

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ  
**Институт проблем управления  
им. В.А. Трапезникова РАН**

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

**ТЕОРИЯ РАСПИСАНИЙ.  
ЗАДАЧИ ЖЕЛЕЗНОДОРОЖНОГО  
ПЛАНИРОВАНИЯ**

Москва 2012

УДК 519.854.2

*Лазарев А.А., Мусатова Е.Г., Гафаров Е.Р., Кварацхелия А.Г.*  
**Теория расписаний. Задачи железнодорожного планирования /**  
Научное издание. — М.: ИПУ РАН, 2012. — 92 с.

Издание посвящено построению моделей и методам решения задач, возникающих при планировании грузовых железнодорожных перевозок.

В зависимости от ограничений на локомотивы, грузоподъёмность составов, предлагаются различные алгоритмы решения задач формирования составов и расписания движения грузовых поездов.

Изложенный материал будет полезен как специалистам в области управления, так и практикам, занимающимся решением задач планирования железнодорожных грузоперевозок.

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

Утверждено к печати Редакционным советом Института

Текст воспроизводится в виде, утвержденном  
Редакционным советом Института

ISBN 978-5-91450-102-7

## Содержание

<b>Введение</b>	<b>5</b>
<b>1 Обзор публикаций по железнодорожной тематике</b>	<b>6</b>
1.1 Стратегические задачи проектирования инфраструктуры железнодорожной сети . . . . .	6
1.2 Стратегические задачи управления парком локомотивов и вагонов . . . . .	7
1.3 Задачи планирования расписаний и маршрутизации грузоперевозок . . . . .	8
1.3.1 Формирование расписания движения поездов . . . . .	8
1.3.2 Задачи маршрутизации перевозок . . . . .	10
1.3.3 Комбинированные модели расписаний и маршрутизации . . . . .	12
1.4 Обзор публикаций по железнодорожной тематике в России за последние годы . . . . .	14
<b>2 Задача формирования и маршрутизации грузовых составов РЖД</b>	<b>19</b>
2.1 Задача формирования железнодорожных составов и маршрутов движения при фиксированных расписаниях . . . . .	20
2.2 Задача с гибким расписанием движения поездов при фиксированных маршрутах . . . . .	24
2.2.1 Первая версия постановки задачи . . . . .	24
2.2.2 Упрощенная модель . . . . .	28
<b>3 Задача оперативного управления движением составов</b>	<b>31</b>
<b>4 Задача формирования грузовых потоков</b>	<b>34</b>
4.1 Постановка задачи . . . . .	34
4.2 Использование решения задачи в деятельности РЖД . . . . .	36
<b>5 Задача формирования составов и расписания движения грузовых поездов</b>	<b>38</b>
5.1 Общая постановка задачи . . . . .	38
5.2 Задачи с двумя станциями. . . . .	39
5.3 Задачи с цепочкой станций . . . . .	40
5.4 Задачи с замкнутой цепочкой станций . . . . .	40
5.5 Задачи со станциями, расположенными в форме звезды . . . . .	41

5.6	Полиномиальная разрешимость задач о станциях . . . . .	41
<b>6</b>	<b>Решение задачи минимизации среднего времени выполнения заказов для двух станций с одним локомотивом</b>	<b>46</b>
<b>7</b>	<b>Задача составления расписания движения поездов между двумя станциями, соединенными однопутной железной дорогой</b>	<b>52</b>
7.1	Сведение STRSP2 к одноприборной задаче . . . . .	53
7.2	Одноприборная задача . . . . .	55
7.3	Алгоритмы для задач с упорядоченными подмножествами $N_1$ и $N_2$ . . . . .	58
7.4	Задачи с частично упорядоченными подмножествами . . . . .	62
<b>8</b>	<b>Задача минимизации максимального взвешенного временного смещения выполнения заказа для двух станций</b>	<b>65</b>
8.1	Задача минимизации общего времени выполнения заказов при ограничении на максимальное взвешенное временное смещение . . . . .	67
8.1.1	Алгоритм решения задачи . . . . .	67
8.1.2	Сложность алгоритма . . . . .	70
8.2	Решение задачи минимизации максимального взвешенного временного смещения . . . . .	70
<b>9</b>	<b>Методы решения задач железнодорожного планирования</b>	<b>74</b>
9.1	Метод изменения параметров . . . . .	74
9.2	Сведение к задаче линейного программирования . . . . .	79
9.3	Метод генерации колонок . . . . .	82
<b>10</b>	<b>Дальнейшие направления исследований</b>	<b>83</b>
<b>11</b>	<b>Список общих обозначений</b>	<b>84</b>
	<b>Список литературы</b>	<b>86</b>

## Введение

В работе проведен обзор современных подходов к решению задач, возникающих в рамках железнодорожной тематики (см. гл. 1). Далее на основе анализа особенностей структуры Российской железнодорожной системы, поставлены задачи, возникающие в области маршрутизации транспортных средств и транспортной логистики, а также разработаны методы решения некоторых из поставленных задач.

В работе получены следующие результаты.

1. Построены модели формирования грузовых составов и расписаний их движения с целью минимизации суммарного запаздывания доставки грузов в пункты назначения (см. гл. 2). Рассмотрен вариант с фиксированными расписаниями и маршрутами, когда необходимо распределить вагоны по составам. Во втором случае при фиксированных маршрутах расписание поездов заранее не задано. При моделировании использовалось понятие пространственно-временного направленного графа.
2. Построена модель оперативного управления движением составом на одном узле железной дороги. Задача заключается в максимизации пропускной способности рассматриваемого узла (см. гл. 3).
3. Построена модель формирования грузовых потоков через сортировочные станции. Данная задача сочетает в себе долгосрочное планирование грузовых потоков через сортировочные станции и формирование грузовых составов с учётом среднесуточных потоков между сортировочными станциями (см. гл. 4).
4. Сформулированы задачи формирования составов и графиков движения грузовых поездов (с двумя станциями, с цепочкой станций и другими базовыми структурами расположения сортировочных станций), для решения которых можно построить точные полиномиальные (псевдополиномиальные) алгоритмы (см. гл. 5). В частности, построены точные полиномиальные алгоритмы решения следующих задач:
  - а) задачи минимизации среднего времени выполнения заказов для двух станций с одним локомотивом (см. гл. 6);
  - б) задачи составления расписания движения между двумя станциями, соединенными однопутной железной дорогой и различ-

ными вариантами монотонных регулярных целевых функций (см. гл. 7);

с) задачи минимизации максимального взвешенного временного смещения выполнения заказа для двух станций, соединенных двухпутной железной дорогой (см. гл. 8).

5. Впервые для задач транспортной логистики удалось построить метрики, которые позволяют оценить абсолютную погрешность целевой функции суммарного запаздывания (см. гл. 9).

Разработанные модели и алгоритмы направлены на минимизацию издержек, увеличение объёма перевозок, уменьшение сроков доставки, повышение пропускной и провозной способности железных дорог.

## 1 Обзор публикаций по железнодорожной тематике

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.

### 1.1 Стратегические задачи проектирования инфраструктуры железнодорожной сети

С первым классом задач связаны следующие работы. В 1994 г. Хиггинс [33] исследовал проблему расположения обгонных путей на однопутных линиях для решения задачи минимизации запаздывания движения составов. В 1990 г. Краай [38] 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 [41] 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 [10] решал задачу оптимизации размещения старых рельсов на участках с менее интенсивным движением взамен их утилизации. В 1997 Феррейра и Мюррей [29] 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.

### 1.2 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 [59] 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 затрат при однопродуктовом потоке в сети. Учитывались пространственные и временные ограничения на предложение и спрос вагонов. В 1998 г. Шерали и Сухарко [56] 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 [60] 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 явля-

ется наличие различных типов локомотивов. В 1998 Пауэлл и Карвальо [51] 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 [14] 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 [57] 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.

### **1.3 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.

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

Одной из самых ранних работ была работа Шпигеля [58], 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 [18] рассматривал модель с однопутной 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 [54] 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 [37] 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 [38] 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 [26] 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 [34] 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 [21] 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 [22] 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 [53] предложил эвристический алгоритм, определяющий конфликты в порядке их появления и последовательно разрешающий их. В 2000 г. Оливейра и Смит [48] предложили постановку задачи для нескольких приборов применительно к железнодорожной тематике. Конфликты разрешались в хронологическом порядке, целевая функция минимизировала суммарное запаздывание.

### 1.3.2 Задачи маршрутизации перевозок

Рассмотрим работы по маршрутизации перевозок. Ньютон [46] 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 [40] 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 [47] 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 програм-

мирования 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. [61] 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 [62]. 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 [28] 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 [30] предложил как альтернативу метаэвристический алгоритм "муравьиных колоний".

В 2007 г. Капрара [24] 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 [25] (1998 г.), Биллионе [15] (2003 г.) и Корнельсен и Ди Стефано [27] (2007 г.) предложили метод раскраски графа для задачи подготовки платформ. В 2007 г. Родригез [52] 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.

### 1.3.3 Комбинированные модели расписаний и маршрутизации

В 1970 г. Морлок и Петерсон [45] 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 ограничения 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 [31] 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 [35], 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 [44] предложили агрегированную стационарную модель распределения грузов, в которой определяются маршруты поездов и количество ваго-

нов. 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 [23] рассмотрел задачу составления расписаний для поездов через главные коридоры для Итальянских железных дорог. Коридор 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. [20] описали метод генерации колонок для составления расписаний поездов в коридорах. В 2005 г. Борндорфер [16] 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 [17] 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 [55] 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 [50] в 2003 г. Либхен и Моринг [42] в 2004 г. рассмотрели более сложные модели с использованием периодической задачи.

В 2009 Лиу и Козан [43] 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 ре-

шения была взята идея смещения узкого места (shifting bottleneck), к которой добавили возможность установки пауз в расписание для устранения ситуаций блокировки, когда на одном однопутном участке одновременно находятся два состава, движущихся в противоположных 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.

#### **1.4 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 грузевых 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 [2] 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 начальных 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 [3] 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 усложнило 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.

Потоковые модели строятся на основе динамической транспортной задачи с задержками (ДТЗЗ), которая позволяет получить в отличие от стандартной статической оптимальный динамический процесс.

Представим модель управления потоками с вагонами разных собственников, описанную в статье [3].

По каждой струе потока могут быть заданы разные скорости и различные стоимости пропуска, а также индивидуальное ограничение по объёму при общей пропускной способности полигона. Оптимизировать управление многоструйными потоками можно с помощью многопродуктовой ДТЗЗ. Под продуктом понимается выделенная струя потока.

Предположим, что в транспортной системе производится, перемещается и потребляется (поглощается)  $K$  видов продукции  $\Pi$ . Обозначим через  $u_{ij}^k(t)$ ,  $t = \overline{0, T-1}$ ,  $k = \overline{1, K}$ , объём поставок  $K$  видов продукции с транспортными расходами  $c_{ij}^k(t)$  на единицу объёма, поступающей на  $p_j$  в момент  $T_0 + t$  и соответственно выходящей из  $p_i$  в момент  $t - t_0$ . Совместное перемещение транспортных потоков ограничено общей пропускной способностью  $v_{ij}(t)$  пути  $(p_i, p_j)$ ,  $i \neq j$ , где  $t_0$ ,  $T_0$  — некоторые заданные моменты времени.

Для пункта производства  $p_i$  заданы объёмы производства  $q_i^k(t)$ , запас  $u_{ij}^k(t)$ , а также функция штрафа (расходы)  $c_i^k(t)$  за хранение единицы запаса  $k$ -го вида продукции на складе ёмкостью  $v_i(t)$  с момента  $t - 1$  до момента  $t$ . Для пункта потребления  $p_j$  заданы объёмы потребления  $q_j^k(t)$ , запас  $u_{ij}^k(t)$ ,  $t = \overline{0, T-1}$ ,  $k = \overline{1, K}$ , функция штрафа  $c_j^k(t)$  за хранение единицы запаса на складе ёмкостью  $v_j(t)$  с момента  $t - 1$  до момента  $t$ .

Динамическая модель многопродуктовой транспортной задачи с задержками для  $m$  числа пунктов производства и  $n$  числа пунктов потребления может быть записана в следующем виде:

минимизировать функционал

$$J = \sum_{k=1}^K \sum_{j=1}^n \sum_{i=1}^m \sum_{t=0}^T c_{ij}^k(t) \cdot u_{ij}^k(t) + \\ + \sum_{k=1}^K \sum_{i=1}^m \sum_{\tau_i}^{T_i - t_i + T - 1} c_i^k(T_0 - T_i + \tau_i) \cdot u_{ii}^k(T_0 - T_i + \tau_i) + \\ + \sum_{k=1}^K \sum_{j=1}^n \sum_{t=1}^T c_j^k(t) \cdot u_{jj}^k(t),$$

где  $T_i = \max(t_{i1}, \dots, t_{in})$ ,  $t_i = \min(t_{i1}, \dots, t_{in})$ , а  $\tau_i$ , — текущее время поставщика, меняющееся в интервале  $[0, T_i - t_i + T - 1]$ , при ограничениях, задаваемых:

а) уравнением динамики изменения запасов  $k$ -го вида продукции для поставщика  $p_i$

$$u_{ii}^k(T_0 - T_i + \tau_i + 1) = u_{ii}^k(T_0 - T_i + \tau_i) + q_i^k(T_0 - T_i + \tau_i) - \\ - \sum_{j=1}^n u_{ij}^k(T_0 - T_i + \tau_i + t_{ij});$$

б) уравнением динамики изменения запасов  $k$ -го вида продукции для потребителя  $p_j$

$$u_{jj}^k(t + 1) = u_{jj}^k(t) - q_j^k(t) + \sum_{i=1}^m u_{ij}^k(t), t = \overline{0, T-1};$$

в) начальными и конечными запасами

$$u_{ii}^k(T_0 - T_i) = u_{ii}^k(T - t_i) = u_{jj}^k(0) = u_{jj}^k(T) = 0, k = \overline{1, K};$$

г) пропускными способностями

$$\sum_{k=1}^K u_{ij}^k(t) \leq v_{ij}(t), i \neq j;$$

$$\sum_{k=1}^K u_{ii}^k(T_0 - T_i + \tau_i) \leq v_i(T_0 - T_i + \tau_i);$$

$$\sum_{k=1}^K u_{jj}^k(t) \leq v_j(t);$$

д) условиями неотрицательности переменных запасов и поставок

$$u_{ii}^k(T_0 - T_i + \tau_i) \geq 0, \tau_i = \overline{0, T_i - t_i + T - 1};$$

$$u_{jj}^k(t) \geq 0, t = \overline{0, T - 1};$$

$$u_{ij}^k(t) u_{ij}^k(T_0 - T_i + \tau_i + t_{ij}) \geq 0, t = \tau_i + t_{ij} - T_i,$$

$$\tau_i = \overline{0, T_i - t_i + T - 1},$$

$$t = \overline{0, T - 1}, T_i = \max(t_{i1}, \dots, t_{in}),$$

$$t_i = \min(t_{i1}, \dots, t_{in}), k = \overline{1, K}.$$

Многопродуктовая динамическая транспортная задача с задержками сводится к статической многопродуктовой транспортной задаче на сети.

Если различные режимы доставки по струям предполагают разную стоимость при различной скорости, то в модель вводится корректировка. В этом случае каждому маршруту  $(p_i, p_j), i \neq j$ , задаётся общая пропускная способность и множество  $\{t_{ij}^1, t_{ij}^2, \dots, t_{ij}^{L_{ij}}\}$  транспортных задержек. Объём поставок, выходящий из  $p_i$  по пути  $(p_i, p_j)$  в момент  $t$  с задержкой  $t_{ij}^l$  прибывает в  $p_j$  в момент  $t + t_{ij}^l, c_{ij}^l(t), i \neq j$ , — расходы на перевозку единицы объёма поставок.

Согласованность вагонопотоков можно обеспечить с помощью метода динамического согласования (МДС). В качестве критерия оптимальности предлагается принять минимум транспортных расходов, расходов на хранение и затрат на перестройку производственных программ поставщиков.

Далее, в предположении о случайном разбросе во времени доставки и ритме потребления, предлагается “модель для оптимизации подвода грузов при существенном разбросе в потреблении”. Стохастическая постановка ДТЗЗ формулируется следующим образом: найти оптимальную по минимуму суммарных затрат на перемещение и простои динамическую структуру потоков с учётом штрафа от недопоставок при случайном разбросе в потреблении. Колебание времени хода учитывается в разбросе потребления.

Предложенные модели получили дальнейшее развитие в статье [4], где предлагается динамическая потоковая модель, которая позволяет рассчитать оптимальный вариант обращения вагонов разных собственников с рядом индивидуальных требований — ограничений по возмож-

ным маршрутам, времени оборота и т.п. Особенностью модели является отображение переходов вагонов из порожнего состояния в груженое и наоборот.

## 2 Задача формирования и маршрутизации грузовых составов РЖД

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.1) сортировочных станций, между которыми осуществляются перевозки грузовых вагонов. Имеется множество заказов на перевозку грузов. Каждый заказ состоит из станции отправления, станции назначения, а также некоторого количества вагонов, необходимых для перевозки груза. Каждый заказ имеет определённый 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  $k$  идентичных заказов. В дальнейшем будем предполагать, что множество заказов состоит из одновагонных заказов.



Рис. 1. Сортировочные станции РЖД

Запаздывание заказа определяется разницей между фактическим временем доставки последнего вагона с грузом на станцию назначе-

ния и директивными 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.

Необходимо составить план формирования грузовых составов и расписание их движения между станциями так, чтобы минимизировать взвешенное суммарное запаздывание всех заказов.

## 2.1 Задача формирования железнодорожных составов и маршрутов движения при фиксированных расписаниях

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 (заказов), 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*)  $G' = (V', A')$ . 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  $a_{i't''}^{\text{srt}} = (v_{i't'}^{\text{srt}}, v_{i't''}^{\text{srt}})$  между соседними вершинами сортировки станции  $i$  (вершины сортировки  $v_{it}^{\text{srt}}$  упорядочены по возрастанию моментов времени  $t$ ).

Для иллюстрации на рис. 2 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. 3 показана часть пространственно-временного графа  $G'$ , соответствующая станции  $i$  графа  $G$ .

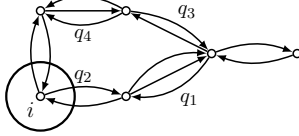


Рис. 2. Пример графа железнодорожной сети

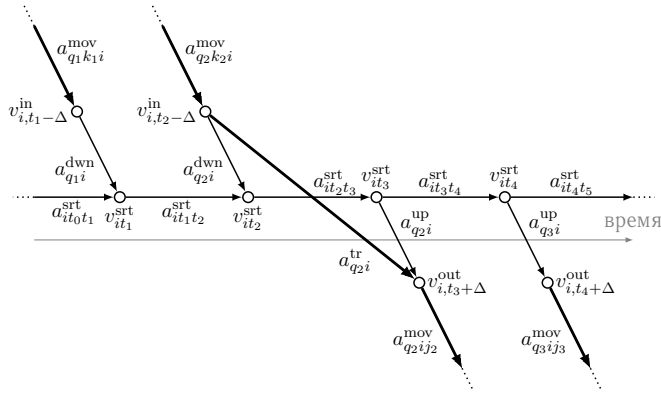


Рис. 3. Пример части пространственно-временного графа (станция  $i$ )

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

$$(v_{i,t_1-\Delta}^{in}, v_{i,t_1}^{srt}, v_{i,t_2}^{srt}, v_{i,t_3}^{srt}, v_{i,t_3+\Delta}^{out})$$

на рис. 3 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  $v_{i0}^{srt}$  и заканчивается в вершине  $v_{jH}^{srt}$ , где  $H$  — длина горизонта планирования (максимальное время прибытия поезда на его конечную станцию). Движение всех вагонов заказа  $(i, j) \in O$  соответствует по-

току величины  $n_{ij}$  от источника  $v_{i0}^{srt}$  до стока  $v_{jH}^{srt}$  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:

$$\sum_{a \in \delta_v^+} f_{ij}^a = n_{ij}, \quad \text{где } v = v_{i0}^{srt}, \quad \forall (i, j) \in O, \quad (1)$$

$$\sum_{a \in \delta_v^-} f_{ij}^a = \sum_{a \in \delta_v^+} f_{ij}^a, \quad \forall (i, j) \in O, \quad \forall v \neq v_{i0}^{srt}, v \neq v_{jH}^{srt}, \quad (2)$$

$$\sum_{a \in \delta_v^-} f_{ij}^a = n_{ij}, \quad \text{где } v = v_{jH}^{srt}, \quad \forall (i, j) \in O. \quad (3)$$

Ограничения (1) задают количество вагонов заказа  $(i, j)$ , находящихся на станции  $i$  в начальный момент времени. Ограничения (3) задают количество вагонов заказа  $(i, j)$ , находящихся на станции  $j$  в конечный момент времени. Условия (2) являются ограничениями сохранения потока.

Ограничения на максимальный вес и длину состава задаются следующим образом:

$$\sum_{(i,j) \in O} l_{ij} \cdot f_{ij}^a \leq L, \quad \forall a \in A^{\text{mov}}, \quad (4)$$

$$\sum_{(i,j) \in O} m_{ij} \cdot f_{ij}^a \leq M, \quad \forall a \in A^{\text{mov}}. \quad (5)$$

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 ин-

тервала, не превышает  $W$ :

$$\sum_{(i,j) \in O} \sum_{\substack{\forall a = a_{qk}^{\text{dwn}}: \\ t_{qk}^{\text{in}} \in (t', t'']}} f_{ij}^a + \sum_{(i,j) \in O} \sum_{\substack{\forall a = a_{qk}^{\text{up}}: \\ t_{qk}^{\text{out}} \in (t', t'']}} f_{ij}^a \leq W, \\ \forall k \in N, \quad \forall (t', t'') \in P. \quad (6)$$

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  $a_{itt'}$ . Ограничения, связывающие переменные  $z$  и  $f$  определяются как

$$z_{ijt} \geq 1 - \frac{1}{n_{ij}} f_{ij}^a, \quad (7)$$

где

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

Используя переменные  $z$ , целевая функция минимизации взвешенного суммарного запаздывания может быть записана как

$$\min \sum_{(i,j) \in O} \sum_{\substack{(t,t') \in I_j: \\ t' > d_{ij}}} w_{ij} \cdot (t' - \max\{t, d_{ij}\}) \cdot z_{ijt}. \quad (8)$$

## 2.2 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).

### 2.2.1 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  $\tau_{ijt}$  — 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.

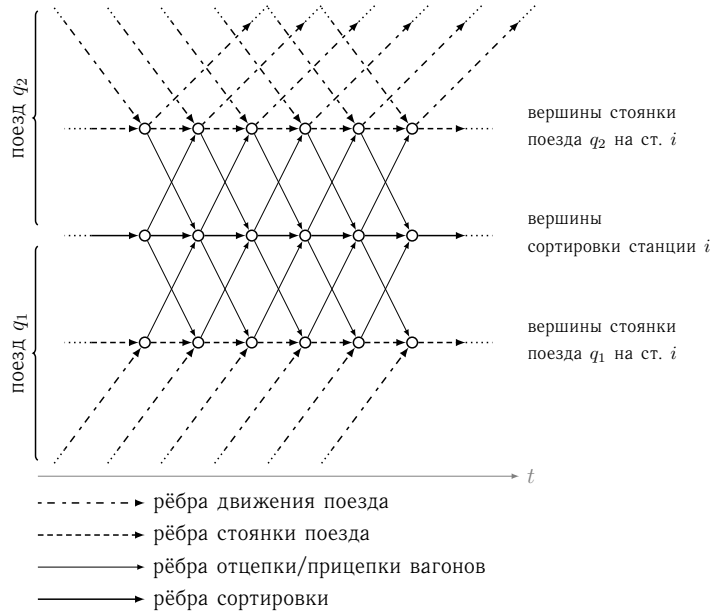
Для решения будет использоваться измененный и увеличенный пространственно-временной граф  $\bar{G} = (\bar{V}, \bar{A})$ . Вершины стоянки поезда  $v_{qit}^{\text{tr}}$  определены для каждой станции  $i \in N$ , каждого поезда  $q \in S_i^{\text{in}} \cup S_i^{\text{out}}$  имеющего станцию  $i$  в 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  $a_{qijt}^{\text{mov}} = (v_{qit}^{\text{tr}}, v_{jqit'}^{\text{tr}})$  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  $t' = t + \tau_{ijt}$ ;
- **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. 4 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. 4.** 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 по-

ездов:

$$\sum_{a \in \delta_v^{q+}} g_q^a = 1, \quad \text{где } v = v_{q,b(q),0}^{\text{tr}}, \forall q \in Q, \quad (9)$$

$$\sum_{a \in \delta_v^{q-}} g_q^a = \sum_{a \in \delta_v^{q+}} g_q^a, \quad \forall q \in Q, \forall v \in \bar{V}^{\text{srt}} \cup \bar{V}_q, \quad (10)$$

$$\sum_{a \in \delta_v^{q-}} g_q^a = 1, \quad \text{где } v = v_{q,e(q),H}^{\text{tr}}, \forall q \in Q. \quad (11)$$

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

$$\sum_{\substack{q \in Q: \\ \exists a = a_{qijt}^{\text{mov}}}} g_q^a \leq \gamma_{ijt}, \quad \forall t \in T, \quad \forall (i, j) \in E. \quad (12)$$

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 (1)–(3) 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  $q$ :

$$f_{ij}^a \leq n_{ij} \cdot g_q^a, \quad \forall (i, j) \in O, \quad \forall q \in Q, \quad \forall a \in \bar{A}_q \setminus \bar{A}^{\text{mov}}. \quad (13)$$

В связи с данным замечанием, ограничения (4)–(5) на максимальные длину и вес состава меняются на:

$$\sum_{(i,j) \in O} l_{ij} \cdot f_{ij}^a \leq L \cdot g_q^a, \quad \forall q \in Q, \quad \forall a \in \bar{A}_q^{\text{mov}}, \quad (14)$$

$$\sum_{(i,j) \in O} m_{ij} \cdot f_{ij}^a \leq M \cdot g_q^a, \quad \forall q \in Q, \quad a \in \bar{A}_q^{\text{mov}}. \quad (15)$$

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 огра-

ничения практически не отличаются от (6). 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  $U$ :

$$\sum_{q \in Q} \sum_{\substack{\forall a = a_{qit}^{\text{down}}: \\ t \in (t', t'')}} g_q^a + \sum_{q \in Q} \sum_{\substack{\forall a = a_{qit}^{\text{up}}: \\ t \in (t', t'')}} g_q^a \leq U, \quad \forall i \in N, \quad \forall (t', t'') \in \bar{P}. \quad (16)$$

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

$$z_{ijt} \geq 1 - \frac{1}{n_{ij}} f_{ij}^a, \quad (17)$$

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:

$$\min \sum_{(i,j) \in O} \sum_{t=d_{ij}+1}^H w_{ij} \cdot z_{ijt}. \quad (18)$$

### 2.2.2 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  $a_{qsi}^{\text{tr}} = (v_{qi}^{\text{in}}, v_{si}^{\text{out}})$  соответствует перецепке вагонов между поездами  $q, s \in N$ ,  $q \neq s$ , на станции  $i \in N$ .

На рис. 5 представлена часть графа  $\tilde{G}$ , соответствующая станции  $i$  (см. рис. 2). В данном примере станция  $i$  является конечной для поезда  $q_1$ , промежуточной для поезда  $q_2$  и начальной для поезда  $q_3$ .

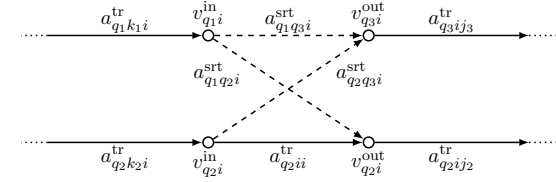


Рис. 5. Пример части графа  $\tilde{G}$  (станция  $i$ )

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  $q$  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  $(i, j) \in O$ .

$$\min \sum_{(i,j) \in O} \sum_{\substack{q \in Q_j^{\text{in}}, \\ a = a_{qj}^{\text{in}}}} \frac{w_{ij}}{n_{ij}} f_{ij}^a T_{qij} \quad (19)$$

$$T_{qij} \geq C_{qj} - d_{ij}, \quad \forall (i, j) \in O, \quad \forall q \in Q_j^{\text{in}}, \quad (20)$$

$$T_{qij} \geq 0, \quad \forall (i, j) \in O, \quad \forall q \in Q_j^{\text{in}}, \quad (21)$$

$$\sum_{q \in Q_i^{\text{out}}} f_{ij}^a = n_{ij}, \quad \text{где } a = a_{qi}^{\text{out}}, \quad \forall (i, j) \in O, \quad (22)$$

$$\sum_{a \in \delta_v^-} f_{ij}^a = \sum_{a \in \delta_v^+} f_{ij}^a, \quad \forall (i, j) \in O, \quad \forall v \neq v_{qi}^{\text{out}}, v \neq v_{qj}^{\text{in}}, \quad (23)$$

$$\sum_{q \in Q_j^{\text{in}}} f_{ij}^a = n_{ij}, \quad \text{где } a = a_{qj}^{\text{in}}, \quad \forall (i, j) \in O, \quad (24)$$

$$\sum_{(i,j) \in O} f_{ij}^a \leq n_{\max} x_{qsk}, \quad \forall a = a_{qsk}^{\text{srt}}, \quad (25)$$

$$\sum_{(i,j) \in O} f_{ij}^a \leq n_{\max}, \quad \forall a \in \tilde{A}^{\text{tr}}, \quad (26)$$

$$S_{si} \geq C_{qi} + \Delta + M(1 - x_{qsi}), \quad \forall i \in N, \forall q \in Q_i^{\text{in}}, \\ s \in Q_i^{\text{out}}, q \neq s. \quad (27)$$

Целевая функция (19) минимизирует суммарное среднее взвешенное запаздывание вагонов. Неравенства (20)–(21) определяют запаздывание каждого поезда на каждой станции относительно директивного срока каждого заказа доставки груза на данную станцию. Множество потоков вагонов задается ограничениями (22)–(24). Неравенства (25)–(26) ограничивают сверху величину потока проходящего по каждому ребру, т.е. задают максимальное число вагонов в каждом составе. Ограничения (27) задают отношения предшествования между прибытием поезда  $q$  на станцию  $i$  и отбытием поезда  $s$  с этой станции в случае перцепки вагонов  $s$   $q$  на  $s$ .

Недостатками формулировки (19)–(27) являются нелинейная целевая функция, а также наличие так называемых “*big-M*” ограничений (27), 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.

### 3 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  $1, 2, \dots, M$ ,  $M = \{m_1, \dots, m_M\}$  — 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  $\mathcal{L}$  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 меньшую 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  $\mathcal{L}$  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:

$$\max_{0 \leq x_j \leq 1} w_1 x_1 + w_2 x_2 + \dots + w_N x_N \quad (28)$$

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  $\alpha_{j_1}(t) = \alpha_{j_2}(t)$ .

## 4 Задача формирования грузовых потоков

### 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  $d_{vw}$  (включая прицепку, отцепку вагонов). Стоимость использования состава равна  $c_{vw}$ . Задачи, когда будут рассматриваться “не кратчайшие” пути, планируется исследовать в 2012 году.

Доставка товара (заказа), назначенного из  $i$  в  $j$ , должна быть выполнена к директивному сроку  $\bar{d}_{ij}$  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  $\bar{d}_j = \max_{i \in V} \bar{d}_{ij}$ .

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  $d \in \{1, 2, \dots, \bar{d}_j\}$  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:

$$\min \sum_{(v, w) \in A} c_{vw} g_{vw}, \quad (29)$$

$$\sum_{(i,v) \in A_j^+} f_{i,v,d_{iv}}^j = n_{ij} \quad \forall i, j \in N, \quad (30a)$$

$$\sum_{(w,v) \in A_j^+} f_{wvd}^j - \sum_{(v,w) \in A_j^+} f_{v,w,d+d_{vw}}^j = 0 \quad \forall v, j \in N, v \neq j, \quad (30b)$$

$$\forall d \in \{1, 2, \dots, \bar{d}_j\},$$

$$\sum_{(v,j) \in A_j^+} \sum_{d=1}^{\bar{d}_j} f_{vjd}^j = \sum_{i \in N \setminus \{j\}} n_{ij} \quad \forall j \in N, \quad (30c)$$

$$\sum_{j \in N} \sum_{(v,w) \in A_j^+} \sum_{d=1}^{\bar{d}_j} f_{vwd}^j \leq n_{\max} \cdot g_{vw} \quad \forall (v,w) \in A, \quad (30d)$$

$$\sum_{w \in N \setminus \{v\}} g_{vw} \leq q_v \quad \forall v \in N, \quad (30e)$$

$$f_{vwd}^j \in \mathbb{Z}_+ \quad \forall j \in N, \forall (v,w) \in A_j^+, \quad (30f)$$

$$\forall d \in \{1, 2, \dots, \bar{d}_j\},$$

$$g_{vw} \in \mathbb{Z}_+ \quad \forall (v,w) \in A, \quad (30g)$$

Целевая функция минимизирует затраты на использование составов. Ограничения (30a)-(30c) гарантируют доставку всех вагонов из пунктов отправления в пункты назначения. Условия (30d) означают, что не превышена вместимость составов. Наконец, неравенства (30e) обеспечивают выполнение ограничений на количество составов.

## 4.2 Использование решения задачи в деятельности РЖД

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. Каж-

дый 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.

Железнодорожная сеть задана графом  $G = (N, E)$ , где  $|N| \approx 76$ . Матрица  $N$  среднесуточных вагонопотоков между сортировочными станциями также известна. Значение  $n_{\max}$  обычно равно 71. Также известны целочисленные значения  $S = (s_i)_{i \in N}$  среднего времени переформирования (в часах) и количество обрабатываемых за день составов каждой станцией  $Q = (q_i)_{i \in N}$ . Величины  $s_i$  изменяются в пределах от 10 до 20 часов.

Матрица  $L = \{l_{ij}\}_{i,j \in N}$  содержит кратчайшие пути (в километрах) между каждой парой станций. Максимальное время доставки  $t_{ij}^{\max}$  вагона между станциями  $i$  и  $j$  обычно составляет  $\lceil l_{ij} * 24/330 \rceil$  часов.

Кроме того, известна целочисленная матрица  $T = \{t_{ij}\}_{i,j \in N}$ , где  $t_{ij}$  — среднее время (в часах), необходимое для передвижения между станциями  $i$  и  $j$  (включая время на замену локомотива на промежуточных станциях). Запаздывание, возникающее при движении поезда между станциями  $v$  и  $w$ , равно среднему времени переформирования состава на станции  $w$ , т.е.  $d_{vw} = t_{vw} + s_w$ . Затраты на состав определяются временем в пути:  $c_{vw} = t_{vw}$ ,  $\forall (v,w) \in A$ .

## 5 Задача формирования составов и расписания движения грузовых поездов

### 5.1 Общая постановка задачи

Рассмотрим проблему формирования составов и графиков движения грузовых поездов. Пусть  $S$  — множество станций, определяющих вершины графа  $G$  железнодорожной сети. Дорога от одной станции до другой разбита семафорами на участки. Даны множеств заказов  $N_s$  на каждой станции,  $N = \bigcup_{s \in S} N_s$  — множество всех заказов. Каждый заказ представляет собой один вагон, для которого известны пункт отправления и пункт назначения. Заказ  $j \in N$  поступает на станцию  $s$  в момент времени  $r_j^s$  и должен быть перевезен в пункт назначения не позже директивного срока  $d_j^s = r_j^s + p_j^s + \Delta$ , где  $p_j^s$  — продолжительность обслуживания заказа, которая определяется нормативно,  $\Delta$  — известная константа. В случае невыполнения заказа начисляется штраф  $z_j^s$ . Каждый заказ характеризуется своей значимостью  $w_j^s$  и весом (массой)  $v_j^s$ .

Пусть  $L$  — 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 тоннами.

**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 станции  $s \in S$  не может одновременно находиться более  $a_s$  составов.

**5. Ограничения по локомотивам и локомотивным бригадам.** Каждый локомотив  $l$  “приписан” к некоторому подмножеству станций

$G_l$ , причем  $G = \bigcup_l G_l, \bigcap_{i \neq j} G_i = \emptyset$ . 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  $C_j^s$  — время завершения доставки  $j$ -го вагона на станцию  $s$ . Можно рассмотреть несколько вариантов целевой функции.

1. Минимизация суммарного запаздывания:

$$\min \sum_{s \in S} \sum_{j \in N_s} w_j^s \max\{0, C_j^s - d_j^s\}.$$

2. Минимизация среднего времени нахождения состава в пути:

$$\min \sum_{s \in S} \sum_{j \in N_s} C_j^s.$$

3. Минимизации максимального временного смещения:

$$\min \max_{j \in N_s, s \in S} \{C_j^s - d_j^s\}.$$

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

$$\max \sum_{j \in \bar{N}} w_j^s - \sum_{j \in N \setminus \bar{N}} z_j^s.$$

Далее рассмотрим некоторые частные случаи описанной выше задачи.

### 5.2 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 интер-

валы времени  $[t_1, t_2], [t_3, t_4], \dots$

5. На участке, соединяющем станции:

- нет семафоров;
- 1 семафор;
- $k$  семафоров.

### 5.3 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.

### 5.4 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.

### 5.5 Задачи со станциями, расположенными в форме звезды

Для начала будем рассматривать четыре станции с расположением в форме звезды (см. рис. 6). 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  $[t_1, t_2], [t_3, t_4], \dots$

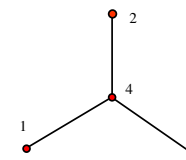


Рис. 6. Станции, расположенные в форме звезды

### 5.6 Полиномиальная разрешимость задач о станциях

В работе [12] 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.

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

В качестве целевого функционала рассматриваются 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. [32], 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 [12] 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 1.** 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 2.** Для последовательных групп выполняется

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

где

$$\begin{aligned} L &= S_{k-1}(t_l, t_r, \nu_l, \nu_r, \mu_r) + f_k(\delta_k), \\ I &= \min_{\nu, t} \{S_{k-1}(t_l, t, \nu_l, \nu, \nu - 1) + f_k(t + p\nu) + S_{k-1}(t, t_r, \nu, \nu_r, \mu_r)\}, \\ R &= f_k(t_r + p\nu_r) + S_{k-1}(t_l, t_r, \nu_l, \nu_r, \mu_r - 1) \text{ если } \mu_r > 0, \text{ иначе } +\infty. \end{aligned}$$

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.** Для параллельных групп выполняется

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

где

$$\begin{aligned} L &= P_{k-1}(t_l, t_r, \mu_r) + f_k(\delta_k), \\ I &= \min_t \{P_{k-1}(t_l, t, b - 1) + f_k(t + p) + P_{k-1}(t, t_r, \mu_r)\}, \\ R &= f_k(t_r + p) + P_{k-1}(t_l, t_r, \mu_r - 1) \text{ если } \mu_r > 0, \text{ иначе } +\infty. \end{aligned}$$

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 1 и 2).

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

---

**Algorithm 1** 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  $t_r \in S$  do
6:           for  $t_l \in S(t_l \leq t_r)$  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 2** 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

```

---

Сформулированные в разделах 5.2–5.5 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 [12] 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 [12] 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 5.2–5.5. Применение полученных алгоритмов для решения задач большой размерности представляется невозможно в силу, хотя и полиномиальной, но большой трудоемкости решения.

## 6 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 помощью 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

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

**Пример 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. 7. 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

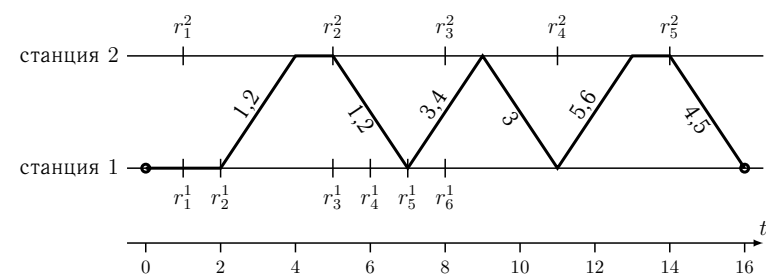


Рис. 7. Допустимое расписание для Примера 1

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. 7 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, большему, 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  $r_k^s$ , таким что,

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

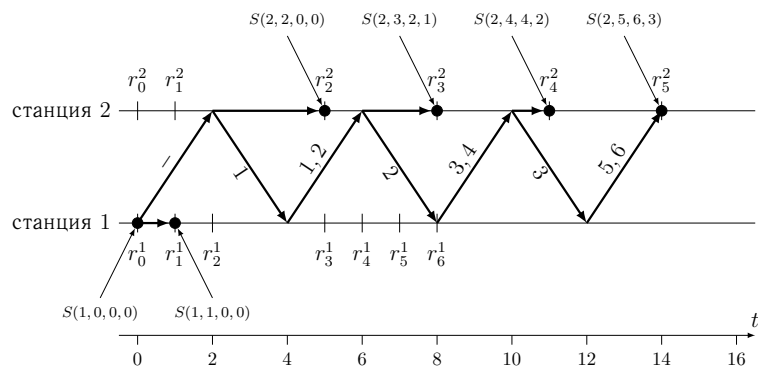
если  $s = 2$ , то  $k - q \leq j < k$  и  $r_k^2 \geq r_i^1 + p$ .

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

$$\min\{P(1, n + 1, n, m), P(2, m + 1, n, m)\}. \quad (31)$$

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

**Пример 2. (продолжение)** На рис. 8 показаны возможные переходы из состояния  $S(1, 0, 0, 0)$  в другие.



**Рис. 8. Возможные переходы из состояния  $S(1, 0, 0, 0)$  в Примере 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.

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

```
1: P  $\leftarrow P(s, k, i, j)$ ;  
2: i  $\leftarrow i$ ; j  $\leftarrow j$ ;  
3: s  $\leftarrow s$ ; t  $\leftarrow 0$ ;  
4: t  $\leftarrow r_k^s$ ;  
5: if s = 1 then  
6:   k1  $\leftarrow k + 1$ ; k2  $\leftarrow j$ ;  
7: end if  
8: if s = 2 then  
9:   k2  $\leftarrow k + 1$ ; k1  $\leftarrow i$ ;  
10: end if  
11: repeat  
12:   while  $r_{k1}^1 < t$  do  
13:     k1  $\leftarrow k1 + 1$ ;  
14:   end while  
15:   while  $r_{k2}^2 < t$  do  
16:     k2  $\leftarrow k2 + 1$ ;  
17:   end while  
18:   if s = 1 then  
19:     if k1 - i <  $q$  and P <  $P(1, k1, i, j)$  then  
20:        $P(1, k1, i, j) \leftarrow P$ ;  
21:     end if  
22:     terminate  $\leftarrow (r_{k1}^1 = t) \ \& \ (k1 - i \leq q)$ ;  
23:   end if  
24:   if s = 2 then  
25:     if k2 - j <  $q$  and P <  $P(2, k2, i, j)$  then  
26:        $P(2, k2, i, j) \leftarrow P$ ;  
27:     end if  
28:     terminate  $\leftarrow (r_{k2}^2 = t) \ \& \ (k2 - j \leq q)$ ;  
29:   end if
```

```
30: if not terminate then  
31:   if s = 1 then  
32:     wagons  $\leftarrow \min\{q, k1 - 1 - i\}$ ;  
33:     i  $\leftarrow i + \text{wagons}$ ;  
34:     s  $\leftarrow 2$ ;  
35:   end if  
36:   if s = 2 then  
37:     wagons  $\leftarrow \min\{q, k2 - 1 - j\}$ ; j  $\leftarrow j + \text{wagons}$ ;  
38:     s  $\leftarrow 1$ ;  
39:   end if  
40:   t  $\leftarrow t + p$ ;  
41:   P  $\leftarrow P + t \cdot \text{wagons}$ ;  
42:   if wagons = 0 then  
43:     idle  $\leftarrow \text{idle} + 1$   
44:   end if  
45: else  
46:   idle  $\leftarrow 0$   
47: end if  
48: until terminate or idle = 2
```

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 (31).

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.

## 7 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:

$$- C_{j'} \geq S_{j'} + \sum_{q=1}^Q p_q, \quad j' \in N';$$

$$- \forall i' \in N'_1, \forall j' \in N'_2 \quad C_{i'} \leq S_{j'} \text{ или } C_{j'} \leq S_{i'}.$$

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 [32], 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 одновременном 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 эквивалентны классическим одноприборным задачам теории расписаний [19] и, следовательно, в случае различных скоростей для поездов на сегменте, некоторые из задач оказываются NP-трудными [19]. Заметим также, что данные задачи могут быть легко сформулированы в терминах многоприборных задач теории расписаний с  $Q$  приборами [19].

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.

### 7.1 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 4.** 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  $\max\{r_{i'}, S_{i'} + p_{max}\}$  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

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

□

Лемма 4 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 поезд  $j$  движется 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 5.** 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:

$$\begin{cases} S_{j'_1} = r_{j'_1}, & C_{j'_1} = S_{j'_1} + P, \\ S_{j'_k} = \max\{r_{j'_k}, S_{j'_{k-1}} + p_{max}\}, & C_{j'_k} = S_{j'_k} + P, \quad k = 2, \dots, n', (*) \\ S_{j'_k} = \max\{r_{j'_k}, S_{j'_{k-1}} + P\}, & C_{j'_k} = S_{j'_k} + P, \quad k = 2, \dots, n', (**). \end{cases} \quad (32)$$

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

Согласно Лемме 4 построенное расписание будет допустимым. Более того, для представленных выше целевых функций, монотонных по времени окончания движения, в соответствии с Леммой 5 алгоритм (32)

по заданной оптимальной последовательности поездов построит оптимальное расписание.

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

## 7.2 Одноприборная задача

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 тре-

бований  $N_1$ ,  $|N_1| = |N'_1|$ , 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  $p = p_{max}$ ,  $st_1 = 2 \cdot TAIL_{right}$ ,  $st_2 = 2 \cdot TAIL_{left}$ , ес- ли  $j \in N_1$ , 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 требова- ний, соответствующую последовательности движения по- ездов  $(j'_1, j'_2, \dots, j'_n)$  между станциями, где требование  $j_k$ ,  $k = 1, 2, \dots, n$ , соответствует поезду  $j'_k$ , а также расписания, по- лученные из данных последовательностей, в которых каждое/ый требование/поезд обслуживается/начинает движение как можно рань- ше (для требований) или в соответствии с алгоритмом (32) (для поездов). Табл. 1 содержит соответствия требований  $j$  и поездов  $j'$ . В табл. 2 указаны соответствия целевых функций задач.

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 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 2. Соответствие значений целевых функций

Таким образом, можно заключить, что для указанных в таблице целевых функций оптимальная последовательность обслуживания тре-

бований  $(j_1, j_2, \dots, j_n)$  соответствует оптимальной последовательности движения поездов  $(j'_1, j'_2, \dots, j'_n)$ , т.е. задачи  $STRSP2| - | -$  можно све- сти к соответствующим задачам  $1|setup - times, N_1, N_2, p_j = p, -| -$  за полиномиальное время. В полученных задачах для одного прибора об- служивание всех требований  $j \in N_1$  начинается не раньше времени  $r = TAIL_{left}$ , 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 време- нами 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 [11]. Некоторые результаты для одноприборных задач с равными продолжительностями обслуживания требований описаны в [13, 39].

**Определение 1.** Назовем расписания для задач  $1|setup - times, N_1, N_2, p_j = p, -| -$  “сдвинутыми влево”, если обслуживание каждого тре- бования начинается в самый ранний из возможных моментов времени. Очевидно, что для любой из упомянутых выше задач существует опти- мальные 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 6.** 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”.  $\square$

### 7.3 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 расписания 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 7.** 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 7. 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 7).

#### Алгоритм 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 (32);
- 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 (32);
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  $(f_{min}, \sigma_{min})$ ;

```

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

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

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 7 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 7.

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 8.** 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  $STRSP2| \sum T_j$ .

#### 7.4 Задачи с частично упорядоченными подмножествами

**Лемма 9.** 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 10.** Предположим, что требования упорядочены в соответствии с леммой 9. 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 9.

Обе леммы могут быть доказаны по аналогии с леммой 7.

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  $x + y \rightarrow \max$  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 11.** 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 9. 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 запаздывающих требований из множества  $\{j_1, j_2, \dots, j_l\}$  предшествует обслуживанию требования  $H_n$  при расписании, а запаздывающие 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  $[t + p, T)$ .

Обозначим  $T_{max} = \max\{t | t \in \Theta\}$ . 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. Ниже представлен алгоритм решения данной задачи.

**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  $(f_{max}, \sigma_{max})$ ;

```

### Алгоритм 3

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

$F = \sum w_j(1 - U_j)$  — максимальное взвешенное число запаздывающих требований.

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. Сложность предложенных алгоритмов для разных типов задач

## 8 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.

Каждый заказ имеет директивный срок  $d_j = r_j + \delta$  — момент времени, до которого заказ может быть доставлен на станцию назначения без опоздания, где  $\delta$  — запас времени на доставку.

Все заказы должны быть обслужены  $q$  поездами, где  $q = \frac{n}{k}$ . Пусть  $C_j$  — время доставки заказа  $J_j \in N$ . Целевая функция задачи записывается как

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

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

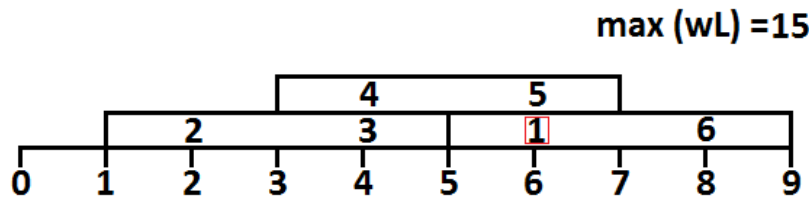


Рис. 9. Оптимальное расписание примера 1

**Пример 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. 9.

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  $J_1$ .

## 8.1 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

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

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  $[r_j, t'_j)$ .

### 8.1.1 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  $X$ .

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  $t^m \geq \max(r_{k \cdot m}, r(S_m), t^{m-1} + \alpha)$ ;
- (iii) все заказы  $J_l$ , для которых выполняется  $t^m + \alpha \geq t'_l$  должны уехать одним из поездов с номерами от 1 до  $m$  т.е.  $J_l \in S_m$ ;
- (iv) после отправки  $m$ -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  $t^m + \alpha$ , а значит, все заказы  $l$ , для которых выполняется неравенство  $t^m + \alpha \geq t'_l$ , не смогут отправиться поездами, номер которых больше  $m$ .

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

Обозначим за  $T_m$  текущее множество заказов, перевозимых  $m$ -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  $t^m + \alpha \geq t'_l$ , то согласно условию (iii) мы включаем заказ  $J_l$  во множества  $S_m, S_{m+1}, \dots, S_q$  и возвращаемся к проверке условия (ii). Если же такого заказа не существует, то условие (iii) может быть не выполнено только в том случае, если к моменту времени  $t^m$  доступно  $x > k$  неотправленных заказов из множества  $S_m$ . Множество этих заказов назовем  $X^0$ . Следовательно,  $x - k$  заказов должны будут отправиться первыми  $m - 1$  поездами. Чтобы отправить данные  $x - k$  заказов первыми  $m - 1$  поездами, будем по порядку во множествах заказов  $T_{m-1}, T_{m-2}, \dots$  искать такие заказы  $j$ , что  $t'_j > t^m + \alpha$ , до тех пор, пока их количество не достигнет значения  $x - k$ . Пусть последний такой заказ был найден на поезде с номером  $s$ . 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  $t'_j > t^m + \alpha$ , 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  $t' > t^m + \alpha$ . 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 поезда с номерами  $s, \dots, m$  и переходим к следующему шагу – отправке поезда  $m + 1$ . Алгоритм прерывает свою работу, если на каком-то шаге мощность какого-либо множества  $S_i$  стала больше  $k \cdot i$ .

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. Положим  $t^0 := -\alpha$ .

### Алгоритм $\Theta(N, y)$

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

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

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

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

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

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

6. Начиная с  $(i-1)$ -го поезда, найти множество заказов  $X'$ , которые будут исключены из рассмотрения.

7. Распределить множество заказов  $(T_s^{i-1} \setminus X') \cup X^0$  между поездами  $s, s+1, \dots, i$  в соответствии с условиями (i)-(iv). Если некоторый заказ  $J_j$  должен быть добавлен в  $S_m, m \in \{s-1, \dots, i\}$ , то  $Add_S(j, m)$ ,  $i := m$  и перейти на шаг 1.

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

### 8.1.2 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.

## 8.2 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 поезда  $i$ . 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 идет 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.

### 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 выпол-

няться 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 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. 10).

**Шаг 2.** Строим расписание  $\Theta(N, y)$  для множества  $N = \{J_1, \dots, J_6\}$  и

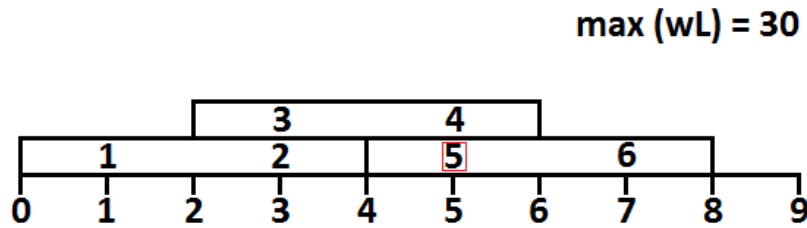


Рис. 10. Расписание  $\min(C_{\max})$  для множества заказов  $N$

$y = 30$ . 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. 11).

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

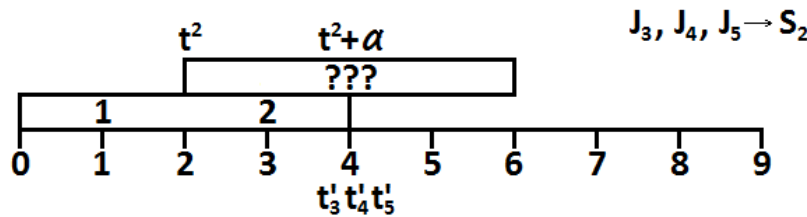


Рис. 11. Противоречие свойству (iii) при отправке второго поезда

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. 12).

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

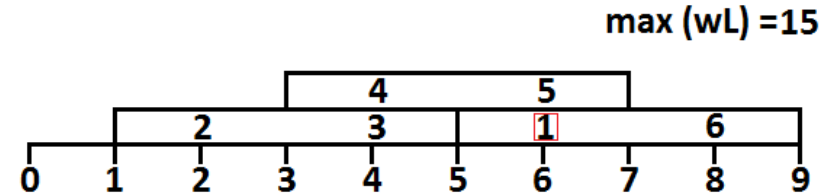


Рис. 12. Расписание  $\Theta(N, 30)$

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

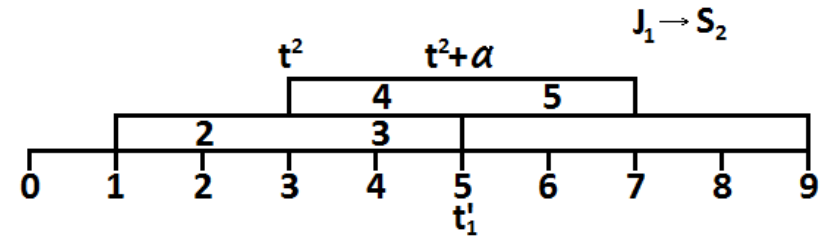


Рис. 13. Противоречие свойству (iii) при построении расписания  $\Theta(N, 15)$

## 9 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 главах, 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 [5]. В ходе работы над проектом были получены новые результаты в развитии метода изменения параметров.

### 9.1 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  $m \in \Omega$  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 12.** Пусть функция  $\rho(A, B)$ , определенная в пространстве параметров  $\Omega$ , при любом расписании  $\pi$  удовлетворяет неравенству

$$|F^A(\pi) - F^B(\pi)| \leq \rho(A, B), \quad (33)$$

тогда

$$F^A(\pi^B) - F^A(\pi^A) \leq 2\rho(A, B), \quad (34)$$

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 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 целевую функцию  $F(\pi)$  в виде

$$\min_{j \in N} \sum T_j(\pi) = \min_{j \in N} \max\{0, C_j(\pi) - d_j\}, \quad (35)$$

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 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

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |d_j^A - d_j^B|. \quad (36)$$

**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  $c, d$ .  $\square$

**Лемма 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

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq n \sum_{j \in N} |p_j^A - p_j^B|. \quad (37)$$

**Доказательство.**

$$\begin{aligned} \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \sum_{j \in N} |\max\{0, C_j^A - d\} - \max\{0, C_j^B - d\}| \leq \\ &\leq \sum_{j \in N} \max\{0, |C_j^A - C_j^B|\} \leq \sum_{j \in N} |C_j^A - C_j^B|. \end{aligned} \quad (38)$$

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 (38) получаем

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

$\square$

**Лемма 15.** Пусть у примеров  $A$  и  $B$  совпадают продолжительности обслуживания и директивные сроки требований:

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

тогда для любого расписания  $\pi$  справедливо неравенство

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq n \max_{j \in N} |r_j^A - r_j^B|. \quad (39)$$

**Доказательство.**

$$\begin{aligned} \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \sum_{j \in N} |\max\{0, C_j^A - d\} - \max\{0, C_j^B - d\}| \leq \\ &\leq \sum_{j \in N} \max\{0, |C_j^A - C_j^B|\} \leq \sum_{j \in N} |C_j^A - C_j^B|. \end{aligned} \quad (40)$$

Для раннего расписания имеем

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

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 (40) получаем

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

□

**Лемма 16.** Функция

$$\rho(A, B) = n \cdot \max_{j \in N} |r_j^A - r_j^B| + n \cdot \sum_{j=1}^n |p_j^A - p_j^B| + \sum_{j=1}^n |d_j^A - d_j^B| \quad (41)$$

удовлетворяет свойствам метрики.

**Доказательство.** Функция (41) 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 (41), а значит и для самой функции. □

**Лемма 17.** Функция (41) при любом расписании  $\pi$  удовлетворяет неравенству

$$|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B| \leq \rho(A, B). \quad (42)$$

**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 требований и одинаковые с примером  $A$  продолжительности обслуживания.

Используя леммы 13–15, получим

$$|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B| \leq |\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^C| + |\sum_{j \in N} T_j^C - \sum_{j \in N} T_j^D| +$$

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

□

Согласно леммам 12 и 17 для функции (41) выполняется

$$\sum_{j \in N} T^A(\pi^B) - \sum_{j \in N} (\pi^A) \leq 2\rho(A, B), \quad (43)$$

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

## 9.2 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.

$$\min f = n \cdot (y^r - x^r) + n \cdot \sum_{j=1}^n (y_j^p - x_j^p) + \sum_{j=1}^n (y_j^d - x_j^d), \quad (44)$$

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.

Для определения эффективности предложенной схемы была проведена серия численных экспериментов. Поиск полиномиально разрешимых примеров проводился в классах  $PR = \{r_j = r, p_j = p, j \in N\}$  и  $PD = \{d_j = d, p_j = p, j \in N\}$ . В первом случае решением является расписание, упорядоченное по неубыванию директивных сроков  $d_j$ , во втором – по неубыванию времен поступления  $r_j$ .

При поиске в классе  $PR$  полиномиально разрешимого примера, ближайшего к заданному примеру, необходимо найти минимум функции

$$f(r, p) = n \cdot \max_{j \in N} |r_j^A - r| + n \cdot \sum_{j=1}^n |p_j^A - p|. \quad (45)$$

**Лемма 18.** Минимум функции (45) достигается в точке  $(r = \frac{r_{max}^A + r_{min}^A}{2}, p \in \{p_1^A, \dots, p_n^A\})$ , где  $r_{max}^A = \max_{j \in N} r_j^A$ ,  $r_{min}^A = \min_{j \in N} r_j^A$ .

**Доказательство.** Зависящее от  $r$  слагаемое функции (45) представимо 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  $\frac{r_{max}^A + r_{min}^A}{2}$ . Пусть зависящее от  $p$  слагаемое  $f(p) = n \sum_{j \in N} |p_j^A - p|$  имеет минимум в точке  $p_0$ , тогда либо  $f'(p_0) = 0$ , либо  $p_0 \in \{p_1^A, \dots, p_n^A\}$ . Поскольку  $f(p)$  – 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.  $\square$

При поиске в классе  $PD$  полиномиально разрешимого примера, ближайшего к заданному примеру необходимо найти минимум функции

$$f(r, p) = \sum_{j \in N} |d_j^A - d| + n \cdot \sum_{j=1}^n |p_j^A - p|. \quad (46)$$

**Лемма 19.** Минимум функции (46) достигается в точке  $(p \in \{p_1^A, \dots, p_n^A\}, d \in \{d_1^A, \dots, d_n^A\})$ .

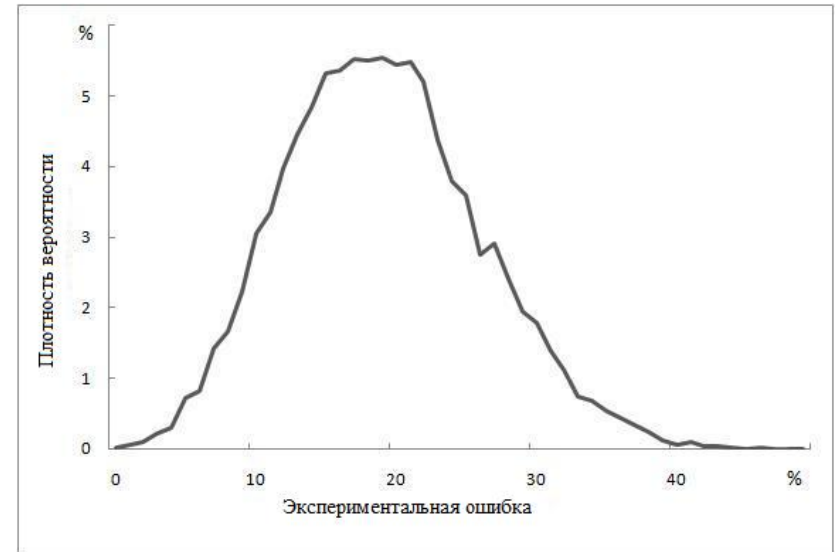
**Доказательство.** Лемма доказывается совершенно аналогично лем-

ме 18.

Эксперименты проводились по следующей схеме. Рассматривались значения  $n = 4, 5, \dots, 10$ , для каждого  $n$  генерировались по 10000 примеров. Параметры примеров генерировались равномерно на интервалах  $[0, 100]$  для  $r_j^A$ ,  $[1, 100]$  для  $p_j^A$  и  $[-100, 100]$  для  $d_j^A$ . 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 (43):

$$\Delta = \frac{F_e - F^*}{2\rho(A, B)}. \quad (47)$$

По итогам экспериментов строились зависимости плотности распределения ошибки  $\Delta$  для различных  $n$ . Все полученные плотности распределения имеют подобную гауссовскому распределению форму. Типичный вид полученной зависимости показан на рис. 14.



**Рис. 14.** Плотность распределения реальной погрешности приближенного решения, в процентах от теоретической.

Было обнаружено, что если поиск полиномиально разрешимого примера ведется в классе  $PD$ , то средняя погрешность решения составляет

$n$	Средняя ошибка в случае $PR$	Средняя ошибка в случае $PD$
4	19%	4,5%
5	19,5%	6,2%
6	19,2%	7,3%
7	19,6%	8,5%
8	19,3%	9,2%
9	19,4%	10%
10	19%	10,5%

**Таблица 4. Зависимость средней ошибки приближенного решения от количества требований**

19% от верхней оценки (43) и не зависит от количества требований  $n$ . Если же поиск проводится в классе  $PR$ , то средняя ошибка  $\Delta$  меняется с ростом  $n$  от 5% до 10%, при этом  $\Delta$  не превосходит 30%. Зависимость средней ошибки  $\Delta$  от  $n$  представлена в таблице 4.

### 9.3 Метод генерации колонок

Идея метода состоит в следующем.

1. Выбрать небольшое количество переменных и решить задачу, используя только их.
2. Найти неиспользуемую переменную, которая при включении улучшит значение целевой функции, либо определить, что таких переменных нет, во втором случае задача решена.
3. Включить переменную в задачу линейного программирования, перерешать и снова перейти к шагу 2.

Таким образом, с помощью метода генерации колонок возможно решать задачи линейного и целочисленного программирования большой размерности. Известны применения этого метода к различным задачам планирования, глобальной доставки, а также маршрутизации транспортных средств, в т.ч. и железнодорожных составов.

Предполагается применять данный метод для решения задач составления расписаний движения поездов.

## 10 Дальнейшие направления исследований

Дальнейшая работа будет направлена на

- 1) построение точных алгоритмов решения частных случаев задач формирования составов и графиков их движения (с цепочкой станций, замкнутой цепочкой и другими структурами расположения станций) (гл. 5);
- 2) разработку приближенных методов решения практических задач (гл. 2–4);
- 3) поиск эффективных метрик рассматриваемых задач, на основе которых будут сформулированы полиномиальные алгоритмы решения с гарантированной абсолютной погрешностью (гл. 9);
- 4) разработку программного обеспечения, реализующего построенные алгоритмы, и последующее его внедрение в отделы управления железнодорожными системами для минимизации издержек, увеличения объема перевозок, повышения пропускной и провозной способности железных дорог России.

Первая часть исследований будет использовать метод динамического программирования, а также связь задач формирования составов и графиков движения с классическими задачами теории расписаний.

Для нахождения решений с гарантированной погрешностью будет проводиться развитие метода изменения параметров на новые классы задач, в т.ч. задачи формирования составов и расписания их движения.

Кроме того, к данным задачам будет адаптирован метод генерации колонок, который широко используется для задач линейного и целочисленного программирования.

## 11 Список общих обозначений

- $n_{ij}$  — количество вагонов, которые необходимо доставить из  $i$  в  $j$ ;  
 $m_{ij}$  — масса вагона;  
 $l_{ij}$  — длина вагона;  
 $h_{ij}$  — длина пути между  $i$  и  $j$  станциями;  
 $Q_i^{in}$  — 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  $v$ ;  
 $c_{vw}$  — стоимость использования состава для транспортировки вагонов из вершины  $v$  в вершину  $w$ ;

- $u_{ij}(t)$  — объём перевозок порожних вагонов со станции  $i$  на станцию  $j$ ;  
 $C_i^j$  — момент доставки  $i$ -го заказа на станцию  $j$ ;  
 $N_j$  — множество заказов на  $j$ -й станции;  
 $L$  — множество локомотивов;  
 $B$  — множество локомотивных бригад;  
 $d_i$  — директивный срок доставки заказа  $i$  на станцию назначения;  
 $r_i$  — время поступления заказа  $i$  на станцию;  
 $p_i$  — продолжительность обслуживания заказа;  
 $w_i$  — коэффициент значимости;  
 $z_i$  — штраф за задержку вагона.

## Список литературы

- [1] Бодюл В.И. Математическая модель распределения вагонного парка по железным дорогам в условиях неравномерности грузовых перевозок // Вестник ВНИИЖТ. — 2006. — №3.
- [2] Козлов П.А., Владимирская И.П. Пути повышения обоснованности технологических решений // Вестник ВНИИЖТ. — 2009. — № 3. С. 8–12.
- [3] Козлов П.А., Владимирская И.П. Построение систем автоматизированного управления потоками вагонов разных собственников // Вестник ВНИИЖТ. — 2009. — № 6. С. 8–11
- [4] Козлов П.А., Владимирская И.П., Тушин Н.А. Оптимальное управление работой вагонов разных собственников // Вестник ВНИИЖТ. — 2010. — № 4.
- [5] Лазарев А.А. Теория расписаний. Оценки абсолютной погрешности и схема приближенного решения задач теории расписания. — М.: МФТИ, 2008.
- [6] Левин Д.Ю. Теория оперативного управления перевозочным процессом. — М.: “Транспортная книга”, 2008.
- [7] Литвак Б.Л. Алгоритм решения динамической транспортной задачи. // В кн.: Системы многосвязного управления. — М.: Наука, 1977. — С.59–69.
- [8] Макаров В.Л. Динамическая транспортная задача. // Труды Ленингр. инж.-экон. ин-та. — 1965. — вып. 58.
- [9] Сурин С.С. Динамическая транспортная задача и некоторые её обобщения. // В сб.: Методы вычислений. Вып. 3. — Л.: Изд-во ЛГУ, 1966.
- [10] 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.
- [11] 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.
- [12] Baptiste Ph. Batching identical jobs. // Math Meth Oper Res. — 2000. — No. 52. P. 355–367.
- [13] 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.
- [14] Beaujon G.J., Turnquist M.A. A Model for Fleet Sizing and Vehicle Allocation. // Transportation Science. — 1991. — No. 25.1. P. 19–45.
- [15] Billionet A. Using integer programming to solve the train-platforming problem. // Transportation Science. — 2003. — No. 37(2). P. 213–222.
- [16] Borndorfer R., Grotchel M., Lukac S., Mitusch M., Schlechte T., Schultz S., Tanner A. An auctioning approach to railway slot allocation. Technical Report 05-45. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2005.
- [17] Borndorfer R., Schlechte T. Solving railway track allocation problems. Technical Report 07-20. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2007b.
- [18] Brannlund U., Lindberg P.O., Nou A., Nilsson J.-E. Railway Timetabling using Lagrangian Relaxation. // Transportation Science. — 1998. — No. 32.4. P. 358–369.
- [19] Brucker P. Scheduling Algorithms. — Springer-Verlag. 2001. — P. 365.
- [20] Cacchiani V., Caprara A., Toth P. A column generation approach to train timetabling on a corridor. // 4OR. — 2008. — No. 6. P. 125–142.
- [21] Cai X., Goh C. J. A fast heuristic for the train scheduling problem. // Computers and Operations Research. — 1994. — No. 21(5). P. 499–510.
- [22] 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.

- [23] *Caprara A., Fischetti M., Toth P.* Modelling and solving the train timetabling problem. // *Operation Research*. — 2002. — No. 50. P. 851–861.
- [24] *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.
- [25] *De Luca Cardillo D., Mione N.*  $k$ -list  $\tau$  colouring of graphs. // *European Journal of Operational Research*. — 1998. — No 106. P. 160–164.
- [26] *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.
- [27] *Cornelsen S., Di Stefano G.* Track assignment. // *Journal of Discrete Algorithms*. — 2007. — No. 5(2). P. 250–261.
- [28] *Delorme X.* Modelisation et resolution de problemes lies a l'exploitation d'infrastructures ferroviaires. PhD thesis, Universite de Valenciennes et du Hainaut Cambresis, 2003.
- [29] *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.
- [30] *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.
- [31] *Gorman M.F.* Operating Plan Model Improves Service Design at Santa Fe Railway. // *Interfaces*. — 1998. — No 28.4 P. 1–12.
- [32] *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.
- [33] *Higgins A., Ferreira L., Kozan E.* Modeling Single-Line Train Operations. // *Transportation Research Record*. — 1995. — No. 1489 P. 9–16.
- [34] *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.
- [35] *Higgins A., Ferreira L., Kozan E.* Modeling Single-Line Train Operations. // *Transportation Research Record*. — 1995. — No. 1489. P. 9–16.
- [36] *Jungnickel D.* *Graphs, Networks and Algorithms*. — Springer, 2007.
- [37] *Jovanovic D., Barker P.T.* Tactical Scheduling of Rail Operations: The SCAN-I System. // *Transportation Science*. — 1991. — No. 25.1. P. 46–64.
- [38] *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.
- [39] *Kravchenko S., Werner F.* Parallel Machine Problems with Equal Processing Times: A Survey. // *Journal of Scheduling*. — 2011. — V. 14. No. 5. P. 435–444.
- [40] *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.
- [41] *LeBlanc L.J.* Global Solutions for a Nonconvex Nonconcave Rail Network Model. // *Management Science*. — 1976. — No. 23.2. P. 131–139.
- [42] *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.
- [43] *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.
- [44] *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.

- [45] *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.
- [46] *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.
- [47] *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.
- [48] *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.
- [49] *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.
- [50] *Peeters L.* Cyclic Railway Timetable Optimization. // PhD thesis, Erasmus Research Institute of Management. — 2003.
- [51] *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.
- [52] *Rodriguez J.* A constraint programming for real-time trains scheduling at junctions. // Transportation Research Part B. — 2007. — No. 41(2) P. 231–245.
- [53] *Sahin I.* Railway traffic control and train scheduling based on inter-train conflict management. // Transportation Research Part B. — 1999. — No. 33 P. 511–534.
- [54] *Sauder R.L., Westerman W.M.* Computer Aided Train Dispatching: Decision Support Through Optimization. // Interfaces. — 1983. — No. 13.6 P. 24–37.
- [55] *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.
- [56] *Sherali H.D., Suharko A.B.* A Tactical Decision Support System for Empty Railcar Management. // Transportation Science. — 1998. — No. 32.2. P. 306–329.
- [57] *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.
- [58] *Szpigiel B.* Optimal train scheduling on a single line railway. // Operations Research. —1973. — No. 72. P. 344–351.
- [59] *Turnquist M.A., Markowicz B.P.* An Interactive Microcomputer-Based Model for Railroad Car Distribution. // Working Paper, Cornell University, Ithaca, NY. — 1989.
- [60] *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.
- [61] *Zwaneveld P. J., Kroon L. G., Romeijn H. E., Salomon M., Dauzere-Peres S., van Hoesel S. P. M., Ambergen H. W.* Routing trains through railway stations: Model formulation and algorithms. // Transportaion Science. — 1996. — No. 30(3). P. 181–194.
- [62] *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.