.е. количество вагонов “перемещенных” по ребру  $a$ . Тип перемещения (движение, стоянка, сцепка, отцепка) зависит от типа ребра. Пусть  $\delta_v^-$  и  $\delta_v^+$  — множество входящих и исходящих рёбер для вершины  $v \in V'$ . Следующие ограничения задают множество потоков, соответствующих движению вагонов:

$$(2.27) \quad \sum_{a \in \delta_v^+} f_{ij}^a = n_{ij}, \quad \text{где } v = v_{i0}^{\text{srt}}, \quad \forall (i, j) \in O,$$

$$(2.28) \quad \sum_{a \in \delta_v^-} f_{ij}^a = \sum_{a \in \delta_v^+} f_{ij}^a, \quad \forall (i, j) \in O, \quad \forall v \neq v_{i0}^{\text{srt}}, v \neq v_{jH}^{\text{srt}},$$

$$(2.29) \quad \sum_{a \in \delta_v^-} f_{ij}^a = n_{ij}, \quad \text{где } v = v_{jT}^{\text{srt}}, \quad \forall (i, j) \in O.$$

Ограничения (2.27) задают количество вагонов заказа  $(i, j)$ , находящихся на станции  $i$  в начальный момент времени. Ограничения (2.29) задают количество вагонов заказа  $(i, j)$ , находящихся на станции  $j$  в конечный момент времени. Условия (2.28) являются ограничениями сохранения потока.

Ограничения на максимальный вес и длину состава задаются следу-

ющим образом:

$$(2.30) \quad \sum_{(i,j) \in O} l_{ij} \cdot f_{ij}^a \leq L, \quad \forall a \in A^{\text{mov}},$$

$$(2.31) \quad \sum_{(i,j) \in O} m_{ij} \cdot f_{ij}^a \leq M, \quad \forall a \in A^{\text{mov}}.$$

Для балансировки загрузки станций вводятся ограничения на максимальное число обслуженных вагонов (сцепок и расцепок) в течение некоторого периода времени. Пусть  $P$  — множество интервалов времени, в течение которых на каждой станции может быть обслужено не более  $W$  вагонов. Ограничения балансировки гарантируют, что для каждого интервала времени из  $P$  и каждой станции  $k \in N$  величина суммарного потока по рёбрам отцепки и прицепки на станции  $k$ , соответствующим прибывшим и отбывшим поездам в течение этого интервала, не превышает  $W$ :

$$(2.32) \quad \sum_{(i,j) \in O} \sum_{\substack{\forall a = a_{qk}^{\text{down}}: \\ t_{qk}^{\text{in}} \in (t', t'']}} f_{ij}^a + \sum_{(i,j) \in O} \sum_{\substack{\forall a = a_{qk}^{\text{up}}: \\ t_{qk}^{\text{out}} \in (t', t'']}} f_{ij}^a \leq W, \\ \forall k \in N, \quad \forall (t', t''] \in P.$$

Для формулировки целевой функции введем бинарные переменные  $z$ . Переменная  $z_{ijt}$ ,  $(i, j) \in O$ ,  $0 \leq t \leq H$  принимает значение 0 тогда и только тогда, когда все вагоны заказа  $(i, j)$  доставлены на станцию назначения к моменту времени  $t$ . Пусть  $I_i$  — множество интервалов времени  $(t, t')$ , соответствующих рёбрам  $a_{itt'}^{\text{srt}}$ . Ограничения, связывающие переменные  $z$  и  $f$  определяются как

$$(2.33) \quad z_{ijt} \geq 1 - \frac{1}{n_{ij}} f_{ij}^a,$$

где

$$a = a_{jtt'}^{\text{srt}}, \quad \forall (i, j) \in O, \quad \forall (t, t') \in I_j : t' > d_{ij}.$$

Используя переменные  $z$ , целевая функция минимизации взвешенного

суммарного запаздывания может быть записана как

$$(2.34) \quad \min \sum_{(i,j) \in O} \sum_{\substack{(t,t') \in I_j: \\ t' > d_{ij}}} w_{ij} \cdot (t' - \max\{t, d_{ij}\}) \cdot z_{ijt}.$$

### 2.3.1 Задача с гибким расписанием движения поездов при фиксированных маршрутах

В данном варианте проблемы также предполагается, что маршруты движения грузовых составов фиксированы. Отличие от предыдущего варианта задачи заключается в том, что расписание поездов не задано, т.е. в дополнение к назначению грузовых вагонов на поезда, нам необходимо для каждого поезда определить его расписание (времена отбытия и прибытия на станции).

#### Первая версия постановки задачи

Для постановки данной задачи, мы дискретизируем шкалу времени с некоторым шагом (например, 1 час). Величина шага влияет на точность модели: чем меньше шаг, тем ближе постановка задачи к реальности. Примем длительность шага равным единице. Обозначим через  $T$  множество моментов времени:  $T = \{0, 1, \dots, H\}$ .

Данный вариант проблемы характеризуется наличием дополнительных данных. Пусть  $p_{ijt}$  — длительность переезда поезда от станции  $i \in N$  к станции  $j \in N$  при начале движения в момент времени  $t \in T$ . Пусть также  $\gamma_{ijt}$  — пропускная способность железной дороги в единицу времени от станции  $i \in N$  к станции  $j \in N$  при начале движения в момент времени  $t \in T$ . Заметим, что длительность переезда и пропускная способность зависят от момента начала движения в связи с переменной загруженностью железной дороги пассажирскими поездами и ремонтными работами. Будем предполагать, что грузовые поезда могут обгонять друг друга только на станциях.



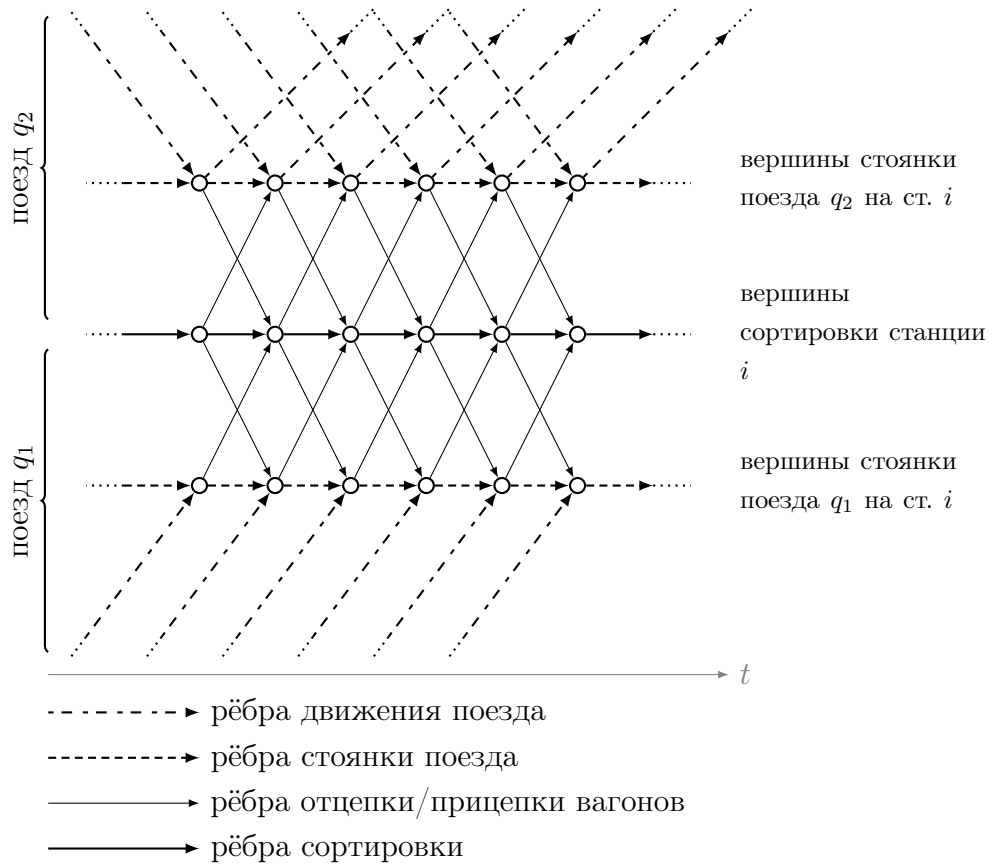
Для решения будет использоваться измененный и увеличенный пространственно-временной граф  $\bar{G} = (\bar{V}, \bar{A})$ . Вершины стоянки поезда  $v_{qit}^{\text{tr}}$  определены для каждой станции  $i \in N$ , каждого поезда  $q \in Q_i^{\text{in}} \cup Q_i^{\text{out}}$  имеющего станцию  $i$  в своем маршруте и каждого момента времени  $t \in T$ . Вершины сортировки  $v_{it}^{\text{srt}}$  определены для каждой станции  $i \in N$  и каждого момента времени  $t \in T$ . Мы обозначим множество вершин из  $\bar{V}^{\text{tr}}$ , соответствующих поезду  $q \in Q$  как  $\bar{V}_q$ .

Имеются следующие типы рёбер:

- **рёбра движения поезда** ( $\bar{A}^{\text{mov}}$ ): каждое ребро  $a_{qijt}^{\text{mov}} = (v_{iq}^{\text{tr}}, v_{jq}^{\text{tr}})$  соответствует движению поезда  $q \in Q$  по его маршруту со станции  $i \in N$  на станцию  $j \in N$  начиная с момента времени  $t \in T$ , здесь  $t' = t + p_{ijt}$ ;
- **рёбра стоянки поезда** ( $\bar{A}^{\text{tr}}$ ): каждое ребро  $a_{qit}^{\text{tr}} = (v_{i,t-1}^{\text{tr}}, v_{it}^{\text{tr}})$  соответствует стоянке поезда  $q \in Q$  (не в режиме сортировки) на находящейся на его маршруте станции  $i \in N$  в момент времени  $t \in T$ ;
- **рёбра отцепки вагонов** ( $\bar{A}^{\text{dwn}}$ ): каждое ребро  $a_{qit}^{\text{dwn}} = (v_{qit}^{\text{tr}}, v_{t+\Delta}^{\text{srt}})$  соответствует отцепке вагонов от поезда  $q \in Q$  на находящейся на его маршруте станции  $i \in N$  в момент времени  $t \in T$ ;
- **рёбра прицепки вагонов** ( $\bar{A}^{\text{up}}$ ): каждое ребро  $a_{qit}^{\text{up}} = (v_{i,t-\Delta}^{\text{srt}}, v_{qit}^{\text{tr}})$  соответствует прицепке вагонов к поезду  $q \in Q$  на находящейся на его маршруте станции  $i \in N$  в момент времени  $t \in T$ ;
- **рёбра сортировки** ( $\bar{A}^{\text{srt}}$ ): каждое ребро  $a_{it}^{\text{srt}} = (v_{i,t-1}^{\text{srt}}, v_{it}^{\text{srt}})$  соответствует нахождению поезда или вагона в состоянии сортировки на станции  $i \in N$  в момент времени  $t \in T$ .

Обозначим множество рёбер из  $\bar{A}^{\text{mov}} \cup \bar{A}^{\text{tr}} \cup \bar{A}^{\text{dwn}} \cup \bar{A}^{\text{up}}$ , соответствующих поезду  $q \in Q$ , как  $\bar{A}_q$ .

На рис. 2.7 представлена часть увеличенного пространственно-временного графа  $\bar{G}$ , соответствующая станции  $i$ . В данном примере станция  $i$  является конечной для поезда  $q_1$  и промежуточной для поезда  $q_2$ .



**Рис. 2.7. Пример части увеличенного пространственно-временного графа  $\bar{G}$**

Расписание движения каждого поезда  $q \in Q$  однозначно определяется путём в графе  $\bar{G}$ , причем данный путь может проходить только по рёбрам, соответствующим данному поезду  $q$ , а также по рёбрам сортировки. Пусть начальной и конечной станцией маршрута поезда  $q \in Q$  являются  $b(q)$  и  $e(q)$ . Тогда путь, соответствующий расписанию поезда  $q$ , начинается в вершине  $v_{q,b(q),0}^{\text{tr}}$  и заканчивается в вершине  $v_{q,e(q),H}^{\text{tr}}$ .

Пусть бинарная переменная  $g_q^a$  определяет, проходит ли путь соответствующий расписанию поезда  $q \in Q$  по ребру  $a \in \bar{A}_q \cup \bar{A}^{\text{srt}}$  или нет. Пусть  $\delta_v^{q-}$  и  $\delta_v^{q+}$  — множество входящих и исходящих рёбер для вершины  $v \in \bar{V}$ , по которым может проходить путь поезда  $q \in Q$ . Следующие ограничения задают множество путей, соответствующих движению поездов:

$$(2.35) \quad \sum_{a \in \delta_v^{q+}} g_q^a = 1, \quad \text{где } v = v_{q,b(q),0}^{\text{tr}}, \forall q \in Q,$$

$$(2.36) \quad \sum_{a \in \delta_v^{q-}} g_q^a = \sum_{a \in \delta_v^{q+}} g_q^a, \quad \forall q \in Q, \forall v \in \bar{V}^{\text{srt}} \cup \bar{V}_q,$$

$$(2.37) \quad \sum_{a \in \delta_v^{q-}} g_q^a = 1, \quad \text{где } v = v_{q,e(q),H}^{\text{tr}}, \forall q \in Q.$$

Количество поездов, проходящих по одному и тому же пути в течение единицы времени, не может превышать пропускную способность этого пути в данный момент времени:

$$(2.38) \quad \sum_{\substack{q \in Q: \\ \exists a = a_{qijt}^{\text{mov}}}} g_q^a \leq \gamma_{ijt}, \quad \forall t \in T, \quad \forall (i, j) \in E.$$

Как и в предыдущем варианте задачи, движение всех вагонов заказа  $(i, j) \in O$  соответствует потоку величины  $n_{ij}$  от источника  $v_{i0}^{\text{srt}}$  до стока  $v_{jH}^{\text{srt}}$  в графе  $\bar{G}$ . Мы продолжим использовать переменные  $f_{ij}^a$  для определения величины потока  $(i, j) \in O$  по ребру  $a \in \bar{A}$ . Ограничения (2.27)–(2.29) задают множество потоков, соответствующих движению вагонов.

Заметим, что теперь поток любого заказа может проходить по ребру  $a \in \bar{A}_q$  только тогда, когда по данному ребру проходит путь соответствующий поезду  $q$ :

$$(2.39) \quad f_{ij}^a \leq n_{ij} \cdot g_q^a, \quad \forall (i, j) \in O, \quad \forall q \in Q, \quad \forall a \in \bar{A}_q \setminus \bar{A}^{\text{mov}}.$$

В связи с данным замечанием, ограничения (2.30)–(2.31) на макси-

мальные длину и вес состава меняются на:

$$(2.40) \quad \sum_{(i,j) \in O} l_{ij} \cdot f_{ij}^a \leq L \cdot g_q^a, \quad \forall q \in Q, \quad \forall a \in \bar{A}_q^{\text{mov}},$$

$$(2.41) \quad \sum_{(i,j) \in O} m_{ij} \cdot f_{ij}^a \leq M \cdot g_q^a, \quad \forall q \in Q, \quad a \in \bar{A}_q^{\text{mov}}.$$

При использовании увеличенного графа  $\bar{G}$ , балансировка загрузки сортировочных станций может осуществляться введением ограничений как на число обслуженных вагонов, так и на число обслуженных поездов в течение некоторого периода времени. В первом варианте ограничения практически не отличаются от (2.32). Рассмотрим теперь второй вариант. Пусть  $\bar{P}$  — множество интервалов времени, в течение которых на каждой станции может быть обслужено не более  $U$  поездов. Ограничения балансировки гарантируют, что для каждого интервала времени из  $\bar{P}$  и каждой станции  $i \in N$  количество поездов, “прошедших” по рёбрам отцепки и прицепки станции  $i$  не превышает  $U$ :

$$(2.42) \quad \sum_{q \in Q} \sum_{\substack{\forall a = a_{qit}^{\text{down}}: \\ t \in (t', t'']}} g_q^a + \sum_{q \in Q} \sum_{\substack{\forall a = a_{qit}^{\text{up}}: \\ t \in (t', t'']}} g_q^a \leq U, \quad \forall i \in N, \quad \forall (t', t''] \in \bar{P}.$$

Для формулировки целевой функции также используются бинарные переменные  $z_{ijt}$ ,  $(i, j) \in O$ ,  $t \in T$ . Ограничения, связывающие переменные  $z$  и  $f$ :

$$(2.43) \quad z_{ijt} \geq 1 - \frac{1}{n_{ij}} f_{ij}^a,$$

где

$$a = a_{j,t+1}^{\text{srt}}, \quad \forall (i, j) \in O, \quad \forall t > d_{ij}.$$

Целевая функция в данном варианте выглядит следующим образом:

$$(2.44) \quad \min \sum_{(i,j) \in O} \sum_{t=d_{ij}+1}^H w_{ij} \cdot z_{ijt}.$$

### Упрощенная модель

В данной версии постановки мы пренебрежем пропускной способностью путей и допустим, что время движения поезда между двумя станциями является константой и не зависит от времени начала движения. Также мы допустим, что веса и длины всех вагонов одинаковы, что приводит к преобразованию ограничений на максимальные вес и длину состава в ограничения на максимальное число вагонов в составе. Обозначим данное число как  $n_{\max}$ .

Для данной постановки мы будем использовать упрощенный направленный граф  $\tilde{G} = (\tilde{V}, \tilde{A})$ . Имеется два типа вершин — прибытия и отбытия. Вершина прибытия  $v_{qi}^{\text{in}}$  (отбытия  $v_{qi}^{\text{out}}$ ) соответствует прибытию (отбытию) поезда  $q \in Q$  на станцию (со станции)  $i \in N$ . Имеется также два типа рёбер — поезда и сортировки. Ребро поезда  $a_{qij}^{\text{tr}} = (v_{qi}^{\text{out}}, v_{qj}^{\text{in}})$  или  $a_{qii}^{\text{tr}} = (v_{qi}^{\text{in}}, v_{qi}^{\text{out}})$  соответствует движению поезда  $q \in Q$  между станциями  $i, j \in N$ , или стоянке поезда  $q$  на станции  $i$ , если  $i = j$ . Ребро сортировки  $a_{qsi}^{\text{srt}} = (v_{qi}^{\text{in}}, v_{si}^{\text{out}})$  соответствует перецепке вагонов между поездами  $q, s \in N$ ,  $q \neq s$ , на станции  $i \in N$ .

На рис. 2.8 представлена часть графа  $\tilde{G}$ , соответствующая станции  $i$  (см. рис. 2.5). В данном примере станция  $i$  является конечной для поезда  $q_1$ , промежуточной для поезда  $q_2$  и начальной для поезда  $q_3$ .

Пусть бинарная переменная  $x_{qsi}$ ,  $i \in N$ ,  $q \in Q_i^{\text{in}}$ ,  $s \in Q_i^{\text{out}}$ ,  $q \neq s$ , определяет, осуществляется ли перецепка вагонов между поездами  $q$  и  $s$  на станции  $i$ . Целочисленная переменная  $f_{ij}^a$  равняется величине потока заказа  $(i, j) \in O$ , проходящему по ребру  $a \in \tilde{A}$ . Переменные  $S_{qi}$ ,  $q \in Q_i^{\text{out}}$ , и  $C_{qi}$ ,  $q \in Q_i^{\text{in}}$ , определяют время отбытия и прибытия поезда  $q$

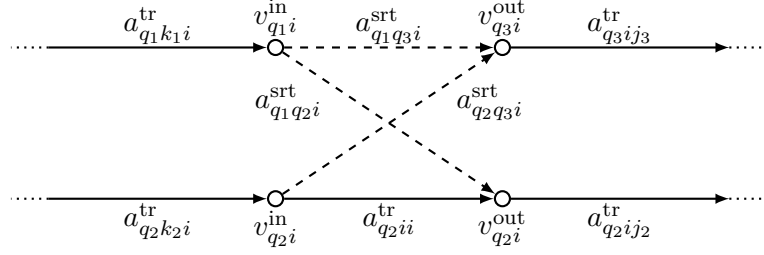


Рис. 2.8. Пример части графа  $\tilde{G}$  (станция  $i$ )

на станцию (со станции)  $i$ . Переменные  $T_{qij}$ ,  $(i, j) \in O$ ,  $q \in Q_j^{\text{in}}$ , определяют запаздывание поезда  $q$  по отношению к директивному сроку заказа  $(i, j) \in O$ .

$$(2.45) \quad \min \sum_{(i,j) \in O} \sum_{\substack{q \in Q_j^{\text{in}}, \\ a = a_{qj}^{\text{in}}}} \frac{w_{ij}}{n_{ij}} f_{ij}^a T_{qij}$$

$$(2.46) \quad T_{qij} \geq C_{qj} - d_{ij}, \quad \forall (i, j) \in O, \quad \forall q \in Q_j^{\text{in}},$$

$$(2.47) \quad T_{qij} \geq 0, \quad \forall (i, j) \in O, \quad \forall q \in Q_j^{\text{in}},$$

$$(2.48) \quad \sum_{q \in Q_i^{\text{out}}} f_{ij}^a = n_{ij}, \quad \text{где } a = a_{qi}^{\text{out}}, \quad \forall (i, j) \in O,$$

$$(2.49) \quad \sum_{a \in \delta_v^-} f_{ij}^a = \sum_{a \in \delta_v^+} f_{ij}^a, \quad \forall (i, j) \in O, \quad \forall v \neq v_{qi}^{\text{out}}, v \neq v_{qj}^{\text{in}},$$

$$(2.50) \quad \sum_{q \in Q_j^{\text{in}}} f_{ij}^a = n_{ij}, \quad \text{где } a = a_{qj}^{\text{in}}, \quad \forall (i, j) \in O,$$

$$(2.51) \quad \sum_{(i,j) \in O} f_{ij}^a \leq n_{\max} x_{qsk}, \quad \forall a = a_{qsk}^{\text{srt}},$$

$$(2.52) \quad \sum_{(i,j) \in O} f_{ij}^a \leq n_{\max}, \quad \forall a \in \tilde{A}^{\text{tr}},$$

$$(2.53) \quad \begin{aligned} S_{si} &\geq C_{qi} + \Delta + \mathcal{M}(1 - x_{qsi}), & \forall i \in N, \forall q \in Q_i^{\text{in}}, \\ & & s \in Q_i^{\text{out}}, q \neq s. \end{aligned}$$

Целевая функция (2.45) минимизирует суммарное среднее взвешенное запаздывание вагонов. Неравенства (2.46)–(2.47) определяют запаздывание каждого поезда на каждой станции относительно директивного

срока каждого заказа доставки груза на данную станцию. Множество потоков вагонов задается ограничениями (2.48)–(2.50). Неравенства (2.51)–(2.52) ограничивают сверху величину потока проходящего по каждому ребру, т.е. задают максимальное число вагонов в каждом составе. Ограничения (2.53) задают отношения предшествования между прибытием поезда  $q$  на станцию  $i$  и отбытием поезда  $s$  с этой станции в случае перцепки вагонов с  $q$  на  $s$ .

Недостатками формулировки (2.45)–(2.53) являются нелинейная целевая функция, а также наличие так называемых “*big-M*” ограничений (2.53), которые существенно ухудшают нижнюю оценку, получаемую релаксаций целочисленных ограничений. Поэтому данный вариант проблемы в упрощенной постановке может решаться методами, которые не используют стандартные алгоритмы целочисленного программирования. Также недостатком упрощенной модели является невозможность формулировки ограничений на балансировку станций. Также нет ограничений на минимальную длительность стоянки поезда в случае, если он участвует в перцепке вагонов.

Однако данная формулировка интересна своей разумной размерностью и понятной структурой. Если потоки заказов фиксированы, т.е. вагоны распределены по поездам, то мы знаем есть ли перцепка вагонов между каждой парой поездов на каждой станции, т.е. нам известны все отношения предшествования между движениями поездов. Таким образом, при фиксированных потоках заказов, задача может быть сформулирована как минимизация взвешенного суммарного запаздывания с отношениями предшествования без ресурсных ограничений. В данной задаче запаздывание каждого вагона может быть найдено нахождением “наидлиннейшего” пути в графе без циклов за линейное время относительно числа рёбер в графе.

## 2.4 Задача формирования грузовых потоков

Данная задача сочетает в себе долгосрочное планирование грузовых потоков через сортировочные станции и формирование грузовых составов с учетом среднесуточных потоков между сортировочными станциями. Вагонопотоки определяются на основе статистических данных и прогнозирования. Горизонт планирования составляет один год. Задача заключается в минимизации транспортных расходов при ограничении на сроки доставки, вместимость составов и эффективность сортировочных станций. Железнодорожная сеть состоит из сортировочных станций, между которыми происходит транспортировка грузовых вагонов. Известны среднесуточные потоки между каждой парой станций. Каждый вагон всегда доставляется по кратчайшему пути между станцией отправления и станцией назначения. Для каждой сортировочной станции необходимо составить следующий план:

- определить, как много поездов должно отправляться ежедневно с данной станции на станции назначения;
- распределить между поездами вагоны, для которых данная станция является либо станцией отправления, либо станцией переформирования.

Количество вагонов, назначенных каждому поезду не должно превышать некоторого числа. Каждый вагон должен быть доставлен в течение определенного времени, которое зависит от длины кратчайшего пути между станциями отправления и назначения. Общее время нахождения вагона в пути зависит от его маршрута и времени, затраченного на переформирование. Последнее необходимо для отсоединения вагона от состава и присоединения к следующему и зависит от параметров сортировочной станции. Количество составов, обслуживаемых ежедневно на сортировочной станции, также зависит от её производительности.



### 2.4.1 Постановка задачи

Рассмотрим связный неориентированный граф  $G = (N, E)$  сети железных дорог. Пусть  $h_{ij}$  — длина ребра  $(i, j) \in E$ . В каждый период времени между вершинами  $i, j \in N$  необходимо перевезти  $n_{ij}$  вагонов, т.е.  $\{n_{ij}\}_{i,j \in N}$  — целочисленная, неотрицательная и несимметричная матрица.

Для транспортировки вагонов доступно некоторое количество идентичных составов, имеющих одинаковую вместимость  $n_{\max}$ .

Поезд доставляет вагоны из вершины  $v \in N$  в вершину  $w \in N$  по кратчайшему из допустимых путей в графе  $G$ , временные затраты при этом составляют  $p_{vw}$  (включая прицепку, отцепку вагонов). Стоимость использования состава равна  $c_{vw}$ .

Доставка товара (заказа), назначенного из  $i$  в  $j$ , должна быть выполнена к директивному сроку  $d_{ij}$  и может быть осуществлена разными поездами на разных участках кратчайшего пути из  $i$  в  $j$ . Количество поездов, одновременно выходящих из вершины  $v \in N$ , не должно превышать  $q_v$ . Задача заключается в отыскании расписания, минимизирующего транспортные расходы, удовлетворяющего соответствующим ограничениям.

Для сокращения записи все товары  $(i, j)$ ,  $i \in N$ , имеющие одинаковое назначение  $j \in N$  будем обозначать  $j$ . Пусть  $d_j = \max_{i \in V} d_{ij}$ .

Рассмотрим полный ориентированный граф  $K = (N, A)$ , в котором составу, доставляющему товар из  $v$  в  $w$ , соответствует дуга  $(v, w) \in A$ .

Пусть граф  $G_j = (N, A_j)$  — подграф графа  $G$ , ребра которого входят в кратчайшие пути к вершине  $j$ .  $A_j$  образует направленное *остовное дерево* с единственным путём из каждой вершины  $i \in N \setminus \{j\}$  в  $j$ , являющимся кратчайшим путем относительно длин ребер  $h_{ij}$ .

Рассмотрим транзитивное замыкание  $G_j^+ = (N, A_j^+)$  графа  $G_j$ . Заметим, что оно является подграфом  $K$ . Таким образом, каждый путь

доставки товара  $(i, j)$  соответствует маршруту между  $i$  и  $j$  в графе  $G_j^+$ .

Пусть  $f_{vwd}^j$  — количество вагонов  $j \in N$ , доставленных к сроку  $d \in \{1, 2, \dots, d_j\}$  в вершину  $w$  составом, идущим без расформирования из  $v$  в  $w$  (по ребру  $(v, w) \in A_j$ );  $g_{vw}$  — количество составов, движущихся по  $(v, w) \in A$  в каждый период времени.

Формулировка задачи примет следующий вид:

$$(2.54) \quad \min \sum_{(v,w) \in A} c_{vw} g_{vw},$$

$$(2.55a) \quad \sum_{(i,v) \in A_j^+} f_{i,v,p_{iv}}^j = n_{ij} \quad \forall i, j \in N,$$

$$\sum_{(w,v) \in A_j^+} f_{wvd}^j - \sum_{(v,w) \in A_j^+} f_{v,w,d+p_{vw}}^j = 0 \quad \forall v, j \in N, v \neq j,$$

$$(2.55b) \quad \forall d \in \{1, 2, \dots, d_j\},$$

$$(2.55c) \quad \sum_{(v,j) \in A_j^+} \sum_{d=1}^{d_j} f_{vjd}^j = \sum_{i \in N \setminus \{j\}} n_{ij} \quad \forall j \in N,$$

$$(2.55d) \quad \sum_{\substack{j \in N: \\ (v,w) \in A_j^+}} \sum_{d=1}^{d_j} f_{vwd}^j \leq n_{\max} \cdot g_{vw} \quad \forall (v, w) \in A,$$

$$(2.55e) \quad \sum_{w \in N \setminus \{v\}} g_{vw} \leq q_v \quad \forall v \in N,$$

$$f_{vwd}^j \in \mathbb{Z}_+ \quad \forall j \in N, \forall (v, w) \in A_j^+,$$

$$(2.55f) \quad \forall d \in \{1, 2, \dots, d_j\},$$

$$(2.55g) \quad g_{vw} \in \mathbb{Z}_+ \quad \forall (v, w) \in A,$$

Целевая функция минимизирует затраты на использование составов. Ограничения (2.55a)-(2.55c) гарантируют доставку всех вагонов из пунктов отправления в пункты назначения. Условия (2.55d) означают, что не превышена вместимость составов. Наконец, неравенства (2.55e) обеспе-

чивают выполнение ограничений на количество составов.

## 2.5 Задача оперативного управления движением составов

Рассматривается задача управления движением составов на некотором узле железных дорог.

Сеть железных дорог на рассматриваемом узле разбивается на участки (границами участков являются путевые семафоры). На каждом из таких участков одновременно может находиться не более одного железнодорожного состава. Участки пронумерованы номерами  $1, 2, \dots, M$ ,  $\mathcal{M} = \{m_1, \dots, m_M\}$  — множество участков. В случае, если между двумя соседними семафорами железная дорога имеет две колеи ("попутная" и "встречная"), то считаем, что заданы два участка дорог, т.е., если рассматривать участки как ресурсы, обслуживающие требования на проезд ж/д составов, то можно считать, что их мощность равна 1.

Среди множества участков выделяются следующие подмножества:

- участки, являющиеся смежными с соседними узлами железных дорог;
- внутренние участки.

На множестве участков задано несимметричное отношение смежности. Два участка  $m_i$  и  $m_j$  считаем смежными в случае, если есть путевой семафор, расположенный на границах данных участков. Несимметричность данного отношения обусловлена тем, что в задаче необходимо выделить понятие *движение по встречным путям*. Таким образом, переход состава  $m_i \rightarrow m_j$  соответствует движению в попутном направлении, а движение  $m_j \rightarrow m_i$  — во встречном направлении. Существуют участки, для которых отношение смежности является симметричным. Обычно, такие участки располагаются в маневровых зонах, в которых

нет необходимости выделять попутное и встречное направления движения.

Понятие несимметричности смежности участков будет использовано при решении задачи, когда движение по встречным путям будет допускаться в "крайнем" случае.

Множество составов  $\mathcal{N}$ , для которых необходимо составить оперативное расписание движения на рассматриваемом узле, определяется следующим образом:

- рассматриваются составы, которые в момент начала вычислений находятся в движении внутри узла. Для таких составов имеем момент поступления  $r_j = 0$ ;
- рассматриваются составы, для которых известно время поступления в один из пограничных для узла участков (в которых данный узел "стыкуется" с соседними узлами железных дорог). Время поступления  $r_j$  для таких составов вычисляется относительно планового времени прибытия состава в пограничный участок;
- рассматриваются составы, которые находятся внутри узла, но фактически на момент начала вычислений не выполняют движение. Например, это грузовые составы, которые находятся на формировании в некоторой сортировочной зоне. Для таких составов также известно плановое время начала движения, которое принимается за  $r_j$ .

Вычисляемое в рамках задачи расписание движения составов из множества  $\mathcal{N}$  учитывает изначально заданные *интервалы недоступности* отдельных участков, которые обуславливаются:

- существующим расписанием движения неучитываемых в рамках решения задачи составов (например, пассажирских поездов и электричек, которые движутся относительно своего утвержденного графика);

- случаями выхода из строя отдельных участков в результате некоторых форс-мажорных обстоятельств (технических поломок, аварий, стихийные бедствия и проч.).

Интервалы доступности участков задаются в виде набора множеств  $E_i = \{[t'_{i1}, t''_{i1}], [t'_{i2}, t''_{i2}], \dots\}$ ,  $i \in \mathcal{M}$ .

Время прохождения состава некоторого участка вычисляется на основе максимально допустимой скорости движения состава и его длины и может корректироваться в процессе решения в большую сторону (что соответствует ситуации, когда состав должен ждать освобождения следующего по маршруту движения участка).

Каждому из составов  $j \in \mathcal{N}$  назначается приоритет (вес)  $w_j$ , который учитывается в процессе построения расписания движения составов. На первом этапе будем рассматривать задачи, когда  $w_j = 1$ ,  $\forall j \in \mathcal{N}$ .

Задача построения оперативного движения составов формулируется следующим образом. Необходимо максимизировать пропускную способность рассматриваемого узла, т.е. обслужить как можно больше составов из множества  $\mathcal{N}$  с учетом заданных весов составов  $w_j$ .

Таким образом, задача по формулировке близка к задаче РАНЕЦ, для которой известны эффективные алгоритмы решения графического (динамического) типа.

В данной постановке задачи необходимо учесть следующие особенности.

- В начале вычислений необходимо зафиксировать правую границу интервала планирования  $T$ . Без этого невозможно рассматривать узел как "Ранец". В целом, данную величину можно определить естественным образом исходя из типа задачи:
  - если решается задача приведения в стационарное состояние, то интервал планирование определяется на основе времени устранения технической неисправности;

— если решается задача построения расписания в штатном режиме, то интервал планирования определяется на основе некоторой заранее заданной константы.

- После фиксации интервала планирования необходимо учитывать, что некоторые составы не могут пройти узел за отведенное время. Поэтому при вычислении целевой функции полученного Ранца необходимо учитывать возможность "частичного помещения предмета в Ранец".
- В случае, когда все составы могут преодолеть узел без особых сложностей и взаимных конфликтов необходимо предварительно ввести в задачу "пустые" составы, наличие которых будет способствовать оптимизации прохождению реальных составов.
- Необходимо исключить искусственные простои поездов (что соответствует, в том числе, искусственному занижению скорости движения составов на свободных участках).

Под  $x_j$  будем понимать значение из интервала  $[0, 1]$ , которое соответствует коэффициенту прохождения узла составом  $j \in \mathcal{N}$ :

- $x_j = 0$  — состав не въехал в узел;
- $x_j = 1$  — состав проехал свой маршрут по участкам узла полностью;
- $0 < x_j < 1$  — состав проехал требуемый маршрут не полностью, значение равно отношению пройденного пути к длине всего маршрута.

Пусть  $\alpha_j(t) \in \mathcal{M}$  обозначает участок, в котором находится состав  $j \in \mathcal{N}$  в момент времени  $t$ . Разобьем весь интервал планирования на равные подинтервалы, границами которых являются моменты времени  $t_1, t_2, \dots, t_T$ . Количество таких точек разбиения соответствует допустимой дискретности рассматриваемого процесса движения составов.

Набор значений  $\alpha_j = \{\alpha_j(t_1), \alpha_j(t_2), \dots, \alpha_j(t_T)\}$  задаёт расписание (график) движения состава по маршрутам узла. Допустимым считается такой график, в котором:

- для любого  $i \in \{1, 2, \dots, T-1\}$  участки  $\alpha_j(t_i)$  и  $\alpha_j(t_{i+1})$  являются смежными (по определению, участок смежен самому себе);
- для любого  $i \in \{1, 2, \dots, T-1\}$  расстояние между центрами участков  $\alpha_j(t_i)$  и  $\alpha_j(t_{i+1})$  не должно быть больше дистанции, которую может пройти состав  $j$  за время  $t_{i+1} - t_i$ , двигаясь со своей нормальной скоростью;
- для любого  $i \in \{1, 2, \dots, T\}$  участок  $\alpha_j(t_i)$  доступен для движения в момент времени  $t_i$ .

Задача оперативного управления движением составов заключается в максимизации взвешенной пропускной способности узла железных дорог:

$$(2.56) \quad \max_{0 \leq x_j \leq 1} w_1 x_1 + w_2 x_2 + \dots + w_N x_N$$

при условиях:

- существования допустимых расписаний  $\alpha_j$  движения тех составов  $j$ , для которых  $x_j > 0$ , т.е. все составы, которые совершили движение по путям ж/д узла, выполнили это по допустимым расписаниям движения;
- отсутствия недопустимых пересечений графиков движения поездов, т.е. не существует расписаний движения двух составов  $j_1$  и  $j_2$ , у которых для некоторого  $t$  выполняется  $\alpha_{j_1}(t) = \alpha_{j_2}(t)$ .

## Глава 3

# Алгоритмы решения задач железнодорожного планирования

В данной главе главным образом рассматриваются точные алгоритмы решения частных случаев задач, поставленных в гл. 2. Показана взаимосвязь задач о двух станциях с задачами теории расписаний для одного прибора. Также предлагается специальный подход к приближённому решению задач железнодорожного планирования.

### 3.1 Решение задачи минимизации среднего времени выполнения заказов для двух станций с одним локомотивом

Имеются две станции, соединенные двухпутной железной дорогой. Необходимо выполнить множества заказов  $N^1 = \{J_1^1, \dots, J_n^1\}$  и  $N^2 = \{J_1^2, \dots, J_m^2\}$  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. Все вагоны одинаковые. Пусть  $r_i^s$  — время поступления заказа  $J_i^s \in N^s$ ,  $s \in \{1, 2\}$ , на станцию  $s$ . Без потери общности предположим, что заказы пронумерованы в порядке их поступления.

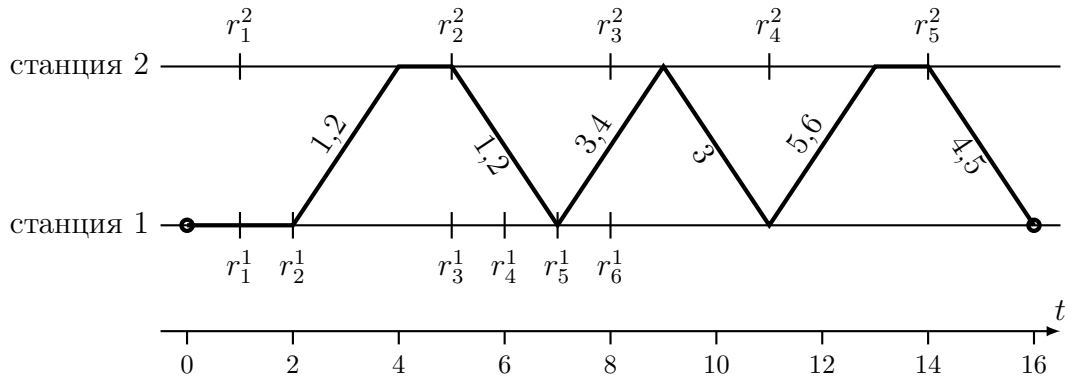
Доставка вагонов с одной станции на другую осуществляется с по-



мощью одного локомотива. Одновременно локомотив может перевезти не более  $q$  вагонов. Пусть  $p$  — время движения состава между станциями. Без потери общности предполагается, что в нулевой момент времени поезд находится на станции 1.

Задача заключается в составлении плана формирования составов с целью минимизации суммарного времени доставки грузов. Пусть  $C_i^s$  — момент доставки заказа  $J_i^s \in N^s$ ,  $s \in \{1, 2\}$ . Тогда целевая функция задачи записывается как

$$\min \sum_{J_i^1 \in N^1} C_i^1 + \sum_{J_i^2 \in N^2} C_i^2.$$



**Рис. 3.1.** Допустимое расписание для Примера 3.3.0.1

**Пример 3.1.0.1.** Рассмотрим следующий пример задачи. Пусть  $n = 6$ ,  $m = 5$ ,  $r^1 = (1, 2, 5, 6, 7, 8)$ ,  $r^2 = (1, 5, 8, 11, 14)$ ,  $p = 2$ ,  $q = 2$ . Допустимое расписание для данного примера проиллюстрировано на рис. 3.1. Здесь жирная линия обозначает движение локомотива между станциями. Цифры вдоль этой линии обозначают номера заказов, соответствующих перевозимым вагонам.

Рассматриваемая задача может быть решена с помощью алгоритма динамического программирования. Алгоритм основан на следующих достаточно очевидных, но важных свойствах задачи.

**Свойство 1.** Существует оптимальное расписание, при котором

- (i) локомотив каждый раз отправляется со станции  $s \in \{1, 2\}$  в момент поступления некоторого заказа  $J_i^s \in N^s$  или в момент прибытия на станцию;
- (ii) при каждом отправлении локомотива со станции  $s \in \{1, 2\}$ , в состав включается максимальное допустимое число вагонов (минимум между вместимостью состава  $q$  и числом поступивших к этому времени на станцию  $s$  и еще не доставленных вагонов);
- (iii) заказы из множества  $N^s$ ,  $s \in \{1, 2\}$ , перевозятся в порядке поступления, то есть  $C_1^1 \leq C_2^1 \leq \dots \leq C_n^1$  и  $C_1^2 \leq C_2^2 \leq \dots \leq C_m^2$ ;
- (iv) локомотив делает не более одного “холостого”, т.е. без вагонов, переезда подряд.

Свойство 1 легко доказывается приведением оптимального расписания ему не удовлетворяющего к другому оптимальному расписанию удовлетворяющего данному свойству. Заметим, что для расписания изображенного на рис. 3.1 выполняются все условия (i)–(iv).

Предлагается алгоритм, основанный на методе динамического программирования, который строит оптимальное расписание, удовлетворяющее Свойству 1. Для алгоритма нам понадобятся три фиктивных заказа  $J_0^1$ ,  $J_{n+1}^1$ ,  $J_{m+1}^2$ . Времена поступления заказа  $J_0^1$  равняются нулю. Времена поступления заказов  $J_{n+1}^1$  и  $J_{m+1}^2$  равняются некоторому числу, большому, чем момент времени, к которому все заказы могут быть доставлены.

Обозначим через  $S(s, k, i, j)$  состояние в котором доставлены первые  $i$ ,  $0 \leq i \leq n$ , заказов из  $N^1$ , первые  $j$ ,  $0 \leq j \leq m$ , заказов из  $N^2$ , и локомотив находится на станции  $s \in \{1, 2\}$  в момент времени  $r_k^s$ , таком что,

$$\text{если } s = 1, \text{ то } k - q \leq i < k \text{ и } r_k^1 \geq r_j^2 + p,$$

$$\text{если } s = 2, \text{ то } k - q \leq j < k \text{ и } r_k^2 \geq r_i^1 + p.$$

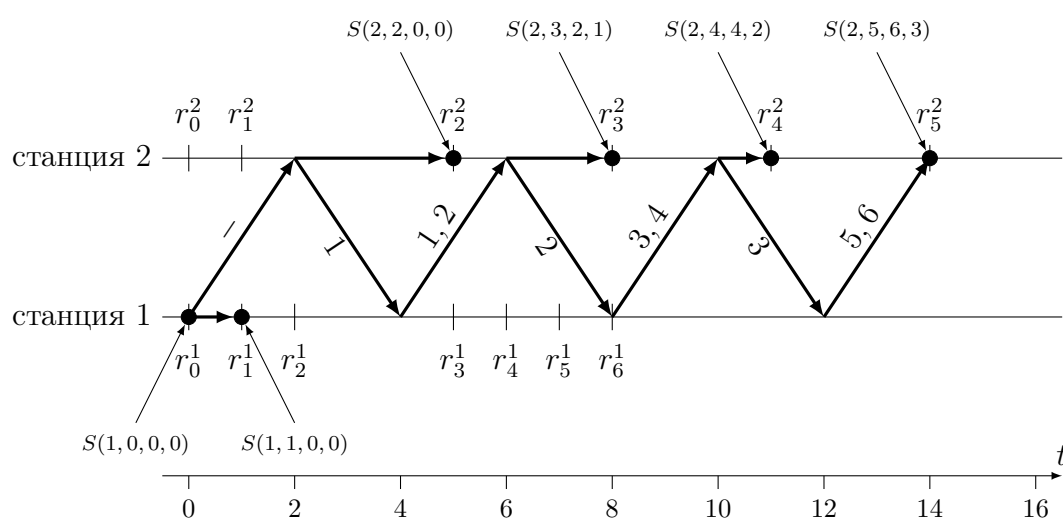
Пусть  $P(s, k, i, j)$  — наименьшее суммарное время доставки грузов в

частичном расписании, приводящем к состоянию  $S(s, k, i, j)$ . Заметим, что оптимальным решением рассматриваемой задачи является

$$(3.1) \quad \min\{P(1, n + 1, n, m), P(2, m + 1, n, m)\}.$$

Мы будем называть состояния  $S(1, n + 1, n, m)$  и  $S(2, m + 1, n, m)$  *конечными*.

**Пример 3.1.0.2. (продолжение)** На рис. 3.2 показаны возможные переходы из состояния  $S(1, 0, 0, 0)$  в другие.



**Рис. 3.2. Возможные переходы из состояния  $S(1, 0, 0, 0)$  в Примере 3.3.0.1**

В данном состоянии у локомотива есть выбор: или остаться на станции 1 до момента времени  $r_1^1$ , то есть перейти в состояние  $S(1, 1, 0, 0)$ , или отправиться на станцию 2 без вагонов. В последнем случае у локомотива так же есть выбор: или остаться на станции 2 до момента времени  $r_2^2$  или сразу же отправиться до станции 1 с одним доступным вагоном.

В дальнейшем “цепочка” продолжается; при каждом прибытии на станцию, согласно условию (i), у локомотива есть выбор:

- 1) или остаться на этой станции до следующего поступления вагона, то есть перейти в некоторое другое состояние,

2) или же сразу отправиться обратно с максимальным допустимым числом вагонов согласно условию (ii).

Отметим, что выбор есть не всегда. Если число доступных вагонов не меньше, чем вместимость состава или все вагоны уже поступили, оптимальным будет всегда вариант 2. При этом, согласно условию (iii), в состав включаются доступные вагоны с наименьшим временем поступления. В нашем примере данная ситуация возникает, когда локомотив прибывает на станцию 1 в моменты времени 4, 8, 12.

Если локомотив прибыл на станцию в момент прибытия вагона, причем число доступных вагонов не превышает вместимость поезда, то он оказывается в некотором состоянии, и “цепочка” прерывается.

В нашем примере данная ситуация возникает, когда локомотив прибывает на станцию 2 в момент времени 14.

Отметим также, что согласно условию (iv), “цепочка” прерывается, когда локомотив два раза подряд сделал “холостой” переезд.

Таким образом, переход из одного состояния в другое соответствует “челночному” или “зигзагообразному” движению локомотива между станциями без остановки, и стоянке на одной из станций в ожидании поступления вагона по окончании этого движения. Заметим, что фаза движения присутствует не обязательно, а фаза стоянки может быть нулевой длительности.

Алгоритм 1 описывает процедуру осуществления всех возможных переходов из данного состояния  $S(s, k, i, j)$  в другие.

Опишем переменные используемые в данной процедуре:

$t$  — текущий момент времени;

$i$  — текущее количество доставленных заказов из множества  $N^1$ ;

$j$  — текущее количество доставленных заказов из множества  $N^2$ ;

**s** — станция, на которой находится локомотив в текущий момент времени;

**k1** — минимальный номер заказа из множества  $N^1$ , доступный в момент времени **t** или позже;

**k2** — минимальный номер заказа из множества  $N^2$ , доступный в момент времени **t** или позже;

**P** — суммарное время доставки уже доставленных заказов;

**idle** — число “холостых” переездов, сделанных локомотивом подряд к текущему моменту времени;

**terminate** — булева переменная, обозначающая прервать ли локомотиву цепочку “челночных” переездов;

**wagons** — число вагонов, включенных в текущий состав.

Длина цепочки “челночных” переездов не превышает  $2(n+m)$ , так как хотя бы один вагон включается в состав каждые два переезда. Поэтому, сложность процедуры представленной в Алгоритме 1 составляет  $O(n+m)$ .

В начале алгоритма динамического программирования все возможные состояния  $S(s, k, i, j)$  упорядочиваются по неубыванию  $r_k^s$ . В этом порядке для каждого состояния, кроме конечных, выполняется Алгоритм 1. Значение оптимального решения определяется по формуле (3.1).

Общее количество состояний составляет порядка  $O(nmq)$ . Учитывая сложность Алгоритма 1, общая сложность алгоритма динамического программирования не превышает  $O(qnm(n+m))$  операций.

Так как вместимость состава  $q$  не превышает  $n+m$  (в противном случае без потери общности  $q$  можно принять равным  $n+m$ ), рассматриваемая задача может быть решена за полиномиальное время.

---

**Algorithm 1** Процедура осуществления всевозможных переходов из состояния  $S(s, k, i, j)$  в другие состояния.

---

```

1:  $\mathbf{P} \leftarrow P(s, k, i, j)$ ;
2:  $\mathbf{i} \leftarrow i$ ;  $\mathbf{j} \leftarrow j$ ;
3:  $\mathbf{s} \leftarrow s$ ;  $\mathbf{t} \leftarrow 0$ ;
4:  $\mathbf{t} \leftarrow r_k^s$ ;
5: if  $\mathbf{s} = 1$  then
6:    $\mathbf{k1} \leftarrow k + 1$ ;  $\mathbf{k2} \leftarrow j$ ;
7: end if
8: if  $\mathbf{s} = 2$  then
9:    $\mathbf{k2} \leftarrow k + 1$ ;  $\mathbf{k1} \leftarrow i$ ;
10: end if
11: repeat
12:   while  $r_{k1}^1 < \mathbf{t}$  do
13:      $\mathbf{k1} \leftarrow \mathbf{k1} + 1$ ;
14:   end while
15:   while  $r_{k2}^2 < \mathbf{t}$  do
16:      $\mathbf{k2} \leftarrow \mathbf{k2} + 1$ ;
17:   end while
18:   if  $\mathbf{s} = 1$  then
19:     if  $\mathbf{k1} - \mathbf{i} < q$  and  $\mathbf{P} < P(1, \mathbf{k1}, \mathbf{i}, \mathbf{j})$  then
20:        $P(1, \mathbf{k1}, \mathbf{i}, \mathbf{j}) \leftarrow \mathbf{P}$ ;
21:     end if
22:      $\text{terminate} \leftarrow (r_{k1}^1 = \mathbf{t}) \ \& \ (\mathbf{k1} - \mathbf{i} \leq q)$ ;
23:   end if
24:   if  $\mathbf{s} = 2$  then
25:     if  $\mathbf{k2} - \mathbf{j} < q$  and  $\mathbf{P} < P(2, \mathbf{k2}, \mathbf{i}, \mathbf{j})$  then
26:        $P(2, \mathbf{k2}, \mathbf{i}, \mathbf{j}) \leftarrow \mathbf{P}$ ;
27:     end if
28:      $\text{terminate} \leftarrow (r_{k2}^2 = \mathbf{t}) \ \& \ (\mathbf{k2} - \mathbf{j} \leq q)$ ;
29:   end if
30:   if not  $\text{terminate}$  then
31:     if  $\mathbf{s} = 1$  then
32:        $\text{wagons} \leftarrow \min\{q, \mathbf{k1} - 1 - \mathbf{i}\}$ ;
33:        $\mathbf{i} \leftarrow \mathbf{i} + \text{wagons}$ ;
34:        $s \leftarrow 2$ ;
35:     end if
36:     if  $\mathbf{s} = 2$  then
37:        $\text{wagons} \leftarrow \min\{q, \mathbf{k2} - 1 - \mathbf{j}\}$ ;  $\mathbf{j} \leftarrow \mathbf{j} + \text{wagons}$ ;
38:        $s \leftarrow 1$ ;
39:     end if
40:      $\mathbf{t} \leftarrow \mathbf{t} + p$ ;
41:      $\mathbf{P} \leftarrow \mathbf{P} + \mathbf{t} \cdot \text{wagons}$ ;
42:     if  $\text{wagons} = 0$  then
43:        $\text{idle} \leftarrow \text{idle} + 1$ 
44:     end if
45:   else
46:      $\text{idle} \leftarrow 0$ 
47:   end if
48: until  $\text{terminate}$  or  $\text{idle} = 2$ 

```

---

## Упражнения

С помощью рассмотренного алгоритма построить расписания для задач со следующими параметрами:

**Упр. 3.1.0.1.**  $n = 6$ ,  $m = 6$ ,  $r^1 = (1, 2, 3, 8, 9, 10)$ ,  $r^2 = (5, 6, 7, 11, 12, 13)$ ,  $p = 2$ ,  $q = 3$ .

**Упр. 3.1.0.2.**  $n = 6$ ,  $m = 6$ ,  $r^1 = (1, 2, 3, 8, 9, 10)$ ,  $r^2 = (5, 6, 7, 11, 12, 15)$ ,  $p = 2$ ,  $q = 3$ .

**Упр. 3.1.0.3.**  $n = 5$ ,  $m = 5$ ,  $r^1 = (1, 2, 3, 4, 5)$ ,  $r^2 = (2, 3, 4, 5, 6)$ ,  $p = 2$ ,  $q = 2$ .

### 3.2 Задача составления расписания движения поездов между двумя станциями, соединенными однопутной железной дорогой

Задача поиска оптимального расписания движения поездов между двумя станциями, соединенными однопутной железной дорогой, (Single Track Railway Scheduling Problem with two stations, STRSP2) формулируется следующим образом.

Пусть однопутная железная дорога, соединяющая 2 станции, разделена на  $Q$  сегментов  $1, 2, \dots, Q$ , и заданы множества  $N'_1$  и  $N'_2$  поездов,  $N'_1 \cap N'_2 = \emptyset$ . Поезда из множества  $N'_1$  следуют со станции 1 на станцию 2, а поезда из  $N'_2$  движутся в противоположном направлении. Множество  $N' = N'_1 \cup N'_2$  содержит  $n'$  поездов,  $|N'_1| = n'_1$ ,  $|N'_2| = n'_2$ ,  $n'_1 + n'_2 = n'$ . Поезда из множества  $N'_1$  проходят сегменты в порядке  $1 \rightarrow 2 \rightarrow \dots \rightarrow Q$ , а поезда из  $N'_2$  в порядке  $Q \rightarrow Q - 1 \rightarrow \dots \rightarrow 1$ . На одном сегменте дороги не может находиться одновременно более одного поезда. Если поезд  $j' \in N'_1$  находится на некотором участке пути, то ни один из поездов  $i' \in N'_2$  не может начать движение и наоборот. Для каждого участка  $q$ ,  $q = 1, 2, \dots, Q$ , задано время прохождения  $p_q$ , за которое любой поезд

$j \in N$  его проходит, т.е. все поезда движутся с одинаковой скоростью. Пусть  $S_{j'}(\Pi)$  и  $C_{j'}(\Pi)$ ,  $j' \in N'$  — моменты начала и окончания движения поезда  $j'$  в расписании  $\Pi$ , т.е.  $S_{j'}(\Pi)$  — время выхода поезда  $j'$  со станции отправления, а  $C_{j'}(\Pi)$  — время прибытия поезда на станцию назначения. Тогда в допустимом расписании выполняются следующие условия:

- $C_{j'} \geq S_{j'} + \sum_{q=1}^Q p_q$ ,  $j' \in N'$ ;
- $\forall i' \in N'_1, \forall j' \in N'_2 C_{i'} \leq S_{j'}$  или  $C_{j'} \leq S_{i'}$ .

Для каждого поезда  $j' \in N'$  заданы директивный срок  $d_{j'} \geq 0$ , приоритет (вес)  $w_{j'} \geq 0$  и время, с которого поезд доступен для отправления  $r_{j'} \geq 0$  (самый ранний из возможных моментов отправления  $S_{j'} \geq r_{j'}$ ). Если  $C_{j'}(\Pi) > d_{j'}$ , то говорят, что поезд  $j'$  запаздывает, при этом принимают  $U_{j'}(\Pi) = 1$ , иначе  $U_{j'}(\Pi) = 0$ . Если  $C_{j'}(\Pi) \leq d_{j'}$ , то поезд  $j'$  не запаздывает. Обозначим  $T_{j'}(\Pi) = \max\{0, C_{j'}(\Pi) - d_{j'}\}$  — запаздывание поезда  $j'$  при расписании  $\Pi$  и  $C_{max}(\Pi) = \max_{j' \in N'} \{C_{j'}(\Pi)\}$  — время окончания всех перевозок при этом расписании. Задачу STRSP2 поиска оптимального расписания  $\pi^*$ , минимизирующего время окончания перевозок  $C_{max}$ , обозначим  $STRSP2|r_j|C_{max}$  (в соответствии с трехпозиционной системой обозначений  $\alpha|\beta|\gamma$  для задач теории расписаний [35], где  $\alpha$  обозначает множество ресурсов,  $\beta$  описывает ограничения,  $\gamma$  содержит целевую функцию). Дополнительно будем рассматривать задачу STRSP2 с другими целевыми функциями и ограничениями:

- минимизация числа запаздывающих поездов при их одновременном поступлении  $STRSP2|| \sum U_j$ ;
- минимизация взвешенного числа запаздывающих поездов при их одновременном поступлении  $STRSP2|| \sum w_j U_j$ ;
- минимизация суммарного времени перевозок  $STRSP2|r_j| \sum C_j$  при заданных моментах поступления поездов на станции отправления;



- минимизация взвешенного суммарного времени при одно-  
временном поступлении поездов на станции отправления  
 $STRSP2 || \sum w_j C_j$ .

Нам не известны публикации по сформулированным выше задачам. Однако стоит отметить, что при  $Q = 1$  данные задачи эквиваленты классическим одноприборным задачам теории расписаний [22] и, следовательно, в случае различных скоростей для поездов на сегменте, некоторые из задач оказываются NP-трудными [22]. Заметим также, что данные задачи могут быть легко сформулированы в терминах многоприборных задач теории расписаний с  $Q$  приборами [22].

Далее представлена полиномиальное сведение задач  $STRSP2$  к специальному случаю одноприборной задачи, одинаковым временем обслуживания всех требований и временем переналадки, а также предлагаются полиномиальные алгоритмы минимизации указанных выше целевых функций для полученной одноприборной задачи.

### 3.2.1 Сведение $STRSP2$ к одноприборной задаче

Введем обозначения:  $p_{max} = \max_{q=1,2,\dots,Q} \{p_q\}$  и  $P = \sum_{q=1}^Q p_q$ .

**Лемма 3.1.** Пусть для поезда  $j' \in N'_1$  время прибытия на станцию назначения определяется как  $C_{j'} = S_{j'} + P$ , а поезд  $i' \in N'_1$  — следующий за ним поезд. Тогда существует допустимое расписание, в котором  $S_{i'} = \max\{r_{i'}, S_{i'} + p_{max}\}$ ,  $C_{i'} = S_{i'} + P$ , т.е. поезд  $i'$  начинает движение со станции отправления в момент  $\max\{r_{i'}, S_{j'} + p_{max}\}$  и движется без остановок.

**Доказательство.** Пусть  $S_{j'}^q, S_{i'}^q, q = 1, 2, \dots, Q$  — моменты начала прохождения отрезка  $q$  поездами  $j'$  и  $i'$  соответственно. Чтобы доказать допустимость рассматриваемого расписания, достаточно показать, что  $S_{i'}^q \geq S_{j'}^q + p_q, q = 1, 2, \dots, Q$ , т.е. поезд  $i'$  начинает движение по сегменту  $q$ , когда поезд  $j'$  уже прошел его. Имеем  $S_{j'}^q = S_{j'} + \sum_{l=1}^{q-1} p_l$  и

$S_{i'}^q = S_{i'} + \sum_{l=1}^{q-1} p_l = \max\{r_{i'}, S_{j'} + p_{max}\} + \sum_{l=1}^{q-1} p_l \geq S_{j'}^q + p_q$ , т.е. лемма верна.  $\square$

Лемма 3.1 определяет периодичность отправления поездов с одной станции, если в расписании они следуют друг за другом. Необходимо отметить, что  $\max\{r_{i'}, S_{j'} + p_{max}\}$  — самый ранний из возможных моментов отправления для поезда  $i'$ , т.к. для участка пути  $q$  величина  $p_q = p_{max}$ . Получаем  $S_{i'}^q = S_{j'}^q + p_q$  и, следовательно,  $|C_{j'} - C_{i'}| \geq p_{max}$  для любых поездов  $j', i'$ , принадлежащих одному из множеств  $N'_1$  или  $N'_2$ .

Из Леммы можно сделать также следующий вывод. Для указанных выше целевых функций существует оптимальное расписание, при котором поезда движутся без остановок, т.е. начав движение во время  $S_j$  поезд  $j$  прибывает на станцию назначения во время  $C_j = S_j + P$ . Далее мы будем рассматривать только расписания данного вида.

**Лемма 3.2.** Для любых двух поездов  $j'$  и  $i'$ , принадлежащих одному из множеств  $N'_1$  или  $N'_2$  выполняется  $|S_{j'} - S_{i'}| \geq p_{max}$  и  $|C_{j'} - C_{i'}| \geq p_{max}$ .

Пусть последовательность поездов  $(j'_1, j'_2, \dots, j'_n)$  задает очередность движения поездов между станциями. Очевидно, что допустимому расписанию соответствует одна и только одна последовательность поездов. Таким образом, оптимальное расписание соответствует *оптимальной последовательности* поездов. Для заданной последовательности  $(j'_1, j'_2, \dots, j'_n)$  расписание можно построить следующим образом:

$$(3.2) \quad \begin{cases} S_{j'_1} = r_{j'_1}, & C_{j'_1} = S_{j'_1} + P, \\ S_{j'_k} = \max\{r_{j'_k}, S_{j'_{k-1}} + p_{max}\}, & C_{j'_k} = S_{j'_k} + P, \quad k = 2, \dots, n', (*) \\ S_{j'_k} = \max\{r_{j'_k}, S_{j'_{k-1}} + P\}, & C_{j'_k} = S_{j'_k} + P, \quad k = 2, \dots, n', (**). \end{cases}$$

(\*) выполняется, если оба поезда  $j'_k$  и  $j'_{k-1}$  принадлежат одному множеству  $N'_1$  или  $N'_2$ , иначе выполняется (\*\*).

Согласно Лемме 3.1 построенное расписание будет допустимым. Более того, для представленных выше целевых функций, монотонных по вре-

мени окончания движения, в соответствии с Леммой 3.2 алгоритм (3.2) по заданной оптимальной последовательности поездов построит оптимальное расписание.

На основе данных свойств в следующем разделе предлагается сведение исходной задачи к одноприборной задаче теории расписаний.

Задано множество  $N = N_1 \cup N_2$ ,  $N_1 \cap N_2 = \emptyset$ , содержащее  $n$  требований, которые должны быть обслужены на одном приборе. Прерывание обслуживания требований не допускается. Одновременно прибором может обслуживаться только одно требование. Продолжительность обслуживания требования равна  $p$ ,  $\forall j \in N$ . Для каждого требования  $j \in N$  заданы директивный срок  $d_j \geq 0$ , вес  $w_j \geq 0$ , и время поступления требования на обслуживание  $r_j \geq 0$ . Допустимое решение задается перестановкой  $\pi = (j_1, j_2, \dots, j_n)$  требований из множества  $N$ , из которой соответствующее расписание может быть получено назначением каждому требованию наиболее раннего из возможных моментов старта. Пусть  $S_{j_k}(\pi)$ ,  $C_{j_k}(\pi) = S_{j_k}(\pi) + p$  — моменты начала и окончания обслуживания требования  $j_k$  при расписании, полученном из перестановки  $\pi$ .

Если  $j_k \in N_1$  и  $j_{k+1} \in N_2$ , то между обслуживанием требований необходимо выполнить переналадку продолжительностью  $st = st_1$ . Если  $j_k \in N_2$  и  $j_{k+1} \in N_1$ , то между обслуживанием требований необходимо выполнить переналадку продолжительностью  $st = st_2$ . Между обслуживанием требований из одного множества переналадка не требуется, т.е.  $st = 0$ . В допустимом расписании выполняется условие  $S_{j_{k+1}} = \max\{r_{j_{k+1}}, C_{j_k} + st\}$ . Будем рассматривать те же целевые функции, что и для задачи *STRSP2*. Если  $C_j(\pi) > d_j$ , то требование  $j$  запаздывает, и полагают  $U_j(\pi) = 1$ , иначе  $U_j(\pi) = 0$ . Если  $C_j(\pi) \leq d_j$ , то требование  $j$  не запаздывает. Пусть  $T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$  — запаздывание работы  $j$ , а  $C_{max}(\pi) = \max_{j \in N}\{C_j(\pi)\}$  — время окончания всех работ.

Обозначим задачу минимизации времени окончания обслуживания

всех требований для одного прибора с одинаковым временем обслуживания, заданными временами поступления работ и заданным временем переналадки через  $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$  в соответствии с общепринятой классификацией.

Задачи  $STRSP2|-|-$  для упомянутых выше целевых функций могут быть сведены к задачам  $1|setup - times, N_1, N_2, p_j = p, -|-$  следующим образом. Подмножество поездов  $N'_1$  соответствует подмножеству требований  $N_1$ ,  $|N_1| = |N'_1|$ , а подмножество поездов  $N'_2$  — подмножеству требований  $N_2$ ,  $|N_2| = |N'_2|$ . Пусть  $q$ ,  $q \in \{1, 2, \dots, Q\}$  — индекс участ-  
ка, для которого  $p_q = p_{max}$ . Обозначим  $TAIL_{left} = \sum_{l=1}^{q-1} p_l$ ,  $TAIL_{right} = \sum_{l=q+1}^Q p_l$ . Положим  $p = p_{max}$ ,  $st_1 = 2 \cdot TAIL_{right}$ ,  $st_2 = 2 \cdot TAIL_{left}$ , если  $j \in N_1$ , то время поступления работы  $r_j = r_{j'} + TAIL_{left}$ , иначе  $r_j = r_{j'} + TAIL_{right}$ . Если  $j \in N_1$ , то  $d_j = d_{j'} - TAIL_{right}$ , в противном случае  $d_j = d_{j'} - TAIL_{left}$ . Веса работ остаются без изменений.

Рассмотрим последовательность обслуживания требований, соответствующую последовательности движения поездов  $(j'_1, j'_2, \dots, j'_n)$  между станциями, где требование  $j_k$ ,  $k = 1, 2, \dots, n$ , соответствует поезду  $j'_k$ , а также расписания, полученные из данных последовательностей, в которых каждое/ый требование/поезд обслуживается/начинает движение как можно раньше (для требований) или в соответствии с алгоритмом (3.2) (для поездов). Табл. 1 содержит соответствия требований  $j$  и поездов  $j'$ . В табл. 2 указаны соответствия целевых функций задач.

Таким образом, можно заключить, что для указанных в таблице целевых функций оптимальная последовательность обслуживания требований  $(j_1, j_2, \dots, j_n)$  соответствует оптимальной последовательности движения поездов  $(j'_1, j'_2, \dots, j'_n)$ , т.е. задачи  $STRSP2|-|-$  можно свести к соответствующим задачам  $1|setup - times, N_1, N_2, p_j = p, -|-$  за полиномиальное время. В полученных задачах для одного прибора обслуживание всех требований  $j \in N_1$  начинается не раньше времени  $r = TAIL_{left}$ ,

Требование (поезд)	Время поступления	Директ. срок	Время старта	Время окончания
$j \in N_1$	$r_j = r_{j'} + TAIL_{left}$	$d_j = d_{j'} - TAIL_{right}$	$S_{j'} + TAIL_{left}$	$C_{j'} - TAIL_{right}$
$j' \in N'_1$	$r_{j'}$	$d_{j'}$	$S_{j'}$	$C_{j'}$
$j \in N_2$	$r_j = r_{j'} + TAIL_{right}$	$d_j = d_{j'} - TAIL_{left}$	$S_{j'} + TAIL_{right}$	$C_{j'} - TAIL_{left}$
$j' \in N'_2$	$r_{j'}$	$d_{j'}$	$S_{j'}$	$C_{j'}$

**Таблица 3.1. Соответствие параметров**

Целевая функция задачи $STRSP2 - -$	Целевая функция задачи $1 setup - times, N_1, N_2, p_j = p, - -$
$\sum w_{j'} U_{j'}$	$\sum w_{j'} U_{j'}$
$\sum T_{j'}$	$\sum T_{j'}$
$\sum w_{j'} C_{j'}$	$\sum w_{j'} C_{j'} + \sum_{j' \in N'_1} w_{j'} \cdot TAIL_{right} + \sum_{j' \in N'_2} w_{j'} \cdot TAIL_{left}$

**Таблица 3.2. Соответствие значений целевых функций**

а для требований  $j \in N_2$  – не раньше момента времени  $r = TAIL_{right}$ . Предлагаемые далее алгоритмы решения предполагают, что все моменты поступления  $r$  равны 0, однако данные алгоритмы могут быть легко преобразованы для решения исходных задач с временами поступления не равными 0.

Можно представить аналогичное сведение задачи  $STRSP2|r_j|C_{max}$  к задаче  $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$ , однако в этом случае не будет строгого соответствия оптимальных значений функции  $C_{max}$ , т.е. оптимальная последовательность выполнения требований  $(j_1, j_2, \dots, j_n)$  может соответствовать неоптимальной последовательности движения поездов  $(j'_1, j'_2, \dots, j'_n)$ . Тем не менее, алгоритм решения задачи  $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$  может быть модифицирован для решения задачи  $STRSP2|r_j|C_{max}$ .

Таким образом, далее представлены алгоритмы решения следующих задач.

- $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$ ;

- $1|setup - times, N_1, N_2, p_j = p, r_j| \sum C_j$  ;
- $1|setup - times, N_1, N_2, p_j = p| \sum w_j C_j$ ;
- $1|setup - times, N_1, N_2, p_j = p| \sum T_j$ ;
- $1|setup - times, N_1, N_2, p_j = p| \sum U_j$ ;
- $1|setup - times, N_1, N_2, p_j = p| \sum w_j U_j$ .

Обзор по одноприборным задачам с переналадками представлен, например, в [13]. Некоторые результаты для одноприборных задач с равными продолжительностями обслуживания требований описаны в [15, 44].

**Определение 1.** Назовем расписания для задач  $1|setup - times, N_1, N_2, p_j = p, -|$  “сдвинутыми влево”, если обслуживание каждого требования начинается в самый ранний из возможных моментов времени. Очевидно, что для любой из упомянутых выше задач существует оптимальные расписания, которые являются “сдвинутыми влево”.

**Определение 2.** Определим множество  $\Theta$  следующим образом:  
 $\Theta = \{t | t = r_j + x_1 \cdot p + x_2 \cdot st_1 + x_3 \cdot st_2, j \in \{1, 2, \dots, n\}, x_1, x_2, x_3 \in \{0, 1, 2, \dots, n\}, x_2 + x_3 \leq x_1\}$ .

Заметим, что множество  $\Theta$  содержит не более  $O(n^4)$  элементов.

**Лемма 3.3.** Во всех “сдвинутых влево” расписаниях моменты начала обслуживания требований принадлежат множеству  $\Theta$ .

**Доказательство.** Пусть в допустимом “сдвинутом влево” расписании  $\Pi$  требование  $j_k$ ,  $1 < k < n$ , является первым требованием, для которого  $S_{j_k} \notin \Theta$ , т.е для предшествующего ему требования  $j_{k-1}$  выполнено  $S_{j_{k-1}} \in \Theta$ . Самый ранний из возможных моментов  $S$  начала обслуживания требования  $j_k$  определяется следующим образом:

$$\begin{cases} S = \max\{r_{j_k}, S_{j_{k-1}} + p\}, & (*) \\ S = \max\{r_{j_k}, S_{j_{k-1}} + p + st_1\}, & (**) \\ S = \max\{r_{j_k}, S_{j_{k-1}} + p + st_2\}. & (***) \end{cases}$$

(\*) выполняется, если требования  $j_k$  и  $j_{k-1}$  принадлежат одному и тому же множеству  $N_1$  или  $N_2$ , (\*\*) выполняется, когда  $j_k \in N_2$  и  $j_{k-1} \in N_1$ , а (\*\*\*) справедливо для случая, когда  $j_k \in N_1$  и  $j_{k-1} \in N_2$ . Очевидно, что  $S \in \Theta$ . Поскольку  $S_{j_k} \notin \Theta$ , получаем  $S < S_{j_k}$ , а значит, расписание  $\Pi$  не является “сдвинутым влево”.  $\square$

### 3.2.2 Алгоритмы для задач с упорядоченными подмножествами $N_1$ и $N_2$

Далее представлены алгоритмы решения следующих задач:

- $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$ ;
- $1|setup - times, N_1, N_2, p_j = p, r_j|\sum C_j$ ;
- $1|setup - times, N_1, N_2, p_j = p|\sum w_j C_j$ ;
- $1|setup - times, N_1, N_2, p_j = p|\sum T_j$ .

Все алгоритмы основаны на одинаковых свойствах оптимального решения и используют одну и ту же процедуру поиска.

Введем обозначения:  $N_1 = \{j_1, j_2, \dots, j_{n_1}\}$  и  $N_2 = \{i_1, i_2, \dots, i_{n_2}\}$ .

**Лемма 3.4.** Для всех указанных выше задач существует оптимальное расписание, при котором требования выполняются в одной из следующих последовательностей.

- для задач  $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$  и  $1|setup - times, N_1, N_2, p_j = p, r_j|\sum C_j$  требования из каждого подмножества  $N_1$  и  $N_2$  обслуживаются в порядке по неубыванию моментов поступления, т.е.  $r_{j_1} \leq r_{j_2} \leq \dots \leq r_{j_{n_1}}$  и  $r_{i_1} \leq r_{i_2} \leq \dots \leq r_{i_{n_2}}$ ;
- для задач  $1|setup - times, N_1, N_2, p_j = p|\sum w_j C_j$  требования из каждого подмножества  $N_1$  и  $N_2$  обслуживаются в порядке по невозрастанию весов, т.е.  $w_{j_1} \geq w_{j_2} \geq \dots \geq w_{j_{n_1}}$  и  $w_{i_1} \geq w_{i_2} \geq \dots \geq w_{i_{n_2}}$ ;

- для задач  $1|setup-times, N_1, N_2, p_j = p|\sum T_j$  требования из каждого подмножества  $N_1$  и  $N_2$  обслуживаются в порядке по неубыванию директивных сроков, т.е.  $d_{j_1} \leq d_{j_2} \leq \dots \leq d_{j_{n_1}}$  и  $d_{i_1} \leq d_{i_2} \leq \dots \leq d_{i_{n_2}}$ .

**Доказательство.** Лемма может быть легко доказана следующим образом. Если в оптимальном расписании  $\Pi$  два требования, принадлежащие одному и тому же подмножеству  $N_1$  или  $N_2$ , обслуживаются с нарушением указанного порядка, то в расписании их можно поменять местами без увеличения значения целевой функции.  $\square$

Далее представлен алгоритм решения задачи  $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$ , и дается объяснение, как он может быть использован для решения других рассматриваемых задач.

Предположим, что требования в  $N_1$  и  $N_2$  упорядочены в соответствии с леммой 3.4. В алгоритме одно за другим рассматриваются требования  $i_1, i_2, \dots, i_{n_2}$ . Для каждого требования  $i_k$ ,  $k = 1, 2, \dots, n_2$ , необходимо рассмотреть все позиции  $l$ ,  $l = 0, 1, 2, \dots, n_1$ , где позиция  $l$  означает, что требование  $j_l$  предшествует требованию  $i_k$  в конструируемом расписании, и  $i_k$  предшествует требованию  $j_{l+1}$ . Если для требования  $i_k$  выбрана позиция  $l$ , то для требования  $i_{k+1}$  могут быть рассмотрены только позиции  $l, l+1, \dots, n_1$  (см. лемму 3.4).

Здесь  $(F, \pi_{opt}) := Sequence(1, 0, -p)$ , где  $\pi_{opt}$  — оптимальная последовательность обслуживания требований,  $F = C_{max}^*$  — минимальное время окончания обслуживания всех требований.

Оценим трудоемкость алгоритма. Множество требований для которых еще не составлено расписание, являющиеся аргументами рекурсивной процедуры, имеют вид

$$\{j_{l+1}, j_{l+2}, \dots, j_{n_1}, i_k, i_{k+1}, \dots, i_{n_2}\},$$



---

**Algorithm 2** Function  $Sequence(k, l, S_{i_{k-1}})$ 

---

```
1: if  $k = n_2 + 1$  then
2:   Назначить обслуживание требований  $j_{l+1}, j_{l+2}, \dots, j_{n_1}$  с момента времени
    $S_{i_{k-1}} + p + st_1$  согласно алгоритму (3.2);
3:    $\sigma := (j_{l+1}, j_{l+2}, \dots, j_{n_1})$ 
4:   Вернуть пару  $(C_{j_{n_1}}, \sigma)$ ;
5: end if
6: if  $l = n_1$  then
7:   Назначить обслуживание требований  $i_k, i_{k+1}, \dots, i_{n_2}$  с момента времени  $S_{i_{k-1}} +$ 
    $p$  согласно алгоритму (3.2);
8:    $\sigma := (i_k, i_{k+1}, \dots, i_{n_2})$ 
9:   Вернуть пару  $(C_{i_{n_2}}, \sigma)$ ;
10: end if
11:  $S := S_{i_{k-1}}$ ;
12:  $f_{min} := \infty$ ;
13:  $\sigma_{min} := ()$ ;
14: for  $pos := l$  to  $n_1$  do
15:   if  $pos = l$  then
16:      $S_{i_k} := \max\{r_{i_k}, S + p\}$ ;
17:      $S := S + p + st_1$ ;
18:      $(f, \sigma) := Sequence(k + 1, l, S_{i_k})$ ;
19:   else
20:      $S_{j_{pos}} := \max\{r_{j_{pos}}, S\}$ ;
21:      $S_{i_k} := \max\{r_{i_k}, S_{j_{pos}} + p + st_2\}$ ;
22:      $(f, \sigma) := Sequence(k + 1, pos, S_{i_k})$ ;
23:      $S := S_{j_{pos}} + p$ ;
24:   end if
25:   if  $f_{min} > f$  then
26:      $f_{min} := f$ ;
27:      $\sigma_{min} := (j_{l+1}, \dots, j_{pos}, i_k, \sigma)$ 
28:   end if
29: end for
30: Вернуть пару  $(f_{min}, \sigma_{min})$ ;
```

---

т.е. множества точно определяются парой индексов  $(k, l)$ . Аргументы  $S_{i_{k-1}}$  принадлежат множеству  $\Theta$ . Следовательно, процедура  $Sequence(k, l, S_{i_{k-1}})$  будет вызвана не более  $O(n^6)$  раз. Трудоемкость процедуры составляет  $O(n)$  операций. Таким образом, трудоемкость алгоритма 2 составляет  $O(n^7)$  операций.

В соответствии с леммой 3.4 алгоритм 2 строит оптимальную последовательность обслуживания требований за время  $O(n^7)$ .

Процедура может быть легко модифицирована для решения задачи  $STRSP2|r_j|C_{max}$ . Для этого необходимо изменить строки 4 и 9 процедуры *Sequence*:

4. Вернуть пару  $(C_{j_{n_1}} + TAIL_{left}, \sigma)$ ;

...

9. Вернуть пару  $(C_{i_{n_2}} + TAIL_{right}, \sigma)$ ;

Для решения задачи  $1|setup - times, N_1, N_2, p_j = p, r_j| \sum C_j$  в процедуру необходимо внести следующие изменения:

4. Вернуть пару  $(\sum_{x:=l+1}^{n_1} C_{j_x}, \sigma)$ , где  $C_{j_x}$  — время окончания обслуживания требования  $j_x$  при частичном расписании, полученном из последовательности  $\sigma$ , где требования обслуживаются с момента времени  $S_{i_{k-1}} + p + st_1$ ;

...

9. Вернуть пару  $(\sum_{x:=k}^{n_2} C_{i_x}, \sigma)$ , где  $C_{i_x}$  — время окончания обслуживания требования  $i_x$  при частичном расписании, полученном из последовательности  $\sigma$ , где требования обслуживаются, начиная с  $S_{i_{k-1}} + p$ ;

...

25. **If**  $f_{min} > f + f_{current}$  **Then**,// где  $f_{current}$  — общее время окончания обслуживания требований из последовательности  $(j_{l+1}, \dots, j_{pos}, i_k)$ , обслуживание которых начинается с момента времени  $S_{i_{k-1}} + p$ ;

26.  $f_{min} := f + f_{current}$ ;

Напомним, что для задачи  $1|setup - times, N_1, N_2, p_j = p, r_j| \sum C_j$  требования множеств  $N_1$  и  $N_2$  должны быть упорядочены согласно лемме 3.4.

Аналогично процедура может быть модифицирована для решения задач  $1|setup - times, N_1, N_2, p_j = p| \sum w_j C_j$  и  $1|setup - times, N_1,$

$N_2, p_j = p | \sum T_j$ . Заметим, что для этих двух задач  $|\Theta| = O(n^3)$ , т.к. все моменты появления работ равны 0, т.е. сложность модифицированных алгоритмов будет равна  $O(n^6)$  операций. Таким образом, имеем следующую лемму.

**Лемма 3.5.** С помощью алгоритма 2 и его модификаций следующие задачи могут быть решены за время  $O(n^7)$  или  $O(n^6)$ :

- $1|setup - times, N_1, N_2, p_j = p, r_j|C_{max}$  и  $STRSP2|r_j|C_{max}$ ;
- $1|setup - times, N_1, N_2, p_j = p, r_j| \sum C_j$  и  $STRSP2|r_j| \sum C_j$ ;
- $1|setup - times, N_1, N_2, p_j = p| \sum w_j C_j$  и  $STRSP2|| \sum w_j C_j$ ;
- $1|setup - times, N_1, N_2, p_j = p| \sum T_j$  и  $STRSP2|| \sum T_j$ .

### 3.2.3 Задачи с частично упорядоченными подмножествами

**Лемма 3.6.** Для задачи  $1|setup - times, N_1, N_2, p_j = p| \sum w_j U_j$  существует оптимальное “сдвинутое влево” расписание, в котором все запаздывающие требования из одного и того же множества  $N_1$  или  $N_2$  упорядочены по неубыванию директивных сроков, т.е.  $d_{j_1} \leq d_{j_2} \leq \dots \leq d_{j_{n_1}}$  и  $d_{i_1} \leq d_{i_2} \leq \dots \leq d_{i_{n_2}}$ .

**Лемма 3.7.** Предположим, что требования упорядочены в соответствии с леммой 3.6. Для задачи  $1|setup - times, N_1, N_2, p_j = p| \sum U_j$  существует оптимальное “сдвинутое влево” расписание и такие индексы  $x, 1 \leq x \leq n_1$  и  $y, 1 \leq y \leq n_2$ , что только требования  $j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}$  не запаздывают и обслуживаются в порядке согласно лемме 3.6.

Обе леммы могут быть доказаны по аналогии с леммой 3.4.

Таким образом, для задачи  $1|setup - times, N_1, N_2, p_j = p| \sum U_j$  необходимо найти индексы  $x$  и  $y$  такие, что  $x + y \rightarrow \min$  и требования  $j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}$  могут быть выполнены вовремя (без запаздывания) в начале расписания. Следовательно, нужно рассмотреть

не более  $(n_1 + 1) \log(n_2 + 1)$  пар  $(x, y)$ . Для каждой из пар решается задача  $1|setup - times, N_1, N_2, p_j = p| \sum T_j$  с множеством требований  $\{j_x, j_{x+1}, \dots, j_{n_1}, i_y, i_{y+1}, \dots, i_{n_2}\}$  с помощью модификации алгоритма 2. Если  $\sum T_j = 0$ , то пара  $(x, y)$  допустима.

**Лемма 3.8.** Задача  $1|setup - times, N_1, N_2, p_j = p| \sum U_j$  разрешима за время  $O(n^7 \log n)$ .

Для задачи  $1|setup - times, N_1, N_2, p_j = p| \sum w_j U_j$  предлагается полиномиальный алгоритм динамического программирования. Алгоритм основан на следующих предположениях. Обозначим требования из  $N = \{H_1, H_2, \dots, H_n\}$ , где  $w_{H_1} \leq w_{H_2} \leq \dots \leq w_{H_n}$ . Если  $w_{H_k} = w_{H_{k+1}}$ , то  $d_{H_k} \leq d_{H_{k+1}}$ . Требования из  $N_1$  и  $N_2$  обозначены и упорядочены в соответствии с леммой 3.6. Пусть  $H_n \in N_2$  и  $H_n = i_k$ . Для  $H_n$  позиция в расписании определяется парой  $(t, l)$ , где  $t \in \Theta$  — время начала обслуживания требования, а индекс  $l = 0, 1, \dots, n_1$  означает, что обслуживание незапаздывающих требований из множества  $\{j_1, j_2, \dots, j_l\}$  предшествует обслуживанию требования  $H_n$  при расписании, а незапаздывающие требования из множества  $\{j_{l+1}, j_{l+2}, \dots, j_{n_1}\}$  обслуживаются после обслуживания требования  $H_n$ . Позиция  $(-, n_1 + 1)$  означает, что требование  $H_n$  запаздывает и обслуживается в конце расписания с некоторого момента времени  $T \in \Theta$ .

Таким образом, для каждой позиции  $(t, l)$  среди  $O(n^4)$  возможных позиций, мы можем разделить исходную задачу на две независимые подзадачи:

- с множеством требований  $N_{left} = \{j_1, j_2, \dots, j_l, i_1, i_2, \dots, i_{k-1}\}$ , которые должны быть обслужены в интервале  $[0, t)$ ;
- с множеством требований  $N_{right} = \{j_{l+1}, j_{l+2}, \dots, j_{n_1}, i_{k+1}, i_{k+2}, \dots, i_{n_2}\}$ , которые должны быть обслужены в интервале  $[t + p, T)$ .

Ниже представлен алгоритм решения данной задачи.

---

**Algorithm 3** Function  $SequenceWU(h, t_1, t_2, I_1, I_2, k_1, k_2, l_1, l_2)$ 

---

```
1:  $f_{max} := -\infty$ ; // взвешенное число запаздывающих требований;
2:  $\sigma_{max} := \{\}$ ;
3: if  $H_h \in N_1$  then
4:    $I = 1$ ;
5:   if  $I_1 = 1$  then  $t_{min} := t_1$ ;
6:   if  $I_1 = 2$  then  $t_{min} := t_1 + st_2$ ;
7:   if  $I_1 = 0$  then  $t_{min} := 0$ ;
8:   if  $I_2 = 1$  then  $t_{max} := t_2 - p$ ;
9:   if  $I_2 = 2$  then  $t_{max} := t_2 - p - st_1$ ;
10:  if  $I_2 = 0$  then  $t_{max} := T_{max}$ ;
11:   $pos_1 := k_1$ ;  $pos_2 := k_2$ ;
12: else
13:    $I = 2$ ;
14:   if  $I_1 = 1$  then  $t_{min} := t_1 + st_1$ ;
15:   if  $I_1 = 2$  then  $t_{min} := t_1$ ;
16:   if  $I_1 = 0$  then  $t_{min} := 0$ ;
17:   if  $I_2 = 1$  then  $t_{max} := t_2 - p - st_2$ ;
18:   if  $I_2 = 2$  then  $t_{max} := t_2 - p$ ;
19:   if  $I_2 = 0$  then  $t_{max} := T_{max}$ ;
20:    $pos_1 := l_1$ ;  $pos_2 := l_2$ ;
21: end if
22: for  $pos := pos_1$  to  $pos_2$  do
23:   for each  $t \in \Theta$ ,  $t_{min} \leq t \leq t_{max}$ ,  $t + p < d_{H_h}$  do
24:     if  $H_h \in N_1$  then
25:       Пусть  $j_l = H_h$ ;
26:        $(\sigma_1, f_1) := SequenceWU(h - 1, t_1, t + p, I_1, I_2, k_1, pos, l_1, l_2 - 1)$ ;
27:        $(\sigma_2, f_2) := SequenceWU(h - 1, t + p, t_2, I_1, I_2, pos + 1, k_2, l_1 + 1, l_2)$ ;
28:     else
29:       Пусть  $i_k = H_h$ ;
30:        $(\sigma_1, f_1) := SequenceWU(h - 1, t_1, t + p, I_1, I_2, k_1, k - 1, l_1, pos)$ ;
31:        $(\sigma_2, f_2) := SequenceWU(h - 1, t + p, t_2, I_1, I_2, k_1 + 1, k_2, pos + 1, l_2)$ ;
32:     end if
33:     if  $f_1 + f_2 + w_{H_h} > f_{max}$  then
34:        $f_{max} := f_1 + f_2 + w_{H_h}$ ;
35:        $\sigma_{max} := \sigma_1 \cup \sigma_2 \cup \{(h, t)\}$ ;
36:     end if
37:   end for
38: end for
39: // Дополнительно рассматриваем случай, когда требование  $H_h$  запаздывает.
40:  $(\sigma_1, f_1) := SequenceWU(h - 1, t_1, t_2, I_1, I_2, k_1, k_2, l_1, l_2)$ ;
41: if  $f_1 > f_{max}$  then
42:    $f_{max} := f_1$ ;
43:    $\sigma_{max} := \sigma_1 \cup \{(h, T_{max})\}$ ;
44: end if
45: Вернуть пару  $(f_{max}, \sigma_{max})$ ;
```

---

Здесь  $T_{max} = \max\{t | t \in \Theta\}$ ,  $(F, SCHEDULE_{opt}) := SequenceWU(n, 0, T_{max}, 0, 0, 0, n_2, 0, n_1)$ , где  $SCHEDULE_{opt}$  — недопустимое расписание, которое может быть трансформировано в оптимальное путем изменения расписания для требований, обслуживание которых назначено с момента време-

ни  $T_{max}$ ,  $F = \sum w_j(1 - U_j)$  — максимальное взвешенное число незапаздывающих требований. Отметим, что для задачи  $1|setup-times, N_1, N_2, p_j = p|\sum w_j U_j$  мощность множества  $|\Theta| = O(n^3)$ , поскольку все моменты поступления требований равны 0.

Несложно оценить трудоемкость данного алгоритма. Множества неупорядоченных требований, выступающие в качестве аргументов рекурсивной процедуры, имеют вид

$$N' = \{j_{l_1}, j_{l_1+1}, \dots, j_{l_2}, i_{k_1}, i_{k_1+1}, \dots, i_{k_2}\},$$

$$N' \cap \{H_{h+1}, H_{h+2}, \dots, H_n\} = \emptyset,$$

т.е. они точно задаются пятью индексами  $h, k_1, k_2, l_1, l_2$ . Аргументы  $t_1, t_2$  принадлежат множеству  $\Theta$ . Таким образом, процедура  $SequenceWU(h, t_1, t_2, I_1, I_2, k_1, k_2, l_1, l_2)$  выполняется не более  $O(n^{5+3+3})$  раз. Трудоемкость рекурсивной процедуры составляет  $O(n^4)$  операций. Следовательно, время работы алгоритма 3 составляет  $O(n^{15})$  операций.

Табл. 3 содержит обобщенную информацию по предложенным алгоритмам и задачам, для которых они могут использоваться.

Железнодорожная задача	Соответствующая задача для одного прибора	Сложность алгоритма
$STRSP2 r_j C_{max}$	$1 setup-times, N_1, N_2, p_j = p, r_j C_{max}$	$O(n^7)$
$STRSP2 r_j \sum C_j$	$1 setup-times, N_1, N_2, p_j = p, r_j \sum C_j$	$O(n^7)$
$STRSP2  \sum w_j C_j$	$1 setup-times, N_1, N_2, p_j = p \sum w_j C_j$	$O(n^6)$
$STRSP2  \sum T_j$	$1 setup-times, N_1, N_2, p_j = p \sum T_j$	$O(n^6)$
$STRSP2  \sum U_j$	$1 setup-times, N_1, N_2, p_j = p \sum U_j$	$O(n^7 \log n)$
$STRSP2  \sum w_j U_j$	$1 setup-times, N_1, N_2, p_j = p \sum w_j U_j$	$O(n^{15})$

**Таблица 3.3. Сложность предложенных алгоритмов для разных типов задач**

### 3.3 Задача минимизации максимального взвешенного временного смещения выполнения заказа для двух станций

Имеется две станции, соединенные двухпутной железной дорогой. Необходимо выполнить множества заказов  $N^1 = \{J_1^1, \dots, J_n^1\}$  и  $N^2 = \{J_1^2, \dots, J_n^2\}$  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  $N^2$  составляются отдельно. Рассматриваемое множество для простоты обозначим  $N = \{J_1, \dots, J_n\}$ . 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  $\alpha$  — время, которое должно разделять моменты отправки двух поездов. Движение пассажирских поездов и плановые ремонтные работы путей учитываются в модели как допустимые интервалы отправки  $[\tau_1, \tau'_1], [\tau_2, \tau'_2], \dots, [\tau_z, \tau'_z]$ . Поезда могут быть отправлены только в момент времени, принадлежащий одному из этих интервалов.

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

$$L_{max} = \max_{j=\overline{1,n}}(w_j(C_j - d_j)).$$

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

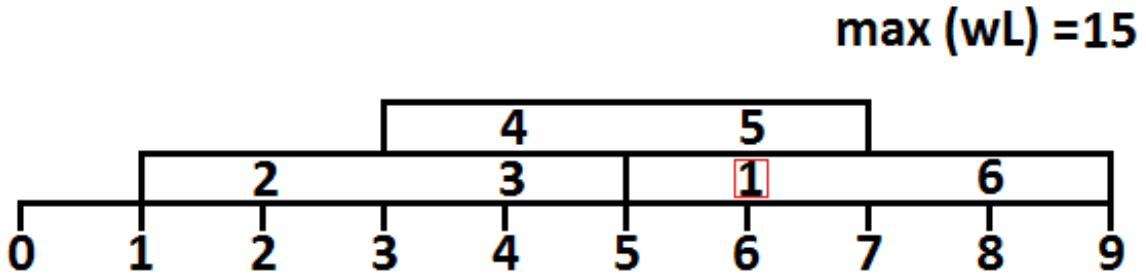


Рис. 3.3. Оптимальное расписание примера 1

**Пример 3.3.0.1.** 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 рис. 3.3.

Здесь начало прямоугольника — время отправки поезда, конец — время прибытия. Цифры внутри прямоугольника обозначают номера заказов, перевозимых поездом. Значение целевой функции равно 15 и достигается на заказе  $J_1$ .

### 3.3.1 Задача минимизации общего времени выполнения заказов при ограничении на максимальное взвешенное временное смещение

Решим вспомогательную задачу. Пусть задано множество заказов  $N$  и положительное число  $y$ . Задача заключается в составлении расписания  $\Theta(N, y)$  удовлетворяющего условию

$$\min \max_{j=1,n} (C_j) \mid \max_{j=1,n} (w_j(C_j - d_j)) < y.$$

Введем дополнительные обозначения. Для данного значения  $y$  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 с момента отправки заказа до прибытия его на станцию назначения требуется время  $p$ , то получаем условие на момент отправки заказа  $j$ :

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

Следовательно, 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 на дорогу время, равное  $p$ , так чтобы каждый заказ был доставлен поездом, отправляющемся в промежутке  $[r_j, t'_j)$ .

## Алгоритм решения задачи

Введем семейство множеств  $S_1 \subset S_2 \subset \dots \subset S_q$  — заказы, которые должны уехать первым поездом ( $S_1$ ), первым или вторым поездом ( $S_2$ ), первым или вторым или третьим ( $S_3$ ) и т.д. Заметим, что до начала работы алгоритма построения  $\Theta(N, y)$  не пусто только множество  $S_q$ , которое состоит из всех  $n = kq$  заказов. 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  $X$ .

Для поезда с номером  $m$ , имеющего текущее значение момента отправки  $t^m$ , обозначим за  $t_\alpha^m$  наименьший момент времени, принадле-

жащий допустимому интервалу отправки, для которого выполняется  $t_\alpha^m \geq t^m + \alpha$ .

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

**Свойство 1.** В расписании  $\Theta(N, y)$  момент времени  $t^m$  отправки поезда с номером  $m$  должен удовлетворять следующим условиям:

- (i) если в момент отправки доступно больше чем  $k$  вагонов, то отправляются вагоны с меньшим значением  $t'$ ;
- (ii) поезд с номером  $m$  не может отправиться раньше момента  $t^m \geq \max(r_{k \cdot m}, r(S_m), t_\alpha^{m-1})$ ;
- (iii) все заказы  $J_l$ , для которых выполняется  $t_\alpha^m \geq t'_l$  должны уехать одним из поездов с номерами от 1 до  $m$  т.е.  $J_l \in S_m$ ;
- (iv) после отправки  $m$ -ого 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 в одном поезде с заказом  $b$  либо более поздними поездами.

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

- поступит  $k \cdot m$  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  $t_\alpha^m$ , а значит, все заказы  $l$ , для которых выполняется неравенство  $t_\alpha^m \geq t'_l$ , не смогут отправиться поездами, номер которых больше  $m$ . Условие (iv) следует из определения множества  $S_m$ .

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

Алгоритм построения расписания  $\Theta(N, y)$  заключается в следующем. На каждом шаге алгоритма производится попытка отправить некоторый поезд  $m$ . Для этого выбирается 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  $t_\alpha^m \geq t'_l$ , 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 заказов  $T_{m-1}, T_{m-2}, \dots$  искать такие заказы  $j$ , что  $t'_j > t_\alpha^m$ , до тех пор, пока их количество не достигнет значения  $x - k$ . Пусть последний такой заказ был найден на поезде с номером  $s$ . Введем обозначение  $T_s^{m-1} = T_s \cup \dots \cup T_{m-1}$ .

Пусть  $X'$  — множество, состоящее из  $x - k$  заказов  $j$ ,  $j \in T_s^{m-1}$ , для которых верно  $t'_j > t_\alpha^m$ , имеющих минимальные моменты  $t'$  из всех таких  $j \in T_s^{m-1}$ .

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  $r_{k \cdot i}$  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  $t' > t_\alpha^m$ . В случае противоречия условию (iii) происходит изменение соответствующего множества  $S_i$ , осуществляется переход к поезду  $i$ , и начинается новый шаг.

Если противоречий не встретилось, то отправляем по этим правилам поезд 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  $k \cdot i$ .

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  $\Theta(N, y)$ .

### Сложность алгоритма

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  $k + 2k + \dots + (q - 1)k = k \frac{(q-1)(q-2)}{2}$ , т.е. будем иметь  $O(q^2k) = O(\frac{n^2}{k})$  операций. Тем самым трудоемкость алгоритма, строящего расписание  $\Theta(N, y)$ , составляет  $O(\frac{n^4}{k})$  операций.

### 3.3.2 Решение задачи минимизации максимального взвешенного временного смещения

Для построение 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 поез-

да  $i$ . Рассмотрим заказ  $j_1$ , 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 идёт поездом  $m_2$ .

Следующим шагом алгоритма построим расписание  $\Theta(N, w_{j_2}L_{j_2})$ , в котором заказ  $J_{j_2}$  должен быть отправлен 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 меньшим, чем  $w_{j_s}L_{j_s}$ , и при этом существует расписание, полученное на  $s$ -м шаге со значением целевой функции  $w_{j_s}L_{j_s}$ , следовательно, оно и будет оптимальным. 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  $|S_m| > k \cdot m$ , прерывает работу алгоритма.

### Алгоритм $\pi(N)$

**0.** Построить расписание, в котором заказы отправляются по возрастанию моментов поступления.

**1.** Положить  $y := \max_{j \in N} (w_j L_j)$ . Найти  $j : w_j L_j = y$ ,  $m : J_j \in T_m$  и запустить  $Add_S(j, m - 1)$ .

**2.** Запустить Алгоритм  $\Theta(N, y)$  и перейти на шаг 1.

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 выход заказа из каждого из этих множеств 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  $y < \beta$ , при этом 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 производятся 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  $O(kq^2) = O(n^2/k)$ . На каждом шаге мы получаем расписание  $\Theta$ , для построения которого необходимо  $O(n^3/k)$  операций. Следовательно, общая трудоёмкость составляет  $O(n^6/k^2)$  операций.

Продemonстрируем работу алгоритма на примере расписания, построенного в примере 3.3.0.1.

**Шаг 1.** Строим для данного множества заказов расписание, удовлетворяющее условию  $\min(C_{\max})$ , находим значение целевой функции  $\max_{i=1, n}(w_i L_i)$  и принимаем его за  $y = 30$  (см. рис. 3.4).

**Шаг 2.** Строим расписание  $\Theta(N, y)$  для множества  $N = \{J_1, \dots, J_6\}$  и  $y = 30$ . Определяем моменты  $t'_i$ ,  $i = 1, \dots, 6$ :  $t'_1 = 10, t'_2 = 3, t'_3 = t'_4 = t'_5 = 4, t'_6 = 9$ . 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. 3.5).

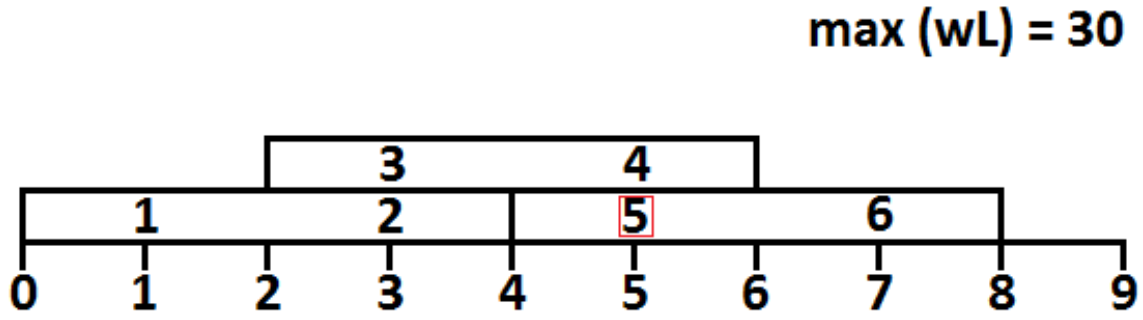


Рис. 3.4. Расписание  $\min(C_{max})$  для множества заказов  $N$

В соответствии с условием (ii) получаем  $t^2 = 2$ , однако есть три заказа,

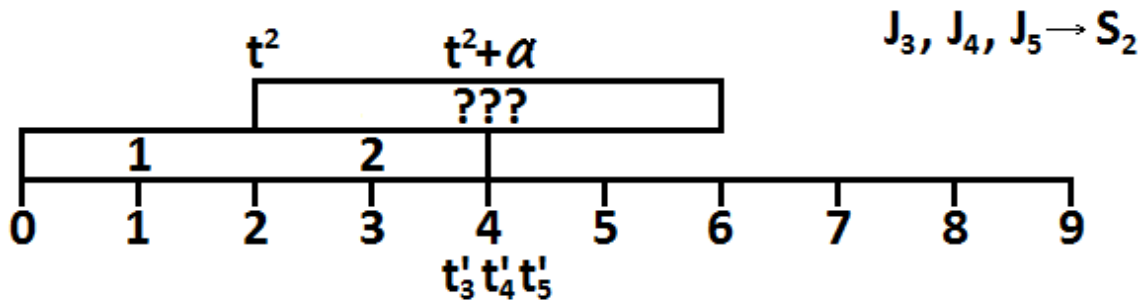


Рис. 3.5. Противоречие свойству (iii) при отправке второго поезда

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  $\Theta(N, y)$ , при котором целевая функция  $\max(wL) = 15$  достигается на заказе  $J_1$  (см. рис. 3.6).

**Шаг 3.** Строим расписание  $\Theta(N, y)$  для значения  $y = 15$ . Имеем

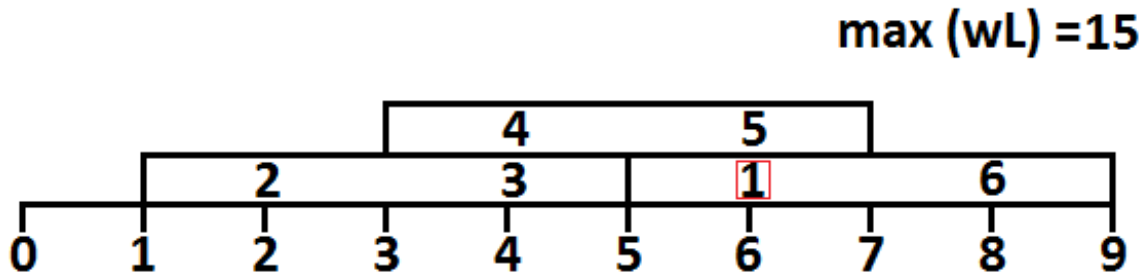


Рис. 3.6. Расписание  $\Theta(N, 30)$

$t'_1 = 5, t'_2 = 1.5, t'_3 = t'_4 = 2.5, t'_5 = 3, 5t'_6 = 6$ . 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,  $t^2 \leq 3$ . 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.

Следовательно, расписание, полученное на шаге 2, будет оптимальным, а значение целевой функции будет равно 15.

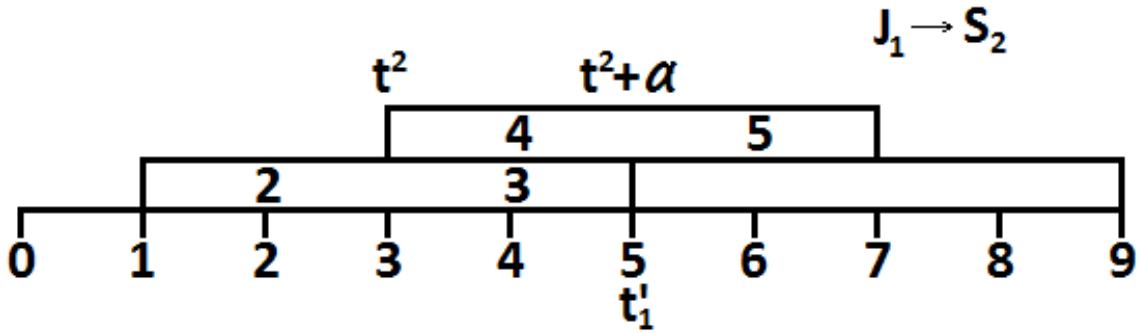


Рис. 3.7. Противоречие свойству (iii) при построении расписания  $\Theta(N, 15)$

### Упражнения

С помощью рассмотренного алгоритма построить расписания для задач со следующими параметрами:

**Упр. 3.3.2.1.**  $n = 9$ ,  $k = 3$ ,  $\alpha = 1$ ,  $p = 3$ ,  $\delta = 4$ ,  $[\tau_1, \tau'_1] = [0, 2]$ ,  $[\tau_2, \tau'_2] = [4, +\infty)$ ,  $r = (0, 0, 0, 0, 1, 1, 2, 3, 3)$ ,  $w = (2, 2, 2, 1, 2, 2, 5, 2, 2)$ .

**Упр. 3.3.2.2.**  $n = 8$ ,  $k = 2$ ,  $\alpha = 1$ ,  $p = 2$ ,  $\delta = 2$ ,  $[\tau_1, \tau'_1] = [0, 2]$ ,  $[\tau_2, \tau'_2] = [4, 4.5]$ ,  $[\tau_3, \tau'_3] = [5.5, +\infty]$ ,  $r = (0, 0, 1, 2, 3, 3, 4, 4)$ ,  $w = (1, 1, 5, 5, 1, 1, 2, 2)$ .



**Упр. 3.3.2.3.**  $n = 8$ ,  $k = 2$ ,  $\alpha = 1$ ,  $p = 2$ ,  $\delta = 3$ ,  $[\tau_1, \tau'_1] = [0, +\infty]$ ,  
 $r = (0, 0, 0, 1, 1, 1, 2, 4)$ ,  $w = (2, 3, 2, 4, 3, 3, 4, 4)$ .

### 3.4 Полиномиальные алгоритмы задач о станциях на основе сведения к задачам теории расписания для одного прибора

В работе [14] описан полиномиальный алгоритм решения класса задач теории расписаний для одного прибора со следующими параметрами:

- требования  $j$  поступают на обслуживание в моменты времени  $r_j > 0$ ;
- продолжительность обслуживания всех требований равна некоторой константе  $p$ ;
- требования обслуживаются в группах. Группой считается набор требований, обслуживаемых “одновременно” с той точки зрения, что моменты начала и завершения обслуживания всех требований в одной группе совпадают между собой.

В задаче рассматривается два механизма объединения обслуживания требований в группах.

- Последовательные группы, когда продолжительность обслуживания (разность между моментами начала и завершения обслуживания) требований в группе с  $k$  требованиями равна  $kp$ , т.е. требования в одной группе обслуживаются последовательно одно за другим. Перед началом обслуживания требований новой группы требуется время для переналадки прибора  $s$ .
- Параллельные группы, когда продолжительность обслуживания требований равна  $p$ , т.е. требования в одной группе обслуживаются одновременно. Задано максимальное количество  $b < n$  требований,

которые могут быть объединены в одну группу, где  $n$  — общее количество требований.

Для каждого конкретного примера задач указывается, какой из механизмов (последовательный или параллельный) используется для объединения требований в группы.

В качестве целевого функционала рассматривается целый класс функций  $F$ , называемых *упорядоченными целевыми функциями*, со следующими свойствами:

- $F$  является прямой суммой некоторых функций от моментов завершения обслуживания, то есть  $F = \sum_i f_i(C_i)$ ;
- $F$  является регулярной функцией, то есть  $\forall i$   $f_i$  является неубывающей функцией;
- для каждой функции  $f_i$  существует такой момент времени  $\delta_i$ , что  $\forall t \geq \delta_i$  выполняется  $f(t) = f(\delta_i)$ ;
- для любых  $i < j$ ,  $\delta_i \leq \delta_j$  и  $t$ , функция-разность  $(f_i - f_j)(t)$  является неубывающей на отрезке  $[0, \delta_i]$ .

Без ограничения общности, предположим, что требования пронумерованы так, что  $\delta_1 \leq \delta_2 \leq \dots \leq \delta_n$ .

Согласно нотации Грэхема и др. [35], такие рассматриваемые задачи обозначаются следующим образом:  $1|s - batch, r_j, p_j = p|F$  (задача для последовательных групп) и  $1|p - batch, b < n, r_j, p_j = p|F$  (задача для параллельных групп).

Условиям упорядоченных целевых функций удовлетворяют следующие критерии: суммарное взвешенное количество запаздывающих требований  $(\sum w_j U_j)$ , суммарный взвешенный момент завершения обслуживания  $(\sum w_j C_j)$ , суммарное запаздывание  $(\sum T_j)$ .

Для указанных задач в последовательном и параллельном случаях в работе [14] предложены алгоритмы динамического типа нахождения

точного решения за полиномиальное время. В случае последовательных групп трудоемкость алгоритма составляет  $O(n^{14})$  операций, в случае параллельных групп –  $O(n^8)$  операций.

Идея алгоритма основана на следующих свойствах оптимальных расписаний.

**Лемма 3.9.** Для случая последовательных групп в оптимальном расписании обслуживание требований начинается и заканчивается в моменты времени из множества

$$\mathcal{S} = \{r_\lambda + \mu p + \nu s, \lambda \in \{1, \dots, n\}, \mu \in \{0, \dots, n\}, \nu \in \{0, \dots, n\}\}.$$

Для случая параллельных групп – из множества

$$\mathcal{P} = \{r_\lambda + \mu p, \lambda \in \{1, \dots, n\}, \mu \in \{0, \dots, n\}\}.$$

Количество элементов в множествах  $\mathcal{S}$  и  $\mathcal{P}$  равно, соответственно,  $O(n^3)$  и  $O(n^2)$ .

Введем понятие *частичных групп*. Частичной группой называется группа из  $\nu$  требований, в которой в действительности упорядочено  $\mu$  требований,  $\mu \leq \nu$ . То есть, в частичных группах возможно появление *пустых* мест, в которых нет никакого требования. Можно сказать, что в этих пустых местах прибор простаивает, но время таких простоев включается в общее время обслуживания требований данной группы.

Пусть  $U_k(t_l, t_r)$ ,  $k \leq n$  и  $t_l \leq t_r$ , обозначает множество требований  $j$ , для которых  $j \leq k$  и  $r_j \in (t_l, t_r]$ .

В случае последовательных групп переменными динамического программирования являются  $k, t_l, t_r, \nu_l, \nu_r, \mu_r$ . Каждая их комбинация определяет подзадачу для требований из множества  $U_k(t_l, t_r)$ , в которой

- частичная группа, содержащая 0 требований из  $U_k(t_l, t_r)$ , начинается в  $t_l$  и завершается в  $t_l + p\nu_l$ ;
- частичная группа, содержащая  $\mu_r$  требований из  $U_k(t_l, t_r)$ , начинается в  $t_r$  и завершается в  $t_r + p\nu_r$ .

Пусть  $S_k(t_l, t_r, \nu_l, \nu_r, \mu_r)$  является минимальным значением целевой функции для указанной подзадачи с последовательными группами. Если расписания, удовлетворяющего условиям подзадачи не существует, то данная величина принимает значение  $+\infty$ .

Аналогично, в случае параллельных групп переменными динамического программирования являются  $k, t_l, t_r, \mu_r$ . Их комбинация определяет подзадачу для требований из множества  $U_k(t_l, t_r)$ , в которой частичная группа, содержащая  $\mu_r$  требований из  $U_k(t_l, t_r)$ , начинает обслуживание в момент времени  $t_r$ .

Пусть  $P_k(t_l, t_r, \mu_r)$  является минимальным значением целевой функции для указанной подзадачи с параллельными группами. Если расписания, удовлетворяющего условиям подзадачи, не существует, то данная величина принимает значение  $+\infty$ .

Работа метода динамического программирования для рассматриваемых задач основана на следующих условиях (реализациях принципа Беллмана).

**Лемма 3.10.** Для последовательных групп выполняется

$$S_k(t_l, t_r, \nu_l, \nu_r, \mu_r) = \min\{L, I, R\},$$

где

$$L = S_{k-1}(t_l, t_r, \nu_l, \nu_r, \mu_r) + f_k(\delta_k),$$

$$I = \min_{\nu, t} \{S_{k-1}(t_l, t, \nu_l, \nu, \nu - 1) + f_k(t + p\nu) + S_{k-1}(t, t_r, \nu, \nu_r, \mu_r)\},$$

$$R = f_k(t_r + p\nu_r) + S_{k-1}(t_l, t_r, \nu_l, \nu_r, \mu_r - 1) \text{ если } \mu_r > 0, \text{ иначе } +\infty.$$

При этом,  $S_0(t_l, t_r, \nu_l, \nu_r, \mu_r)$  принимается равным нулю, если  $t_l + p\nu_l + s \leq t_r$  и  $+\infty$  в противном случае.

**Лемма 3.11.** Для параллельных групп выполняется

$$P_k(t_l, t_r, \mu_r) = \min\{L, I, R\},$$

где

$$L = P_{k-1}(t_l, t_r, \mu_r) + f_k(\delta_k),$$

$$I = \min_t \{P_{k-1}(t_l, t, b-1) + f_k(t+p) + P_{k-1}(t, t_r, \mu_r)\},$$

$$R = f_k(t_r + p) + P_{k-1}(t_l, t_r, \mu_r - 1) \text{ если } \mu_r > 0, \text{ иначе } +\infty.$$

При этом,  $P_0(t_l, t_r, \mu_r)$  принимается равным нулю, если  $t_l + p \leq t_r$  и  $+\infty$  в противном случае.

Таким образом, данные условия определяют следующие алгоритмы динамического построения величин  $S_k$  и  $P_k$  (Алгоритмы 4 и 5).

---

**Algorithm 4** Вычисление величин  $S_k(t_l, t_r, \nu_l, \nu_r, \mu_r)$

---

```

1: for  $k = 1 \rightarrow n$  do
2:   for  $\nu_r = 0 \rightarrow n$  do
3:     for  $\nu_l = 0 \rightarrow n$  do
4:       for  $\mu_r = 0 \rightarrow \nu_r$  do
5:         for  $t_r \in \mathcal{S}$  do
6:           for  $t_l \in \mathcal{S}(t_l \leq t_r)$  do
7:             Вычислить значения  $L, I, R$ ;
8:              $S_k(t_l, t_r, \nu_l, \nu_r, \mu_r) \leftarrow \min\{L, I, R\}$ ;
9:           end for
10:        end for
11:      end for
12:    end for
13:  end for
14: end for

```

---



---

**Algorithm 5** Вычисление величин  $P_k(t_l, t_r, \mu_r)$

---

```

1: for  $k = 1 \rightarrow n$  do
2:   for  $\mu_r = 0 \rightarrow b$  do
3:     for  $t_r \in \mathcal{P}$  do
4:       for  $t_l \in \mathcal{P}(t_l \leq t_r)$  do
5:         Вычислить значения  $L, I, R$ ;
6:          $P_k(t_l, t_r, \mu_r) \leftarrow \min\{L, I, R\}$ ;
7:       end for
8:     end for
9:   end for
10: end for

```

---

Трудомкость применения указанных алгоритмов требует  $O(n^{14})$  операций в случае последовательных групп, и  $O(n^8)$  в случае параллельных групп.

Сформулированные в разделах 2.1.2–2.1.5 задачи о станциях могут быть представлены в виде рассматриваемых задач теории расписаний для случая параллельных групп. В данном представлении в качестве требований выступают вагоны, которые необходимо доставить с одной станции на другую. Обслуживание требований в группе означает включение вагонов в состав, продолжительность обслуживания каждого требования равна времени движения состава от одной станции к другой.

В случае с задачей о двух станциях предложенный в [14] алгоритм получения точного решения для параллельных групп применяется непосредственным образом. Таким образом, задача с двумя станциями разрешима за  $O(n^8)$  операций.

Для вариантов из  $m$  станций (расположенных последовательно, замкнутых между собой и для варианта расположения звездой) алгоритм получения точного решения представляет собой расширение основного подхода из [14] на случай динамического перебора решений для частных подзадач из двух станций. Трудоемкость такого подхода оценивается в  $O(n^{8m})$  операций и является полиномиальной величиной при фиксированном значении  $m$ .

Отметим, что качественным выводом из полученных результатов является теоретическая *полиномиальная разрешимость* задач о станциях из разделов 2.1.2–2.1.5. Применение полученных алгоритмов для решения задач большой размерности представляется невозможно в силу, хотя и полиномиальной, но большой трудоемкости решения.

## 3.5 Приближённые методы решения задач железнодорожного планирования

Некоторые из сформулированных выше задач являются NP-трудными, в связи с чем возникает проблема поиска приближенных решений в таких задачах. С другой стороны, как было показано в предыдущих параграфах, рассмотренные задачи железнодорожного планирования могут быть сведены к классическим задачам теории расписаний.

Для нахождения решений с гарантированной точностью в похожих задачах теории расписаний хорошо зарекомендовал себя метод изменения параметров [6], которому посвящён следующий параграф.

### 3.5.1 Метод изменения параметров

Рассмотрим задачу теории расписаний, характеризующуюся набором параметров  $\Omega = \{\omega_1, \omega_2, \dots, \omega_m\}$ , к примеру, для задачи минимизации суммарного запаздывания в роли параметров выступают времена поступления заказов  $r_j$ , продолжительности обслуживания  $p_j$  и директивные сроки  $d_j$ ,  $j = 1, \dots, n$ , всего  $m = 3n$  параметров.

Индивидуальный пример  $A$  задачи полностью характеризуется значениями  $m$  параметров  $\omega \in \Omega$  и потому может быть рассмотрен как точка в  $m$ -мерном пространстве. Значения параметров, относящихся к примеру  $A$  будем обозначать  $\omega^A$ . Значения целевой функции примера  $A$  при некотором расписании  $\pi$  будем обозначать через  $F^A(\pi)$ .

**Лемма 3.12.** Пусть функция  $\rho(A, B)$ , определенная в пространстве параметров  $\Omega$ , при любом расписании  $\pi$  удовлетворяет неравенству

$$(3.3) \quad |F^A(\pi) - F^B(\pi)| \leq \rho(A, B),$$

тогда

$$(3.4) \quad F^A(\pi^B) - F^A(\pi^A) \leq 2\rho(A, B),$$

где  $\pi^A$  и  $\pi^B$  — оптимальные расписания примеров  $A$  и  $B$  соответственно.

**Доказательство.**

$$\begin{aligned} F^A(\pi^B) - F^A(\pi^A) &\leq |F^A(\pi^B) - F^B(\pi^B)| + (F^B(\pi^B) - F^B(\pi^A)) + \\ &+ |F^B(\pi^A) - F^A(\pi^A)| \leq 2\rho(A, B) + (F^B(\pi^B) - F^B(\pi^A)) \leq 2\rho(A, B). \square \end{aligned}$$

В ряде случаев функция  $\rho(A, B)$  удовлетворяет аксиомам метрики и может быть рассмотрена как расстояние между примерами  $A$  и  $B$ .

Пусть введена метрика  $\rho$ , задающая расстояние между двумя примерами с различными исходными параметрами и  $A$  — полиномиально неразрешимый пример. Схема нахождения приближенного решения заключается в выполнении следующих двух шагов. На первом шаге параметры примера  $A$  изменяются таким образом, чтобы полученный пример  $B$  принадлежал полиномиально (псевдополиномиально) разрешимому классу примеров. На втором шаге для решения примера  $B$  используется известный для данного класса примеров полиномиальный (псевдополиномиальный) алгоритм, затем полученное расписание применяется к примеру  $A$ .

Согласно Лемме 3.12 абсолютная погрешность найденного решения не будет превышать удвоенного расстояния  $\rho(A, B)$  между примерами  $A$  и  $B$ . Ясно, что минимальная оценка погрешности будет получена в случае, когда на первом шаге схемы будет найден пример  $B$  минимизирующий значение функции  $\rho(A, B)$ . Таким образом, решение задачи сводится к отысканию для заданного примера  $A$  наиболее близкого примера из заданного полиномиально разрешимого класса.

Далее будет рассмотрен вариант схемы для задачи суммарного запаздывания  $1|r_j| \sum T_j$ .

Выберем в качестве целевой функции  $F(\pi)$  выражение:

$$(3.5) \quad \min \sum_{j \in N} T_j(\pi) = \min \sum_{j \in N} \max\{0, C_j(\pi) - d_j\},$$



где  $C_j(\pi)$  — момент окончания обслуживания требования  $j$  при расписании  $\pi$ . Для сокращения записи будем опускать  $(\pi)$ , если понятно о каком расписании идет речь.

Расписание  $\pi = (j_1, \dots, j_n)$  называется ранним, если при этом расписании каждое требование  $j_k, k \in N$  начинает обслуживаться в наиболее ранний допустимый момент времени: либо в момент его поступления  $r_{j_k}$ , либо в момент окончания обслуживания предыдущего требования  $C_{j_{k-1}}$ . Очевидно, что оптимальное расписание является ранним. В дальнейшем будут рассматриваться только ранние расписания.

**Лемма 3.13.** Пусть у примеров  $A$  и  $B$  совпадают времена поступления и продолжительности обслуживания требований:

$$\begin{aligned} r_j^A &= r_j^B = r_j, \\ p_j^A &= p_j^B = p_j, j \in N. \end{aligned}$$

тогда для любого расписания  $\pi$  справедливо неравенство

$$(3.6) \quad \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |d_j^A - d_j^B|.$$

**Доказательство.**

$$\begin{aligned} \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \sum_{j \in N} |\max\{0, C_j - d_j^A\} - \max\{0, C_j - d_j^B\}| \leq \\ &\leq \sum_{j \in N} \max\{0, |d_j^B - d_j^A|\} \leq \sum_{j \in N} |d_j^B - d_j^A|, \end{aligned}$$

где было использовано известное неравенство:

$$|\max\{a, b\} - \max\{c, d\}| \leq \max\{|a - c|, |b - d|\}$$

для любых действительных чисел  $a, b$  и  $c, d$ . □

**Лемма 3.14.** Пусть у примеров  $A$  и  $B$  совпадают времена поступления

и директивные сроки требований:

$$\begin{aligned} r_j^A &= r_j^B = r_j, \\ d_j^A &= d_j^B = d_j, j \in N. \end{aligned}$$

тогда для любого расписания  $\pi$  справедливо неравенство

$$(3.7) \quad \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq n \sum_{j \in N} |p_j^A - p_j^B|.$$

**Доказательство.**

$$\begin{aligned} (3.8) \quad \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \sum_{j \in N} |\max\{0, C_j^A - d\} - \max\{0, C_j^B - d_j\}| \leq \\ &\leq \sum_{j \in N} \max\{0, |C_j^A - C_j^B|\} \leq \sum_{j \in N} |C_j^A - C_j^B|. \end{aligned}$$

Для раннего расписания имеем

$$|C_{j_1}^A - C_{j_1}^B| = |r_{j_1} + p_{j_1}^A - r_{j_1} - p_{j_1}^B| = |p_{j_1}^A - p_{j_1}^B| \leq \sum_{j \in N} |p_j^A - p_j^B|.$$

Предположим, что для некоторого  $j_k$  выполняется

$$|C_{j_k}^A - C_{j_k}^B| \leq \sum_{i=1}^k |p_{j_i}^A - p_{j_i}^B|,$$

в этом случае

$$\begin{aligned} |C_{j_{k+1}}^A - C_{j_{k+1}}^B| &\leq |\max\{r_{j_{k+1}} + p_{j_{k+1}}^A, C_{j_k}^A + p_{j_{k+1}}^A\} - \\ &\quad - \max\{r_{j_{k+1}} + p_{j_{k+1}}^B, C_{j_k}^B + p_{j_{k+1}}^B\}| \leq \\ &\leq \max\{|p_{j_{k+1}}^A - p_{j_{k+1}}^B|, |C_{j_k}^A + p_{j_{k+1}}^A - C_{j_k}^B - p_{j_{k+1}}^B|\} \leq \\ &\leq \sum_{i=1}^{k+1} |p_{j_i}^A - p_{j_i}^B| \leq \sum_{j \in N} |p_j^A - p_j^B|, \end{aligned}$$

тогда из (3.8) получаем

$$|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B| \leq \sum_{j \in N} |C_j^A - C_j^B| \leq \sum_{k \in N} \sum_{j \in N} |p_j^A - p_j^B| = n \sum_{j \in N} |p_j^A - p_j^B|.$$

□

**Лемма 3.15.** Пусть у примеров  $A$  и  $B$  совпадают продолжительности обслуживания и директивные сроки требований:

$$p_j^A = p_j^B = p_j, \\ d_j^A = d_j^B = d_j, j \in N.$$

тогда для любого расписания  $\pi$  справедливо неравенство

$$(3.9) \quad |\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B| \leq n \max_{j \in N} |r_j^A - r_j^B|.$$

**Доказательство.**

$$(3.10) \quad \begin{aligned} |\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B| &\leq \sum_{j \in N} |\max\{0, C_j^A - d\} - \max\{0, C_j^B - d_j\}| \leq \\ &\leq \sum_{j \in N} \max\{0, |C_j^A - C_j^B|\} \leq \sum_{j \in N} |C_j^A - C_j^B|. \end{aligned}$$

Для раннего расписания имеем

$$|C_{j_1}^A - C_{j_1}^B| = |r_{j_1}^A + p_{j_1} - r_{j_1}^B - p_{j_1}| = |r_{j_1}^A - r_{j_1}^B| \leq \max_{j \in N} |r_j^A - r_j^B|.$$

Предположим, что для некоторого  $j_k$  выполняется

$$|C_{j_k}^A - C_{j_k}^B| \leq \max_{j \in N} |r_j^A - r_j^B|,$$

в этом случае

$$\begin{aligned}
|C_{j_{k+1}}^A - C_{j_{k+1}}^B| &\leq |\max\{r_{j_{k+1}}^A + p_{j_{k+1}}, C_{j_k}^A + p_{j_{k+1}}\} - \\
&\quad - \max\{r_{j_{k+1}}^B + p_{j_{k+1}}, C_{j_k}^B + p_{j_{k+1}}\}| \leq \\
&\leq \max\{|r_{j_{k+1}}^A - r_{j_{k+1}}^B|, |C_{j_k}^A - C_{j_k}^B|\} \leq \max_{j \in N} |p_j^A - p_j^B|,
\end{aligned}$$

тогда из (3.10) получаем

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |C_j^A - C_j^B| \leq \sum_{k \in N} \max_{j \in N} |r_j^A - r_j^B| = n \max_{j \in N} |r_j^A - r_j^B|.$$

□

**Лемма 3.16.** Функция

$$(3.11) \quad \rho(A, B) = n \cdot \max_{j \in N} |r_j^A - r_j^B| + n \cdot \sum_{j=1}^n |p_j^A - p_j^B| + \sum_{j=1}^n |d_j^A - d_j^B|$$

удовлетворяет свойствам метрики.

**Доказательство.** Функция (3.11) очевидно является симметричной, неотрицательной и равняется нулю тогда и только тогда, когда  $A = B$ . В силу известных свойств модуля, неравенство треугольника выполняется для каждого из слагаемых функции (3.11), а значит и для самой функции. □

**Лемма 3.17.** Функция (3.11) при любом расписании  $\pi$  удовлетворяет неравенству

$$(3.12) \quad \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \rho(A, B).$$

**Доказательство.** Пусть пример  $C$  имеет директивные сроки как у примера  $B$  и одинаковые с примером  $A$  времена поступления и продолжительности обслуживания требований. Пусть далее, пример  $D$  имеет одинаковые с примером  $B$  директивные сроки и времена поступления тре-

бований и одинаковые с примером  $A$  продолжительности обслуживания.

Используя леммы 3.13–3.15, получим

$$\begin{aligned}
\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^C \right| + \left| \sum_{j \in N} T_j^C - \sum_{j \in N} T_j^D \right| + \\
&+ \left| \sum_{j \in N} T_j^D - \sum_{j \in N} T_j^B \right| \leq \sum_{j=1}^n |d_j^A - d_j^B| + n \cdot \max_{j \in N} |r_j^A - r_j^B| + \\
&+ n \cdot \sum_{j=1}^n |p_j^A - p_j^B| \leq \rho(A, B).
\end{aligned}$$

□

Согласно леммам 3.12 и 3.17 для функции (3.11) выполняется

$$(3.13) \quad \sum_{j \in N} T^A(\pi^B) - \sum_{j \in N} (\pi^A) \leq 2\rho(A, B),$$

где  $\pi^A$  и  $\pi^B$  — оптимальные расписания примеров  $A$  и  $B$  соответственно.

Полученная метрика для задачи суммарного запаздывания является сепарабельной, что значительно упрощает задачу ее минимизации — можно минимизировать зависящие от директивных сроков, моментов поступления и продолжительностей обслуживания части метрики отдельно.

### 3.5.2 Метрики для общего случая целевой функции

Рассмотрим общий случай аддитивной целевой функции:

$$(3.14) \quad F(\pi) = \sum_{j \in N} \phi_j(\pi, r_1, \dots, r_n, p_1, \dots, p_n, d_j).$$

В этом случае метрика может быть построена в виде:

$$(3.15) \quad \rho(A, B) = \sum_{j \in N} \sum_{i \in N} (R_{ji} |r_j^A - r_j^B| + P_{ji} |p_j^A - p_j^B|) + \sum_{j \in N} D_j |d_j^A - d_j^B|,$$

Здесь где  $R_{ji}, P_{ji}, D_j$  – константы Липшица для функций  $\phi_i$  по переменным  $r_j, p_j$  и  $d_j, j, i \in N$ .

Аналогичная формула имеет место для случая минимаксной целевой функции:

$$(3.16) \quad F(\pi) = \max_{j \in N} \phi_j(\pi, r_1, \dots, r_n, p_1, \dots, p_n, d_j).$$

$$(3.17) \quad \rho(A, B) = \sum_{j \in N} (R_j |r_j^A - r_j^B| + P_j |p_j^A - p_j^B|) + D \max_{j \in N} |d_j^A - d_j^B|,$$

Обратим внимание, что полученные метрики также являются сепарабельными.

### 3.5.3 Сведение к задаче линейного программирования

Рассмотрим случай, когда некоторый полиномиально разрешимый класс примеров определяется системой линейных неравенств вида

$$\mathcal{A} \cdot R^B + \mathcal{B} \cdot P^B + \mathcal{C} \cdot D^B \leq H,$$

где  $R^B = (r_1^B, \dots, r_n^B)^T, P^B = (p_1^B, \dots, p_n^B)^T, D^B = (d_1^B, \dots, d_n^B)^T, p_j^B \geq 0, r_j^B \geq 0, j \in N$ , верхний индекс<sup>T</sup> обозначает транспонирование,  $\mathcal{A}, \mathcal{B}, \mathcal{C}$  – матрицы размера  $m \times n$ , и  $H$  –  $m$ -мерный вектор.

Тогда для отыскания в полиномиально разрешимом классе примера  $B$ , ближайшего к примеру  $A$ , достаточно решить следующую задачу линейного программирования.

$$(3.18) \quad \min f = n \cdot (y^r - x^r) + n \cdot \sum_{j=1}^n (y_j^p - x_j^p) + \sum_{j=1}^n (y_j^d - x_j^d),$$

при условиях

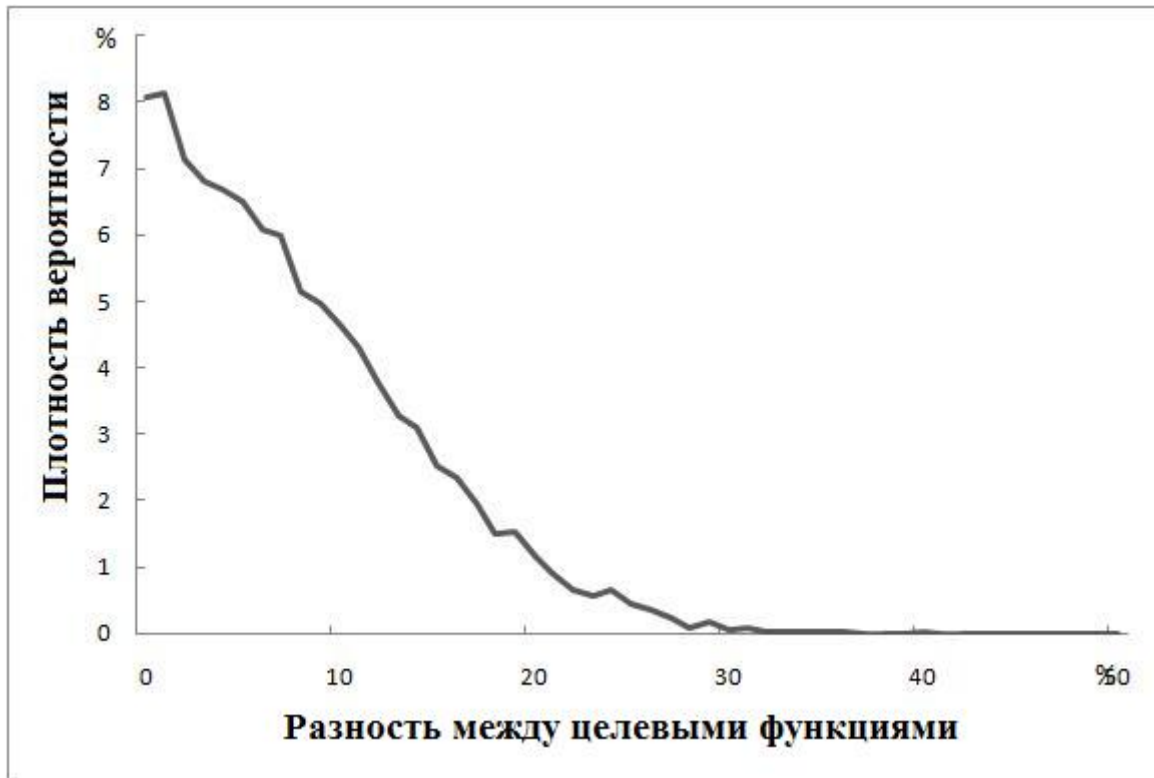
$$\begin{aligned} x^r &\leq r_j^A - r_j^B \leq y^r, \\ x_j^p &\leq p_j^A - p_j^B \leq y_j^p, \\ x_j^d &\leq d_j^A - d_j^B \leq y_j^d, \\ r_j^B &\geq 0, p_j^B \geq 0, j \in N, \\ \mathcal{A} \cdot R^B + \mathcal{B} \cdot P^B + \mathcal{C} \cdot D^B &\leq H. \end{aligned}$$

В полученной задаче линейного программирования присутствуют  $7n + 2$  переменных:  $r_j^B, p_j^B, d_j^B, x_j^p, y_j^p, x_j^d, y_j^d, x^r, y^r, j = 1, \dots, n$ . Для её решения существуют полиномиальные алгоритмы, поэтому вся схема нахождения приближенного решения в рассматриваемом случае является полиномиальной.

### 3.5.4 Численные эксперименты

Для определения эффективности предложенной схемы была проведена серия численных экспериментов. Поиск полиномиально разрешимых примеров проводился в следующих классах:

- $\{\mathcal{PR} : p_j = p, r_j = r, j \in N\}, \rho(A, B) = n \cdot \sum_{j=1}^n |p_j^A - p| + n \cdot \max_{j \in N} |r_j^A - r|,$
- $\{\mathcal{PD} : p_j = p, d_j = d, j \in N\}, \rho(A, B) = n \cdot \sum_{j \in N} |p_j^A - p| + \sum_{j \in N} |d_j^A - d|,$
- $\{\mathcal{RD} : r_j = r, d_j = d, j \in N\}, \rho(A, B) = n \cdot \max_{j \in N} |r_j^A - r| + \sum_{j \in N} |d_j^A - d|,$
- $\{\mathcal{P} : p_j = p, j \in N\}, \rho(A, B) = n \cdot \sum_{j \in N} |p_j^A - p|,$
- $\{\mathcal{R0} : r_j = 0, j \in N\}, \rho(A, B) = n \cdot \max_{j \in N} |r_j^A - r|.$



**Рис. 3.8.** Плотность распределения модуля разности суммарных запаздываний примеров, в процентах от значения метрики.

Здесь, для каждого класса указана функция метрики  $\rho(A, B)$ ,  $A$  – произвольный пример,  $B$  – пример принадлежащий соответствующему классу. В первых трех классах решением является расписание, упорядоченное по неубыванию свободного параметра. Алгоритмы решения задач последних двух классов представлены в [14] и [7] и имеют сложности  $O(n^7)$  и  $O(n^4 \sum p_j)$ , соответственно.

Для нахождения в указанных классах полиномиально разрешимого примера  $B$ , ближайшего к заданному примеру, необходимо найти минимум функций.

$$(3.19) \quad f(r) = n \cdot \max_{j \in N} |r_j^A - r|.$$



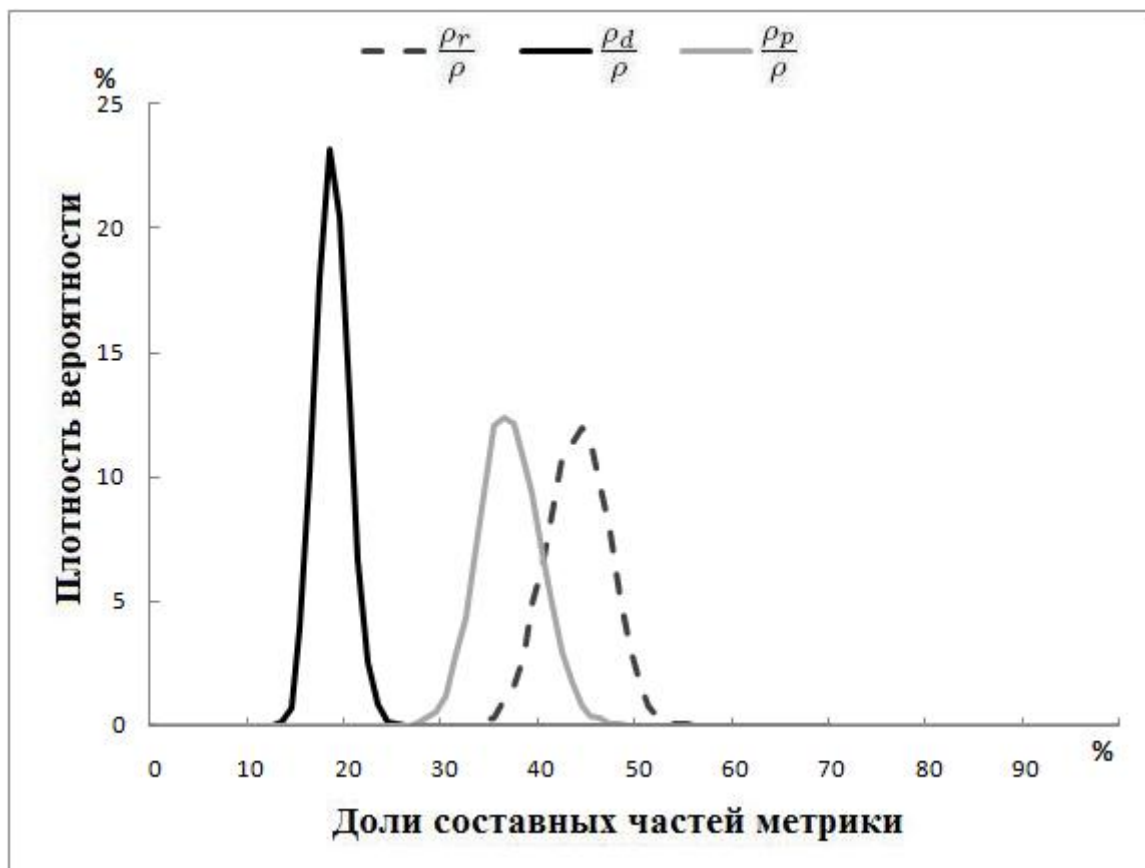


Рис. 3.9. Плотность распределения процентных долей частей метрики, зависящих от отдельных параметров.

$$(3.20) \quad g(p) = n \cdot \sum_{j=1}^n |p_j^A - p|.$$

$$(3.21) \quad h(d) = \sum_{j \in N} |d_j^A - d|.$$

**Лемма 3.18.** • Минимум функции (3.19) достигается в точке  $r = \frac{r_{max}^A + r_{min}^A}{2}$ , где  $r_{max}^A = \max_{j \in N} r_j^A$ ,  $r_{min}^A = \min_{j \in N} r_j^A$ .

- Минимум функции (3.20) достигается в точке  $p \in \{p_1^A, \dots, p_n^A\}$ .
- Минимум функции (3.21) достигается в точке  $d \in \{d_1^A, \dots, d_n^A\}$ .



Рис. 3.10. Плотность распределения реальной погрешности приближенного решения, в процентах от теоретической.

**Доказательство.** Функция  $f(r)$  представима в виде

$$n \cdot \max_{j \in N} |r_j^A - r| = \max\{r - r_{min}^A, r_{max}^A - r\} = \frac{r_{max}^A - r_{min}^A}{2} + \left| r - \frac{r_{max}^A + r_{min}^A}{2} \right|$$

и очевидно имеет минимум в точке  $\frac{r_{max} + r_{min}}{2}$ .

Пусть функция  $g(p)$  имеет минимум в точке  $p_0$ , тогда либо  $f'(p_0) = 0$ , либо  $p_0 \in \{p_1^A, \dots, p_n^A\}$ . Поскольку  $g(p)$  — кусочно-линейная функция, обращение её производной в нуль означает, что функция является константой на некотором интервале  $[p_k^A, p_{k+1}^A]$ ,  $k = 1, \dots, n - 1$  а значит граничные точки  $p_k^A$  и  $p_{k+1}^A$  также являются точками минимума.

Последнее утверждение леммы о минимуме функции  $h(d)$  доказывается аналогично.

□

Были проведены несколько серий экспериментов. Во всех сериях использовались примеры с параметрами, распределенными равномерно на

интервалах  $[1, 100]$  для  $p_j^A$ ,  $[p_j, \sum_{j \in N} p_j]$  для  $d_j^A$  и  $[0, d_j - p_j]$  для  $r_j^A$ .

В первой серии экспериментов проверялись результаты леммы 3.17, влияющей на величину погрешности. Для каждого  $n = 10, 20, \dots, 100$  генерировалось 10000 пар примеров. Для каждой пары вычислялась величина  $\frac{|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B|}{\rho(A, B)}$ . Также для определения параметров, имеющих наибольшее влияние на функцию метрики вычислялись процентные величины вкладов частей метрики, зависящих от продолжительностей обслуживания, директивных сроков и моментов поступления.

Результаты представлены в таблице 3.4. Среднее значение  $\frac{|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B|}{\rho(A, B)}$  меняется от 5% до 10% при росте  $n$ , а части функции метрики, зависящие от продолжительностей обслуживания, директивных сроков и моментов поступления дают вклады 35%, 20% и 25% в общую величину функции, соответственно.

Графики полученных плотностей распределения для  $n = 60$  показаны на рис. 3.8 и рис. 3.9.

Вторая серия экспериментов служила проверке метода изменения параметров непосредственно. Эксперименты проводились по следующей схеме. Рассматривались значения  $n = 4, 5, \dots, 10$ , для каждого  $n$  генерировались по 10000 примеров. К каждому примеру применялась вышеописанная схема для нахождения приближенного решения со значением целевой функции  $F_e$ , затем при помощи алгоритма ветвей-и-границ искалось точное решение с оптимальным значением целевой функции  $F^*$ . Далее вычислялось  $\Delta$  — отношение реальной погрешности схемы  $\delta = F_e - F^*$  к её верхней оценке, определяемой неравенством (3.13):

$$(3.22) \quad \Delta = \frac{F_e - F^*}{2\rho(A, B)}.$$

По итогам экспериментов строились зависимости плотности распределения ошибки  $\Delta$  для различных  $n$ . Типичный вид полученной зависимости показан на рис. 3.10.

$n$	$\frac{ \sum T_j^A - T_j^B }{\rho}$	$\frac{\rho_r}{\rho}$	$\frac{\rho_p}{\rho}$	$\frac{\rho_d}{\rho}$
10	11,7%	35,6%	42,3%	20,6%
20	10,4%	39,7%	39,4%	19,4%
40	8,9%	42,4%	37,4%	18,6%
60	7,8%	43,6%	36,6%	18,3%
80	7,3%	44,4%	34,4%	18%
100	6,7%	44,9%	35,7%	17,9%

**Таблица 3.4. Средняя разница между целевыми функциями и доли составных частей метрики**

$n$	$\mathcal{PR}$	$\mathcal{PD}$	$\mathcal{RD}$	$\mathcal{P}$	$\mathcal{R0}$
4	2,5%	4,6%	20,8%	1,8%	2,9%
5	2,6%	4,8%	23,1%	1,9%	2,8%
6	2,6%	4,6%	24,6%	1,9%	2,7%
7	2,6%	4,7%	26%	1,9%	2,5%
8	2,5%	4,6%	27%	2%	2,3%
9	2,4%	4,7%	27,9%	2%	2,2%
10	2,4%	4,6%	28,6%	1,9%	2,1%

**Таблица 3.5. Средняя экспериментальная погрешность в процентах от теоретической**

Было обнаружено, что если поиск полиномиально разрешимого примера ведется в классе  $RD$ , то средняя погрешность решения растет от 20% до 30% от верхней оценки (3.13) при увеличении  $n$ , это показывает, что расписание по возрастанию продолжительностей обслуживания плохо применимо для примеров с выбранным распределением параметров. Для остальных классов средняя погрешность решения не зависит от  $n$ , и составляет несколько процентов от максимальной теоретической. Столь малая погрешность обусловлена тем, что примерно в 20% случаев метод изменения параметров давал точное решение задачи. Зависимость средней ошибки  $\Delta$  от  $n$  представлена в таблице 3.5.

## Список общих обозначений

$n_{ij}$  — количество вагонов, которые необходимо доставить со станции  $i$  на станцию  $j$ ;

$J_{ijk}$  —  $k$ -й заказ (вагон) для отправки со станции  $i$  на станцию  $j$ ;

$m_{ijk}$  — масса вагона;

$l_{ijk}$  — длина вагона;

$h_{ij}$  — длина пути между  $i$  и  $j$  станциями;

$d_{ijk}$  — директивный срок доставки  $k$ -го заказа со станции  $i$  на станцию назначения  $j$ ;

$r_{ijk}$  — время поступления  $k$ -го заказа на станцию  $i$  для отправки на станцию  $j$ ;

$p_{ij}$  — время передвижения состава между станциями  $i$  и  $j$ ;

$C_i^j$  — момент доставки  $i$ -го заказа на станцию  $j$ ;

$L$  — множество локомотивов;

$B$  — множество локомотивных бригад;

$w_i$  — коэффициент значимости;

$z_i$  — штраф за задержку вагона.

$Q_i^{in}$  — множество поездов, прибывающих на станцию  $i$ ;

$Q_i^{out}$  — множество поездов, отбывающих со станции  $i$ ;

$t_{qi}^{in}$  — время прибытия на станцию  $i$  поезда  $q$ ;

$t_{qi}^{out}$  — время отбытия со станции  $i$  поезда  $q$ ;

$v_{it}^{out}$  — вершина отбытия со станции  $i$  поезда  $q$  в момент  $t_{qi}^{out}$ ;

$v_{it}^{in}$  — вершина прибытия на станцию  $i$  поезда  $q$  в момент  $t_{qi}^{in}$ ;

$v_{it}^{srt}$  — вершина сортировки на станцию  $i$  поезда  $q$ ;

$a_{qij}^{mov}$  — ребро движения поезда  $(v_{it'}^{out}, v_{jt''}^{in})$ , соответствующее движению поезда  $q \in Q$  со станции  $i$  на следующую на его маршруте станцию  $j$ ;

$a_{qi}^{tr}$  — ребро стоянки поезда  $(v_{it'}^{in}, v_{it''}^{out})$ , соответствующее стоянке поезда  $q \in Q$  на промежуточной на его маршруте станции  $i$ ;

$a_{qi}^{dwn}$  — ребро отцепки вагонов  $(v_{it'}^{in}, v_{i,t+\Delta}^{srt})$ , соответствующее отцепке вагонов от поезда  $q \in Q$  на находящейся на его маршруте станции  $i$ ;

$a_{qi}^{up}$  — ребро прицепки вагонов  $(v_{i,t-\Delta}^{srt}, v_{it}^{out})$ , соответствующее прицепке вагонов к поезду  $q \in Q$  на находящейся на его маршруте станции  $i$ ;

$a_{it't''}^{srt}$  — ребро сортировки  $(v_{it'}^{srt}, v_{it''}^{srt})$  между соседними вершинами сортировки станции  $i$ ;

$f_{ij}^a$  — количество вагонов, “перемещенных” по ребру  $a$  из  $i$  в  $j$ .

$\delta_v^-, \delta_v^+$  — множества входящих и исходящих рёбер для вершины  $v$ .

# Литература

- [1] *Бодюл В.И.* Математическая модель распределения вагонного парка по железным дорогам в условиях неравномерности грузовых перевозок // Вестник ВНИИЖТ. — 2006. — №3.
- [2] *Данциг Дж.* Линейное программирование. Его применение и обобщение — М.: Изд-во “Прогресс”, 1966.
- [3] *Козлов П.А., Владимирская И.П.* Пути повышения обоснованности технологических решений // Вестник ВНИИЖТ. — 2009. — № 3. С. 8–12.
- [4] *Козлов П.А., Владимирская И.П.* Построение систем автоматизированного управления потоками вагонов разных собственников // Вестник ВНИИЖТ. — 2009. — № 6. С. 8–11
- [5] *Козлов П.А., Владимирская И.П., Тушин Н.А.* Оптимальное управление работой вагонов разных собственников // Вестник ВНИИЖТ. — 2010. — № 4.
- [6] *Лазарев А.А.* Теория расписаний. Оценки абсолютной погрешности и схема приближенного решения задач теории расписания. — М.: МФТИ, 2008.
- [7] *Лазарев А.А., Гафаров Е.Р.* Теория расписаний. Минимизация суммарного запаздывания для одного прибора — М.: Вычислительный центр им. А.А. Дородницына РАН, 2006.
- [8] *Левин Д.Ю.* Теория оперативного управления перевозочным процессом. — М.: “Транспортная книга”, 2008.

- [9] *Литвак Б.Л.* Алгоритм решения динамической транспортной задачи. // В кн.: Системы многосвязного управления. — М.: Наука, 1977. — С.59–69.
- [10] *Макаров В.Л.* Динамическая транспортная задача. // Труды Ленингр. инж.-экон. ин-та. — 1965. — вып. 58.
- [11] *Сурин С.С.* Динамическая транспортная задача и некоторые её обобщения. // В сб.: Методы вычислений. Вып. 3. — Л.: Изд-во ЛГУ, 1966.
- [12] *Acbarya, D., Martland C.D., J. M. Sussman* Using Expert Systems and Optimization Techniques for Rail Relay Scheduling. // Proceedings of the ASCE First International Conference on Advanced Technologies in Transportation Engineering, San Diego CA, Feb., 1989. — P. 378-383.
- [13] *Allahverdi A., Ng C.T., Cheng T.C.E., Kovalyov M.Y.* A survey of scheduling problems with setup times or costs. // European Journal of Operational Research. — 2008. — No. 187. P. 985–1032.
- [14] *Baptiste Ph.* Batching identical jobs. // Math Meth Oper Res. — 2000. — No. 52. P. 355–367.
- [15] *Baptiste Ph., Brucker P., Knust S. and Timkovsky V.G.* Ten notes on equal-processing-time scheduling. // 4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies. — 2004. — V. 2. P. 111–127.
- [16] *Beaujon G.J., Turnquist M.A.* A Model for Fleet Sizing and Vehicle Allocation. // Transportation Science. — 1991. — No. 25.1. P. 19–45.
- [17] *Bellman R.* Mathematical aspects of scheduling theory // Journal of the Society of Industrial and Applied Mathematics. — 1956. — Vol. 4. P. 168–205.
- [18] *Billionet A.* Using integer programming to solve the train-platforming problem. // Transportation Science. — 2003. — No. 37(2). P. 213–222.



- [19] *Borndorfer R., Grotschel M., Lukac S., Mitusch M., Schlechte T., Schultz S., Tanner A.* An auctioneng approach to railway slot allocation. Technical Report 05-45. Konrad-Zuse-Zentrum fur Informationstechnik Berlin, 2005.
- [20] *Borndorfer R., Schlechte T.* Solving railway track allocation problems. Technical Report 07-20. Konrad-Zuse-Zentrum fur Informationstechnik Berlin, 2007b.
- [21] *Brannlund U., Lindberg P.O., Nou A., Nilsson J.-E.* Railway Timetabling using Lagrangian Relaxation. // *Transportation Science.* — 1998. — No. 32.4. P. 358–369.
- [22] *Brucker P.* Scheduling Algorithms. — Springer-Verlag. 2001. — P. 365.
- [23] *Cacchiani V., Caprara A., Toth P.* A column generation approach to train timetabling on a corridor. // *4OR.* — 2008. — No. 6. P. 125–142.
- [24] *Cai X., Goh C. J.* A fast heuristic for the train scheduling problem. // *Computers and Operations Research.* — 1994. — No. 21(5). P. 499–510.
- [25] *Cai X., Goh C. J., Mees A. I.* Greedy heuristics for rapid scheduling of trains on a single track. // *IIE Transactions.* — 1998. — No. 30. P. 481–493.
- [26] *Caprara A., Fischetti M., Toth P.* Modelling and solving the train timetabling problem. // *Operation Research.* — 2002. — No. 50. P. 851–861.
- [27] *Caprara A., Galli L., Toth P.* 04. solution of the train platforming problem.// *ATMOS 2007 — 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization and Systems, Dagstuhl, Germany, 2007.*
- [28] *De Luca Cardillo D., Mione N.* k l-list  $\tau$  colouring of graphs. // *European Journal of Operational Research.* — 1998. — No 106. P. 160–164.

- [29] *Carey M., Lockwood D.* A model, algorithms and strategy for train pathing. // The Journal of Operational Research Society. — 1995. — No 46(8) P. 988–1005.
- [30] *Cornelsen S., Di Stefano G.* Track assignment. // Journal of Discrete Algorithms. — 2007. — No. 5(2). P. 250–261.
- [31] *Delorme X.* Modelisation et resolution de problemes lies a l’exploitation d’infrastructures ferroviaires. PhD thesis, Universite de Valenciennes et du Hainaut Cambresis, 2003.
- [32] *Ferreira L., Murray M.H.* Modelling Rail Track Deterioration and Maintenance: Current Practices and Future Needs. // Transport Reviews. — 1997. — No. 17.3. P. 207–221.
- [33] *Gandibleux X., Jorge J., Angibaud S., Delorme X., Rodriguez J.* An ant colony optimization inspired algorithm for the set packing problem with application to railway infrastructure. // In Proceedings of the Sixth Metaheuristics International Conference (MIC2005). — 2005. P. 390–396.
- [34] *Gorman M.F.* Operating Plan Model Improves Service Design at Santa Fe Railway. // Interfaces. — 1998. — No 28.4 P. 1–12.
- [35] *Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.* Optimization and approximation in deterministic machine scheduling: a survey. // Ann. Discrete Math. — 1979. — No. 5. P. 287–326.
- [36] *Higgins A., Ferreira L., Kozan E.* Modeling Single-Line Train Operations. // Transportation Research Record. — 1995. — No. 1489 P. 9–16.
- [37] *Higgins A., Kozan E., Ferreira L.* Optimal scheduling of trains on a single line track. // Transportation Research Part B. — 1996. — No. 30B(2). P. 147–161.

- [38] *Higgins A., Ferreira L., Kozan E.* Modeling Single-Line Train Operations. // Transportation Research Record. — 1995. — No. 1489. P. 9–16.
- [39] *Jha K.C., Ahuja R.K. Sahin G.* New Approaches for Solving the Block-to-Train Assignment Problem. // Networks. — 2008. — No. 51(1). P. 48–62.
- [40] *Jungnickel D.* Graphs, Networks and Algorithms. — Springer, 2007.
- [41] *Jovanovic D., Barker P.T.* Tactical Scheduling of Rail Operations: The SCAN-I System. // Transportation Science. — 1991. — No. 25.1. P. 46–64.
- [42] *Kantorovich L.* On the translocation of masses // C. R. (Doklady) Acad. Sci. URSS (N. S.). — 1942. — No. 37. P. 199–201.
- [43] *Kraay D., Barker P.T., Chen B.T.* Optimal Pacing of Trains in Freight Railroads: Model Formulation and Solution. // Operations Research. — 1991. — No. 39.1. P. 82–99.
- [44] *Kravchenko S., Werner F.* Parallel Machine Problems with Equal Processing Times: A Survey. // Journal of Scheduling. — 2011. — V. 14. No. 5. P. 435–444.
- [45] *Kwon O.K., Martland C.D., Sussman J.M.* Routing and Scheduling Temporal and Heterogeneous Freight Car Traffic on Rail Networks. // Transportation Research. — 1998. — No. 34E.2. P. 101–115.
- [46] *LeBlanc L.J.* Global Solutions for a Nonconvex Nonconcave Rail Network Model. // Management Science. — 1976. — No. 23.2. P. 131–139.
- [47] *Liebchen C., Mohring R.* The modelling power of the periodic event scheduling problem: Railway timetables - and beyond. // Technical Report 2004/20, Technische Universitat Berlin, Institut fur Mathematik. — 2004.

- [48] *Liu S.-Q., Kozan E.* Scheduling trains as a blocking parallel-machine job shop scheduling problem. // Computers and Operations Research. — 36(10) P. 2840–2852.
- [49] *Marin A., Salmeron J.* Tactical Design of Rail Freight Networks - Part I: Exact and Heuristic Methods. // European Journal of Operational Research. — 1996. — No. 90, P. 26–44.
- [50] *Morlok E.K., Peterson R.B.* Final Report on a Development of a Geographic Transportation Network Generation and Evaluation Model. // Journal of the Transportation Research Forum. — 1970. — No. 11. P. 71–105.
- [51] *Monge G.* Mémoire sur la théorie des déblais et de remblais. Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année. — 1781. — P. 666–704.
- [52] *Newton H. N., Barnhart C., Vance P. H.* Constructing Railroad Blocking Plans to Minimize Handling Costs. // Transportation Science. — 1998. — No. 32.4. P. 330–345.
- [53] *Nozick L.K., Morlok E.K.* A Model for Medium-Term Operations Planning in an Intermodal Rail-Truck Service. // Transportation Research. — 1997. — No. 31A.2. P. 91–107.
- [54] *Oliveira O., Smith B. M.* A job shop scheduling model for the single track-railway timetabling problem. // Technical Report 2000.21, University of Leeds. — 2000.
- [55] *de Oliveira E.S.* Solving Single Track Railway Scheduling Problem Using Constraint Programming. // The University of Leeds, School of Computing, PhD Thesis. — 2001. P. 129.
- [56] *Peeters L.* Cyclic Railway Timetable Optimization. // PhD thesis, Erasmus Research Institute of Management. — 2003.

- [57] *Powell W.B., Carvalho T.A.* Dynamic Control of Logistics Queuing Networks for Large-Scale Fleet Management. // Transportation Science. — 1998a. — No. 32.2. P. 90–109.
- [58] *Rodriguez J.* A constraint programming for real-time trains scheduling at junctions. // Transportation Research Part B. — 2007. — No. 41(2) P. 231-245.
- [59] *Sahin I.* Railway traffic control and train scheduling based on inter-train conflict management. // Transportation Research Part B. — 1999. — No. 33 P. 511–534.
- [60] *Sauder R.L., Westerman W.M.* Computer Aided Train Dispatching: Decision Support Through Optimization. // Interfaces. — 1983. — No. 13.6 P. 24–37.
- [61] *Serafini P., Ukovich W.* A mathematical model for periodic scheduling problems. // Society for Industrial and Applied Mathematics Journal on Discrete Mathematics. — 1989. — No. 2(4). P. 550–581.
- [62] *Sherali H.D., Suharko A.B.* A Tactical Decision Support System for Empty Railcar Management. // Transportation Science. — 1998. — No. 32.2. P. 306–329.
- [63] *Sherali H.D., Tuncbilek C.H.* Static and Dynamic Time-Space Strategic Models and Algorithms for Multilevel Rail-Car Fleet Management. // Management Science. — 1997. — No. 43.2. P. 235–250.
- [64] *Szpigel B.* Optimal train scheduling on a single line railway. // Operations Research. — 1973. — No. 72. P. 344–351.
- [65] *Turnquist M.A., Markowicz B.P.* An Interactive Microcomputer-Based Model for Railroad Car Distribution. // Working Paper, Cornell University, Ithaca, NY. — 1989.
- [66] *Wolsey L.A., Nemhauser G.L.* Integer and Combinatorial Optimization. — N.Y.: John Wiley & Sons Inc., 1988.

- [67] *Ziarati K., Soumis F., Desrosiers J., Gelnis S., Saintonge A.* Locomotive Assignment with Heterogeneous Consists at CN North America. // European Journal of Operations Research. — 1997. — No. 97.2. P. 281–292.
- [68] *Zwaneveld P. J., Kroon L. G., Romeijn H. E., Salomon M., Dautzere-Peres S., van Hoesel S. P. M., Ambergen H. W.* Routing trains through railway stations: Model formulation and algorithms. // Transportation Science. — 1996. — No. 30(3). P. 181–194.
- [69] *Zwaneveld P. J., Kroon L. G., van Hoesel S. P. M.* Routing trains through a railway station based on a node packing model. // European Journal of Operational Research. — 2001. — No. 128 P. 14–33.

ЛАЗАРЕВ АЛЕКСАНДР АЛЕКСЕЕВИЧ  
МУСАТОВА ЕЛЕНА ГЕННАДЬЕВНА  
КВАРАЦХЕЛИЯ АЛЕКСАНДР ГОНЕРОВИЧ  
ГАФАРОВ ЕВГЕНИЙ РАШИДОВИЧ

**ТЕОРИЯ РАСПИСАНИЙ.  
ЗАДАЧИ УПРАВЛЕНИЯ ТРАНСПОРТНЫМИ  
СИСТЕМАМИ**

Подписано в печать 30.06.2012

Объем 10 п.л. Тираж 25 экз.

Заказ №

Оформление обложки: М.А. Лазарев, М.О. Пороснова.

Физический факультет МГУ

имени М.В. Ломоносова

119991, Москва, ГСП-1, Ленинские горы, д.1, стр.2.

Отпечатано в отделе оперативной печати

физического факультета МГУ