

Министерство образования и науки Российской Федерации
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(государственный университет)
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

Выпускная квалификационная работа магистра
по направлению 010600 “Прикладные математика и физика”
студента 077а группы Никомарова Сергея Евгеньевича

"Оптимизация графиков оборота пассажирских поездов"

Научный руководитель:
д.ф.-м.н. Н.Н.Кузюрин

г. Москва

2016

Содержание

| | |
|---|-----------|
| СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ | 4 |
| ВВЕДЕНИЕ | 5 |
| 1. ПОСТАНОВКА ЗАДАЧИ | 7 |
| 2. ОБЗОР СУЩЕСТВУЮЩИХ МОДЕЛЕЙ..... | 8 |
| 2.3. Модель RSR | 12 |
| 2.4. Модель RSR-E | 15 |
| 2.5. Модели RSR-M и RSR-ME | 17 |
| 2.6. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ О МОДЕЛЯХ RSR, RSR-E, RSR-M, RSR-ME | 19 |
| 2.7. ДРУГИЕ МОДЕЛИ..... | 20 |
| 3. ИССЛЕДОВАНИЕ ЗАДАЧИ И ПОСТРОЕНИЕ РЕШЕНИЯ | 22 |
| 3.1. ВХОДНЫЕ ДАННЫЕ | 23 |
| 3.3. ОСНОВНЫЕ ПОНЯТИЯ МОДЕЛИ..... | 27 |
| 3.5. ОГРАНИЧЕНИЯ МОДЕЛИ..... | 31 |
| 3.6. КРИТЕРИИ МОДЕЛИ..... | 32 |
| 3.7. СВЕДЕНИЕ К ЗАДАЧЕ О НАЗНАЧЕНИЯХ..... | 34 |
| 3.9. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ | 39 |
| 3.10. ПОСТРОЕНИЕ МНОЖЕСТВА ОБОБЩЕННЫХ РАБОТ И НАЧАЛЬНОГО РЕШЕНИЯ ЗАДАЧИ | 39 |
| 3.11. ФУНКЦИЯ ОЦЕНКИ СТОИМОСТИ ПЕРЕКЛЮЧЕНИЯ | 40 |
| 3.11. ГЕНЕРАЦИЯ ДВУДОЛЬНОГО ГРАФА ДОПУСТИМЫХ ПЕРЕКЛЮЧЕНИЙ И ДОБАВЛЕНИЕ ФИКТИВНОЙ ВЕРШИНЫ..... | 43 |
| 3.12. РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ. | 44 |
| 4. ОПИСАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ | 46 |
| 4.1. ИСПОЛЬЗОВАННЫЙ ИНСТРУМЕНТАРИЙ | 46 |
| 4.2. ОБЩАЯ СХЕМА РАБОТЫ И СТРУКТУРА СИСТЕМЫ..... | 47 |
| 4.3. РЕЗУЛЬТАТЫ ПРИМЕНЕНИЯ СИСТЕМЫ | 48 |
| 5. ЗАКЛЮЧЕНИЕ | 50 |
| ПРИЛОЖЕНИЕ А. ДАННЫЕ, СВЯЗАННЫЕ С ОПИСАНИЕМ ВРЕМЕНИ И ВРЕМЕННЫХ ИНТЕРВАЛОВ..... | 53 |
| ПРИЛОЖЕНИЕ Б. ГРАФИКИ ДВИЖЕНИЯ И ОБОРОТА ПОЕЗДОВ. | 54 |
| ПРИЛОЖЕНИЕ С. ОБЩАЯ СХЕМА УСТРОЙСТВА ПРОГРАММНОЙ СИСТЕМЫ. | 58 |

Аннотация

Целями данной работы было изучение различных моделей пассажирских перевозок, решающих задачи поиска оптимального плана работ для железнодорожных составов, построение модели пассажирских перевозок, учитывающей технологические требования, принятые на российских железных дорогах, и составление оптимального плана работ для составов с помощью этой модели. Произведен обзор существующих моделей пассажирских перевозок, особое внимание уделено моделям семейства RSR. В качестве основы для собственной модели выбрана модель RSR-E. Основываясь на ней, разработана модифицированная модель перевозок, поддерживающая дополнительные критерии эффективности плана работ и принятые в РЖД технологические ограничения. На основе модели реализована программная система для составления плана работ железнодорожных составов в дальних пассажирских перевозках.

Список используемых сокращений и обозначений

РЖД – Российские Железные Дороги.

ж/д – железнодорожный.

Состав, ж/д состав – железнодорожный состав.

км – километры

АСУ – автоматизированная система управления

Введение

Организация железнодорожных перевозок – сложная и актуальная проблема в современном мире. Разница между эффективным и неэффективным решениями может выражаться в десятках миллионов рублей. На практике организация перевозок - это целый комплекс взаимосвязанных задач, для решения каждой из которых применяются свои методы. Очень многое зависит от нюансов конкретной задачи – грузовые это перевозки или пассажирские, пригородные или дальние. Разные типы перевозок имеют очень сильно отличающиеся технологические требования, что влечет за собой большие различия в методах решения задач. В данной работе рассматриваются пассажирские перевозки с фокусом на поездах дальнего следования.

Одной из важных задач при организации перевозок является поиск для железнодорожных составов плана работ, называемого *графиком оборота составов*. Она решается на уже составленном для перевозок по железнодорожной сети *графике движения*, под которым понимают множество поездов, которые будут осуществлять перевозки. Составленный график движения подразумевает, что для этих поездов уже определены их маршруты, расписания и даты отправления рейсов.

График движения включает тысячи рейсов поездов, в то время как число физических составов, выполняющих эти рейсы, значительно меньше. Очевидно, что для реализации графика движения каждый состав должен выполнять несколько рейсов. Т.е., выполнив одну поездку, состав отправляется (*оборачивается*) в другой рейс и т.д., пока не вернется в исходное состояние, с которого он начинал первый рейс. Такой цикл называется *оборотом* состава, а перечень рейсов и очередность их выполнения для всех поездов графика движения и будет графиком оборота.

Данная работа посвящена поиску эффективных графиков оборота составов. От правильности их составления зависит не только возможность реализации запланированного графика движения поездов, но и количество используемых ж/д составов, что особенно важно в дальних перевозках. Допустим, состав отправляется в путь утром, а возвращается обратно через сутки (тоже утром или днем). Снова отправиться в путь он сможет только утром следующего дня, полный оборот у него занимает трое суток. Если поезд ежедневный, то для обеспечения «ежедневности» ему потребуется три состава поездов. В то же время, если со станции вечером отправляется другой ежедневный поезд, который возвращается вечером через сутки (т.е. его выполнение его оборота также требует три состава), то для этих двух поездов можно составить *«сложный оборот»*. Не будем удерживать первый состав на

станции до утра следующего дня, а отправим его вечерним рейсом в этот же день. Тогда полный оборот состава, выполняющего оба рейса подряд – пять суток. При такой организации работы для тех же двух ежедневных поездов потребуется не шесть, а лишь пять составов. Стоимость одного пассажирского вагона на 2016 год составляет 35-40 млн. рублей, что делает экономию даже в один состав очень весомой.

В работе предложены критерии оценки эффективности графика оборотов, главным из которых является минимизация числа используемых им составов. Помимо этого критерия присутствует оценка устойчивости плана работ к задержкам составов. В реальной жизни график оборота редко составляют с нуля, обычно используют уже существующий, прошлогодний график. При оценке оптимальности найденного графика учитывается отклонение от предыдущего плана работ, т.к. специалисты РЖД предпочитают работать, внося малые изменения в уже существующие хорошо изученные решения.

В ходе работы рассматривается модель, учитывающая принятые на железной дороге ограничения, такие как требования поезда к типам вагонов выполняющего его состава или минимальное время стоянки состава между двумя станциями. Также расширено понятие периодичности хождения поездов, позволяющее применять модель для дальних пассажирских поездов и точнее вычислять количество задействованных в обороте составов. У приведенной модели показана сводимость к классической задаче о назначениях и предложен способ поиска оптимального относительно рассмотренных критериев графика оборота. Для построенной на основе модели программной системы приведены результаты расчетов для поездов нескольких железнодорожных депо.

1. Постановка задачи

Целью работы является составление графика оборота, удовлетворяющего ряду технологических ограничений РЖД и оптимального относительно заданных критериев. В работе предполагается, что график движения пассажирских поездов, для которого строится график оборота, уже составлен и задает начальный набор рейсов поездов, которые необходимо выполнить.

Данная работа рассматривает только пассажирские перевозки, уделяя особое внимание дальним (следующим на расстояние свыше 700 км) и местным (следующим на расстояние от 150 км до 700 км) перевозкам ([4]). В дальнейшем, говоря о дальних перевозках, будем под ними понимать и дальние, и местные перевозки. График оборота пригородных перевозок имеет дополнительное технологическое требование – все поезда в начальном наборе должны входить в один оборот. Как будет показано в данной работе, данное требование сильно усложнит задачу, делая ее NP-сложной (определение классов сложности задач, в частности, NP-сложности можно прочесть в [5] и [6]). Рассмотрение методов решения такой задачи выходит за рамки данной работы.

Использование эффективного графика оборота может привести к существенной экономии денежных средств, как уже было показано во введении. Для нахождения этого графика в работе были поставлены следующие задачи:

1. Изучить существующие модели для задачи составления графика оборота.
2. Разработать модель задачи, позволяющую найти график оборота пассажирских поездов, оптимальный относительно заданных критериев и удовлетворяющий технологическим требованиям РЖД.
3. Создать программную реализацию, использующую созданную модель для поиска оптимального графика оборота для дальних пассажирских поездов.

2. Обзор существующих моделей

Железнодорожные перевозки – узкая предметная область со своими специфичными понятиями. До обзора существующих моделей перевозок необходимо сначала ознакомиться с ними.

2.1. Основные понятия

Описание данных, связанных с железной дорогой удобно делать в терминах графов. Мы работаем с *железнодорожной сетью*, которую можно представить в виде графа G , вершинами которого являются станции, а ребрами – железнодорожные перегоны между ними. Для простоты считаем этот граф неориентированным. *Маршрут* на железнодорожной сети понимаем как маршрут в графе G . Определение маршрута в графе можно найти в [7].



Рисунок 1. Часть железнодорожной сети.

Одним из основных используемых понятий является *поезд*. Под поездом понимается объект, обладающий следующими свойствами:

- 1) Номер поезда;
- 2) *Маршрут поезда* – маршрут на железнодорожной сети, по которому ходит поезд. Станции отправления и прибытия – первая и последняя станции маршрута поезда, их называют *концами поезда*. Первую станцию также называют *началом поезда*, а последнюю – *концом поезда*.
- 3) Расписание поезда;
- 4) Множество рейсов поезда;
- 5) *Схема состава* – описание вагонов, входящих в состав. Представляется в виде набора пар – тип вагона + количество вагонов этого типа в составе. Схема состава поезда показывает, из каких вагонов должен состоять состав, выполняющий поезд.

Согласно принятым в РЖД соглашениям, номер поезда и его концы однозначно задают поезд в графике движения поездов.

Не стоит путать поезд и физический железнодорожный *состав*, который его выполняет. Под составом понимают физические, реальные вагоны, ходящие вместе по маршруту поезда в соответствии с его расписанием, т.е. *выполняющие* поезд.

Выполнение поезда составом называют *рейсом поезда*. Согласно правилам РЖД по составлению графика движения поездов, в день поезд может иметь не более одного рейса. Таким образом, рейс полностью задается датой отправления выполняющего его состава с начальной станции.

Типы данных, связанные с описанием времени подробнее описаны в приложении А. Время отправления поезда – время суток, в которое поезд отправляется.

Время выполнения – время, которое состав проводит в пути, длина временного интервала между моментами отправления поезда из начала и прибытием поезда в конец.

Понятие “расписание” упоминалось уже несколько раз, но, несмотря на его широкую распространенность в области перевозок, в данной работе большее значение имеют времена отправления поезда из начала, а также время выполнения рейса. Поэтому опишем расписание нестрого. Под *расписанием* понимают сопоставление каждой станции маршрута, кроме начальной и конечной станций, двух моментов времени (объекты типа DateTime, см. приложение А). Первый – *время прибытия поезда на станцию*, второй – *время отправления поезда со станции*, временной интервал между ними – *время стоянки* поезда на станции. У начальной станции есть только время отправления, у конечной станции – только время прибытия. В качестве даты для момента отправления с первой станции выбирается произвольная дата, т.к. в

расписании важны только временные интервалы, которые поезд проводит в пути или на станции, и времена суток, в которые состав прибывает на станцию или отправляется с нее дальше. Чтобы узнать, где состав будет находиться в заданный момент времени при выполнении конкретного рейса, вместо произвольной даты подставляют дату отправления этого рейса с начальной станции поезда.

На времена в расписании накладываются очевидные ограничения – время прибытия на станцию должно быть меньше или равно времени отправления поезда со станции, время прибытия поезда на следующую станцию маршрута больше времени отправления с предыдущей станции.

| Приб. | Стоянка | Отпр. | Расч. | Станция |
|-------|---------|-------|-------|---------------|
| | | 10:40 | | Вологда I |
| 11:06 | 40 | 11:46 | | Бушуйха |
| 13:31 | 10 | 13:41 | | Красильниково |
| 15:59 | 20 | 16:19 | | Якшанга |
| 17:01 | 40 | 17:41 | | Блины |
| 19:34 | | | | Киров |

Рисунок 2. Пример расписания для поезда из Кирова в Вологду.

Показаны только станции, на которых поезд есть стоянка.

Оборотом называется упорядоченный набор поездов $(tr_1, tr_2, \dots, tr_n)$. Состав при выполнении оборота последовательно идет по маршрутам поездов набора. В задаче составления оборотов поездов дальнего следования на поезда почти всегда накладывается ограничение: маршрут поезда tr_k заканчивается в станции начала маршрута tr_{k+1} (и то же самое для tr_n и tr_1). После выполнения оборота состав оказывается на первой станции маршрута первого поезда оборота, что позволяет ему снова отправиться на выполнение оборота. Таким образом, оборот является циклическим планом работ для составов.

Сам процесс сопоставления двух поездов для добавления в оборот называют *увязкой*.

В дальних перевозках используют обороты вида “туда и обратно”, состоящие только из двух поездов – прямого поезда из станции А в станцию Б и обратного из Б в А. Такой оборот называют *парой поездов* и это наиболее часто применяемый на практике тип оборотов среди дальних пассажирских поездов. Также его называют

простым оборотом (оборотов из одного поезда не существуют). Обороты из большего числа поездов будем называть *сложными оборотами*.

Время выполнения оборота – длина временного интервала между двумя последовательными стартами состава, выполняющего оборот. Можно также сказать, что время выполнения оборота – сумма времени, которое состав проводит в пути, и времени, которое он стоит на станциях в ожидании. Для времени выполнения оборота верен следующий очевидный результат:

Лемма 1. Время выполнения оборота является целым числом дней.

Доказательство: Пусть T_1 и T_2 – времена двух последовательных стартов состава, выполняющего оборот. Тогда время выполнения оборота по определению равно:

$$T_{об} = T_2 - T_1 \quad (1)$$

Рассмотрение моментов времени обычно начинается с выбора начального момента времени. Пусть начальный момент времени равен T_0 . Произвольный момент времени T можно представить как сумму нескольких составляющих:

$$T = T_0 + T_{дата} + T_{время суток} \quad (2)$$

$T_{время суток}$ – для заданного момента времени это длина временного интервала между этим моментом времени и полночью того же дня (ноль часов, ноль минут, дата та же).

$T_{дата}$ – часть временного интервала, состоящая из даты – целого числа лет, месяцев и дней. Для простоты можно считать, что это целое число дней, прошедших с момента начала отсчета. Тогда, подставляя (2) в (1):

$$T_{об} = (T_{дата 2} - T_{дата 1}) + (T_{время суток 2} - T_{время суток 1}) \quad (3)$$

Как уже упоминалось, рейсы поезда с разными датами отправления отбывают из начальной станции в одно и то же время суток. Поэтому время суток у обоих стартов одинаковое, и второе слагаемое в (3) равно нулю.

Первое слагаемое же равно какому-то натуральному числу, т.к. $T_{дата 2} > T_{дата 1}$ и оба целые числа. Значит, время выполнения оборота – целое число, что и требовалось доказать.

2.2. Существующие модели перевозок

Существующие модели перевозок обычно формулируются в терминах теории расписаний. В них рейсы поездов служат работами, а составы играют роль исполнителей работ. С точки зрения теории расписаний это задачи с несколькими

исполнителями. Различия в моделях часто связаны с различиями в технологических требованиях разных железных дорог, в одних моделях все исполнители одинаковы, в других же это не так и у работ есть требования к исполнителю, например, для выполнения работы нужен состав с определенной схемой состава. Очень часто, после формулировки задачи в терминах теории расписаний, она решается сведением к задаче целочисленного программирования. Существующие модели можно разбить на несколько типов:

1. RSR – Rolling Stock Rostering, наиболее простая, базовая модель перевозок.
2. RSR-E – более сложная модель, разрешающая движение пустых поездов между станциями.
3. RSR-M и RSR-ME – более сложные модели, учитывающие техническое обслуживание поездов. RSR-M соответствует модели RSR, а RSR-ME – модели RSR-E.

При сравнении моделей основными критериями являются:

1. Возможность применения модели для поиска оптимального графика оборота с учетом технологических требований РЖД.
2. Гибкость модели, возможность менять ее параметры. Способность модели описывать как можно больше различных ситуаций, встречающихся при организации перевозок.

2.3. Модель RSR

Модель RSR (Rolling Stock Rostering) является одной из базовых моделей для пассажирских перевозок. В ней для поездов используются следующие базовые предположения([1], [2]):

- 1) Для каждой станции число прибытий поездов равно числу отправлений;
- 2) Все поезда одинаковы – т.е. имеют одинаковую схему состава.
- 3) Все поезда ходят с одинаковой периодичностью, например, ежедневно.

При данных предположениях для заданного набора поездов с известными расписаниями требуется построить план выполнения их рейсов, использующий как можно меньше составов. Эту задачу также называют задачей RSR (*RSR problem*). По

набору поездов строится ориентированный граф G . Каждому поезду из входа ставятся в соответствие две вершины в G . Первая вершина представляет событие отправления поезда из начала, вторая – событие прибытия поезда в конец. Вершины, символизирующие отправление поезда, называют *вершинами отправления*, вершины, символизирующие прибытие, называют *вершинами прибытия*.

Ребра в графе G также бывают двух типов. Ребра первого типа соединяют вершину отправления поезда из входных данных с его вершиной прибытия и направлены от отправления к прибытию. Ребра второго типа представляют стоянки состава на станции между выполнением поездов. Они проводятся между вершинами, представляющими события прибытия или отправления на одной и той же станции. Добавляем ребра второго типа к G до тех пор, пока каждая вершина G не будет соединена ровно с двумя ребрами (рисунок 3.).

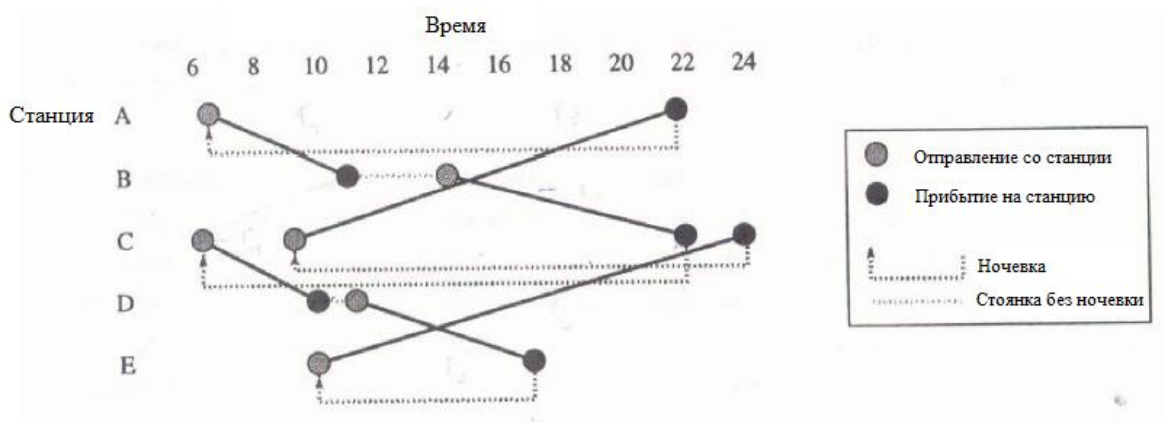


Рисунок 3. Пример графа G для шести поездов ([1]).

Каждый контур в графе G представляет какой-то циклическому плану работ для составов, т.е. обороту. В силу леммы 1 время выполнения оборота будет целым числом дней. Поэтому, чтобы понять, сколько необходимо составов для реализации оборота из ежедневных поездов, надо просто поглядеть на время выполнения оборота. Число требуемых составов равно числу дней, требуемых на выполнение оборота. Таким образом, обороты с меньшим временем выполнения более выгодны для нас, так как требуют меньше составов.

Пусть задан ориентированный граф $G_I = (V, E)$, где V – множество вершин графа, а E – множество его ребер. Под *покрытием вершин графа контурами* понимается поиск в G_I множества контуров, попарно не имеющих общих вершин,

таких, что любая вершина из множества вершин графа V принадлежит одному из контуров.

Составление плана выполнения рейсов поездов является поиском покрытия вершин графа G контурами. Будем считать стоимостью контура время выполнения соответствующего ему оборота. Тогда наиболее эффективным планом, очевидно, будет тот, которому соответствует покрытие циклами минимальной суммарной стоимости.

Существуют различные способы представления модели на компьютере:

- 1) Алгоритм минимального совершенного паросочетания;
- 2) Линейное программирование;
- 3) Алгоритм “Fast rostering”;
- 4) Некоторые другие способы.

Опишем алгоритм минимального совершенного паросочетания, подробнее о других алгоритмах можно прочитать в [1] и [2]. В данном алгоритме рассматриваются два множества вершин. Одно множество представляет все события отправления поездов, второе – все события прибытия. Для каждого события прибытия на станцию мы проводим ориентированные ребра от него ко всем событиям отправления с той же станции (рисунок 4). Рассмотрим теперь двудольный взвешенный ориентированный граф $G = (V, A)$ с функцией веса $c: A \rightarrow \mathbf{R}$, где:

V – множество вершин графа, является объединением описанных выше двух множеств вершин. Множество вершин, соответствующих событиям прибытия, образуют первую долю, а множество вершин, соответствующих отправлениям – вторую долю;

A – множество ребер графа G , состоит из ребер, построенных описанным выше способом;

$c(a)$ - вес ребра $a \in A$, равен времени ожидания состава между прибытием на станцию и соответствующим отправлением с нее.

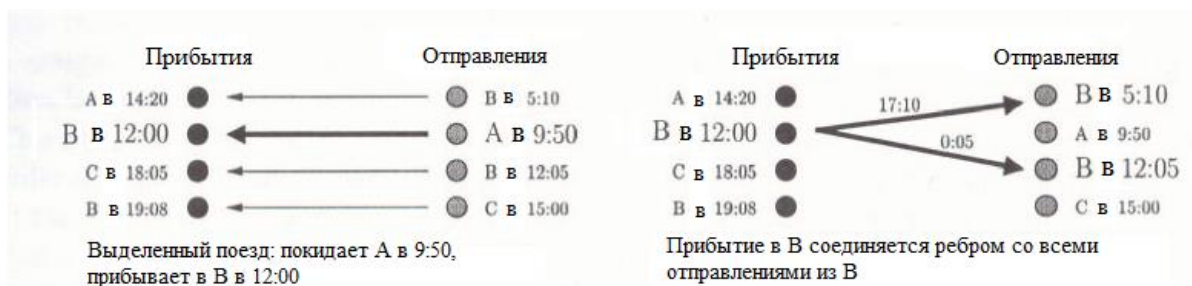


Рисунок 4. Алгоритм минимального совершенного паросочетания для четырех поездов ([1]).

Для построенного графа ищется минимальное совершенное паросочетание. Дадим его определение ([7]).

Для графа $G = (V, E)$, *паросочетание в G* — это множество попарно несмежных рёбер, то есть рёбер, не имеющих общих вершин. *Совершенным паросочетанием* называют паросочетание, в котором участвуют все вершины графа. То есть, любая вершина графа сопряжена ровно одному ребру, входящему в паросочетание. Для взвешенного графа *минимальным совершенным паросочетанием* называется его паросочетание, имеющее минимальный суммарный вес составляющих его ребер.

Поиск минимального совершенного паросочетания является давно известной задачей, для которой существует полиномиальный алгоритм решения. Найденное паросочетание дает нам необходимый план работ и число требуемых составов.

Для модели RSR справедлив следующий результат:

Теорема 1. Проблема RSR может быть решена за время $O(n \cdot \log(n))$.

Теорема доказывается для алгоритма “Fast rostering”, подробней с ее доказательством можно ознакомиться в [1] и [2].

Данная модель является довольно упрощенной и не учитывает многих технологических ограничений, используемых при организации железнодорожных перевозок. Например, нет важного ограничения на минимальное время, которое состав должен провести на станции после прибытия, предполагается, что он может уйти со станции мгновенно. Нет также и оценки решения с точки зрения консервативности и устойчивости к задержкам. Тем не менее, модель очень важна, так как закладывает фундамент для дальнейшей работы.

2.4. Модель RSR-E

Модель RSR-E (E от empty) является усложнением модели RSR. В данной модели при составлении плана работ для составов разрешено вводить пустые поезда для того, чтобы избавиться от первого предположения модели RSR – равенства для

станций числа прибывших поездов числу отправленных. Модифицированную задачу составления плана работ составов называют *задачей RSR-E*.

Под *пустым поездом из станции А в станцию В* понимается поезд из А в В, который вводят специально для соединения в оборот поезда, прибывающего на станцию А, с поездом, отправляющимся со станции В. Очень часто (но не всегда) такие поезда нигде не останавливаются и не принимают никаких пассажиров, за что они и получили свое название. В РЖД для таких поездов принят термин *засылочный поезд*. Засылочные поезда в основном используются в пригородных перевозках, например, при соединении поездов в единый оборот.

Пустых поездов нет среди множества поездов, поданного на вход. Их вводят уже при решении задачи. Сначала по начальному набору поездов строится граф, такой же, как и в модели RSR. Следующим шагом мы добавляем к построенному графу ребра, соответствующие пустым поездам до тех пор, пока каждая вершина не будет соединена ровно с двумя ребрами.

С построенным графом работают так же, как и с графом для RSR модели. Точно также в нем ищется покрытие вершин графа контурами минимальной суммарной стоимости. Для модели RSR-E тоже можно использовать алгоритм минимального совершенного паросочетания. Для этого его необходимо слегка модифицировать.

Как и в RSR, алгоритм строит двудольный граф G , первая доля которого состоит из вершин, символизирующих прибытия поездов на станции, а вторая – из вершин, символизирующих отправления со станций. Но теперь ребра от прибытия могут проводиться к любому отправлению, а не только к отправлениям с той же станцией. Ребра, идущие к отправлениям, несовпадающим по станции с прибытием, соответствуют пустым поездам. Их вес вычисляется как сумма времени хода состава от станции прибытия к станции отправления и времени стоянки состава в ожидании отправления. В построенном двудольном графе в дальнейшем ищут минимальное совершенное паросочетание теми методами, что и в RSR модели.

Задача о нахождении минимального совершенного паросочетания в двудольном графе имеет полиномиальный по времени алгоритм решения, что влечет за собой существование полиномиального по времени решения для задачи RSR-E.

Данная модель позволяет избавиться от части ограничений модели RSR. Пустые поезда позволяют описать использование засылочных поездов в пригородных перевозках РЖД. Тем не менее, она содержит те же недостатки, что были упомянуты при описании RSR модели.

2.5. Модели RSR-M и RSR-ME

Модели RSR-M и RSR-ME (M от maintenance) являются усложнением модели RSR, связанным с добавлением к модели описания технической поддержки. Модель RSR-M – модификация модели RSR, а модель RSR-ME – модели RSR-E. Модели описаны вместе, т.к. отличаются лишь наличием в RSR-ME уже описанных выше пустых поездов.

Добавление описания технического обслуживания к модели означает, что к входным данным задачи добавляется множество *станций техподдержки*. Составы должны периодически проходить техосмотр на этих станциях с целью выявления неисправностей. Таким образом, каждый оборот должен включать в себя заход на техническую станцию для проверки составов. Задачи составления плана работ для составов с учетом техобслуживания по аналогии с предыдущими моделями называют *задачами RSR-M и RSR-ME*.

В моделях предполагается, что технический осмотр и поддержка происходят мгновенно, при прохождении составом станции. Это более реалистичное предположение, чем может показаться на первый взгляд. Состав, пришедший на станцию для проверки, может быть сразу заменен другим, уже проверенным составом. К тому же на технических станциях на случай поломки состава держат запасные составы для замены.

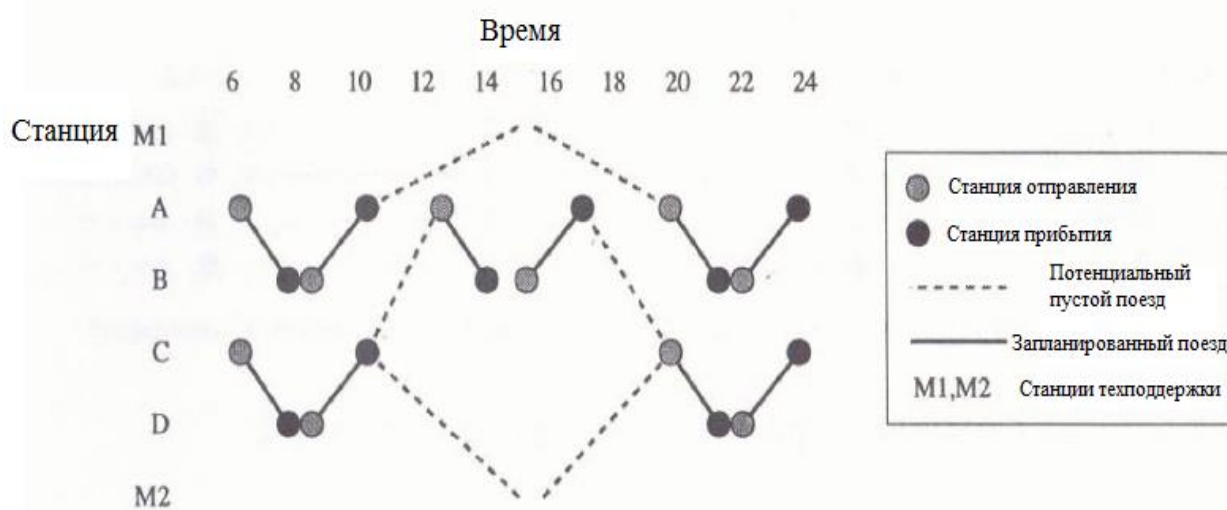


Рисунок 5. Граф для десяти поездов с двумя станциями техподдержки ([1]).

Графы моделей RSR-M и RSR-ME аналогичны графам для моделей RSR и RSR-E, вершины символизируют события прибытия и отправления поездов на станции, только теперь к станциям добавлены станции техподдержки, к которым могут идти пустые поезда. Каждая вершина соединена ровно с двумя ребрами. Пример графа модели на рисунке 5.

Введение техобслуживания сильно усложняет составление плана работ для составов. Приведем некоторые результаты, полученные о сложности задач RSR-M и RSR-ME. Далее будут использованы некоторые понятия из теории алгоритмов, такие как NP-задача. Мы не будем рассматривать их здесь, они подробно рассмотрены, например, в [5] и [6]. Напомним только, что алгоритм называется *C-приближенным*, если при любых исходных данных он находит допустимое решение со значением целевой функции, отличающимся от оптимума не более чем в C раз ([6]).

Теорема 2. Задачи RSR-M и RSR-ME являются APX-трудными.

Доказательство теоремы приведено в [2]. Мы не приводим формального определения APX и APX-сложной задачи, т.к. это потребует углубления в понятия теории сложности алгоритмов, не связанные с данной работой. Подробное описание есть в [6]. Отметим неформально, что APX-трудность задачи означает, что для задачи существует константа $r > 1$, такая, что даже аппроксимация задачи с мультипликативной ошибкой, не превышающей r , является NP-сложной. Таким образом, попытки построения все лучших и лучших аппроксимаций для задач RSR-M и RSR-ME в итоге всегда приведут к алгоритму с не полиномиальным временем выполнения.

Тем не менее, для задач существуют полиномиальные приближенные алгоритмы.

Теорема 3. Для задачи RSR-M существует 2-приближенный полиномиальный алгоритм.

Теорема 4. Для задачи RSR-ME, у которой пустые поезда имеют симметричные стоимости, существует 5-приближенный полиномиальный алгоритм. Под симметричными стоимостями понимается то, что стоимость пустого поезда от станции А к станции Б равна стоимости пустого поезда от станции Б к станции А.

Доказательства теорем 3 и 4 вместе с соответствующими приближенными алгоритмами приведены в [1] и [2]. Вместе с теоремой 2 они дают следующее утверждение ([2]):

Следствие 1. Задачи RSR-M и RSR-ME являются APX-полными.

Как видно из приведенных результатов, модели RSR-M и RSR-ME являются достаточно сложными сами по себе, без дополнительных ограничений. В то же время, при составлении оборотов для дальних пассажирских перевозок, которым мы отдаем особое внимание, времена стоянок составов на станциях всегда являются достаточно большими для проведения техобслуживания. Обычно составы проходят большую проверку в станциях-концах поезда. Введение станций техподдержки более актуально для пригородных перевозок, для дальних перевозок большую важность приобретает учет технологических ограничений.

2.6. Дополнительные сведения о моделях RSR, RSR-E, RSR-M, RSR-ME

При рассмотрении семейства моделей RSR мы предполагали периодичность найденных решений, в частности, что для ежедневных поездов расписания поездов будут повторяться каждый день. Если состав сегодня после выполнения рейса поезда номер 1 переключается на рейс поезда номер 2, то завтра он будет делать то же самое. Такие решения называют *однодневными назначениями* (one day assignment). В общем случае предположение о периодичности может быть не всегда верно, поиск решения для неперiodической ситуации является сложной задачей. Однако следующая теорема может быть полезна для проверки оптимальности решения для неперiodической ситуации.

Теорема 5. Для задач RSR, RSR-E и RSR-M оптимальное решение, найденное в предположении об однодневных назначениях поездов, является оптимальным решением задачи.

Подробнее о теореме написано в [1] и [2]. Теорема 5 показывает, что предположение об однодневных назначениях поездов не ухудшает решение задачи.

Однако, для модели RSR-ME это утверждение будет неверным, соответствующий пример приведен в [1].

2.7. Другие модели.

На сегодняшний день существует целый ряд работ, посвященных задаче составления плана работ для составов. Как уже отмечалось выше, часть различий в этих работах обусловлена разницей в технологических требованиях, принятых на разных железных дорогах. Например, для одной и той же организации РЖД, существует большая разница в требованиях к пригородным и дальним перевозкам.

Тем не менее, очень многие модели имеют сходство с моделями семейства RSR или даже основаны на них. Например, потоковая модель (flow model), предлагаемая для шведских и немецких железных дорог ([11]). Как и в моделях семейства RSR, в ней делается предположение о периодичности планов работ для составов. В модели строится схожий взвешенный ориентированный граф, вершины которого представляют события прибытия и отправления поездов на станции, а запланированным рейсам поездов соответствуют ребра между вершинами, соответствующими отправлению и прибытию поезда.

Потоковая модель учитывает множество ограничений, таких как минимальное время, которое состав должен провести на станции, требования поездов к типу вагонов, который должен иметь выполняющий их состав и другие ограничения. Также она имеет сложную, многокритериальную целевую функцию и поддерживает описание техобслуживания для составов.

Несколько отличающиеся модели были описаны в [3] для итальянских железных дорог и в [10] для китайских. В данных моделях также строится взвешенный ориентированный граф, но вершины этого графа представляют рейсы поездов, а не события прибытия и отправления поездов на станции. Ребро же соответствует переключению состава с рейса одного поезда на рейс другого.

Данные модели так же, как и потоковая модель, учитывают ряд технологических требований, содержат описание техобслуживания и стараются найти решение, оптимальное с точки зрения сразу нескольких критериев. Однако, в отличие от потоковой модели, в данных моделях все составы предполагаются одинаковыми.

Модель для итальянских дорог в качестве решения ищет один общий оборот для заданных поездов, аналогичное требование существует для пригородных перевозок РЖД. Китайская модель ищет набор оборотов, что как будет показано в дальнейшем, является более простой задачей, однако, она более сфокусирована на описании технического обслуживания составов.

В данном обзоре были описаны основные существующие типы моделей для составления графика оборота составов. Модели RSR, RSR-E, RSR-M и RSR-ME являются наиболее известными и хорошо изученными моделями. Однако сами по себе они малоприспособлены для реального применения. На практике применяют их модификации, которые содержат ряд дополнительных критериев и ограничений. Нами также были рассмотрены некоторые из таких моделей, применяемых для реальных железных дорог.

3. Исследование задачи и построение решения

В данной работе рассматривается модель для составления графика оборота составов для заданных поездов. Модель является модификацией модели RSR-E. Особое внимание отдается работе с дальними пассажирскими перевозками.

Добавление к модели описания технического обслуживания составов является сложной задачей, которая не является наиболее актуальной для дальних перевозок. На данном этапе техобслуживание составов не учитывается.

В качестве основы взята модель RSR-E, т.к. модели RSR-M и RSR-ME связаны с техобслуживанием составов и потому не подходят нам. Пустые поезда в RSR-E хорошо описывают засылочные поезда, применяемые в пригородных перевозках РЖД.

Так же как и в описанных моделях, мы предполагаем, что планы работ для составов периодичны. Мы несколько обобщим понятие периодичности со случая ежедневных поездов на случай произвольного периода.

Задача состоит в построении графика оборота составов для заданного множества поездов. Решение должно быть оптимальным относительно следующих критериев:

- 1) Количество используемых решением составов. Это самый важный для нас критерий, т.к. именно минимизация числа задействованных составов ведет к существенной экономии денежных средств.
- 2) Консервативность решения – в большинстве случаев составление нового графика оборота не делается с нуля, на практике специалисты часто модифицируют уже существующее решение. Желательно, чтобы решение не менялось при незначительных изменениях в задаче.
- 3) Устойчивость построенного решения к задержкам поездов.

Также решение должно удовлетворять технологическим требованиям РЖД:

- 1) Одним из основных требований является ограничение на время стоянки состава на станции между выполнением рейсов. Оно должно быть не меньше минимального допустимого времени стоянки.
- 2) Поезда требуют, чтобы состав, выполняющий их рейсы, имел определенную схему состава.

- 3) Переход состава с рейса одного поезда на рейс другого возможен только в двух случаях. Либо станция-конец первого поезда совпадает со станцией-началом второго, либо между этими станциями можно послать засылочный поезд.
- 4) Пользователь может вручную указать для любого поезда множество поездов, на рейсы которых разрешено переключаться составу после выполнения рейса этого поезда.

Входные данные для модели содержат:

- 1) Временной интервал, для которого решается задача;
- 2) Множество поездов, для которых надо составить график оборота;
- 3) Начальное решение задачи;
- 4) Множество станций увязки;
- 5) Набор управляющих параметров задачи;
- 6) Множество пар поездов, соответствующих вручную заданным пользователем допустимым переключениям состава между поездами;

На выход выдается либо список оборотов в виде множества упорядоченных наборов поездов, либо сообщение, что решения нет и частичное решение задачи – тоже в виде списка оборотов.

3.1. Входные данные

Опишем подробнее данные, подаваемые на вход задачи. Начнем с временного интервала, для которого она решается. Графики оборота составов составляются для какого-то временного интервала – на год, на лето или на месяц. Он задается двумя датами – датой начала и датой конца. Эти две даты подаются на вход задачи.

В нашей модели часть данных о поездах, например, маршрут и расписание, не используется и является избыточной. Под *поездом* мы будем понимать объект со следующими свойствами:

- 1) Номер поезда;
- 2) Станции отправления и прибытия поезда, которые, как уже говорилось ранее, называют *концами поезда*. Первую станцию маршрута поезда будем кратко называть *началом поезда*, последнюю - *концом поезда*;

- 3) Множество дат отправлений рейсов поезда;
- 4) Тип периодичности поезда;
- 5) Время суток, в которое поезд отправляется из начала (например, 14:30). Кратко будем называть его *временем отправления поезда*;
- 6) Время выполнения составом рейса поезда, т.е. время в пути (например, 1 день, 15 часов и 40 минут). Заметим, что время отправления и время выполнения однозначно определяют время суток, в которое поезд прибывает на конечную станцию. Будем называть это время *временем прибытия поезда*;
- 7) *Схема состава* – описание вагонов, входящих в состав. Представляется в виде набора пар – тип вагона + количество вагонов этого типа в составе.

Как уже отмечалось, в РЖД номер поезда вместе с его концами однозначно задают поезд в графике движения поездов. Номера и концов достаточно для того, чтобы идентифицировать поезд во входных данных.

В дальнейшем мы будем сравнивать между собой схемы составов поездов. Схемы составов будем считать одинаковыми, если множество типов вагонов в схемах совпадает, и они содержат одинаковое количество вагонов каждого типа.

Ранее уже говорилось, что для какого-то поезда рейс полностью задается датой отправления состава с первой станции маршрута, множество всех рейсов поезда соответственно задается множеством дат отправления поезда. Это множество дат также называют *периодичностью курсирования поезда*. Мы считаем, что все даты, составляющие периодичность курсирования, лежат внутри заданного для задачи временного интервала. Периодичности курсирования можно классифицировать по типам. Например, можно говорить о ежедневных или еженедельных поездах. Эта классификация зависит от временного диапазона, для которого изучается периодичность курсирования поезда. Описание алгоритма, вычисляющего тип периодичности поезда по множеству дат отправления его рейсов, выходит за рамки данной работы, мы считаем, что тип периодичности для поезда уже задан. Подробнее о периодичности написано в разделе 3.3.

Среди исходных данных также есть *начальное решение задачи*. Начальное решение также является множеством оборотов, оно содержит информацию для каждого поезда из входных данных о его присутствии в каком-то обороте. Этот оборот, если он есть, назовем *текущим оборотом* поезда. На практике у поездов текущий оборот почти всегда задан. Для дальних поездов это, обычно, пара поездов – простой универсальный оборот. Обычно в качестве начального решения выступает график

оборотов, который использовался ранее. Например, РЖД часто составляет графики оборота составов на год, так что в роли начального решения может выступать прошлогодний график оборота.

Мы рассматриваем не все множество станций-концов поездов из входных данных, а лишь некоторое его подмножество. Станции из этого подмножества назовем *станциями увязки*. Как уже говорилось, мы работаем с существующим планом работ для составов, менять его разрешено только на станциях увязки.

Набор управляющих параметров задачи – это множество различных констант, подаваемых на вход задачи. Для добавления гибкости модели многие используемые критерии и ограничения были сделаны параметризованными. Соответствующие параметры, например, минимальное время стоянки состава на станции увязки, содержатся в наборе управляющих параметров задачи.

Наша модель предполагает возможность задания вручную для поезда списка поездов, с которыми его можно увязывать. Это можно представить в виде множества пар (tr_0, tr_i) , где tr_0 - поезд, для которого задается ограничение, tr_i - один из поездов, с которыми tr_0 можно увязывать, $i \in N$. Объединение всех таких множеств и подается на вход. По умолчанию предполагается, что данное ограничение не задано, т.е. поезд может увязываться с любым другим поездом из входных данных, если это не запрещено другими ограничениями.

Как и во многих других существующих моделях, мы предполагаем, что все поезда, поданные на вход задачи, ходят с одинаковой периодичностью. В RSR моделях принималось предположение об однодневных назначениях поездов. В дальних пассажирских перевозках часто встречаются поезда, которые ходят два раза в неделю или один раз в несколько дней. Для описания таких поездов необходимо расширить понятие периодичности.

3.2. Периодичность поездов

О периодичности поезда говорят в контексте некоторого выбранного временного интервала. Мы рассматриваем периодичность относительно временного интервала, для которого решается задача. В дальнейшем, мы будем называть его *интервалом дат*.

Назовем временной интервал длиной ровно n дней, $n \in N, n \geq 1$, n -дневным временным интервалом. Он задается датой начала и числом дней в интервале.

Рассмотрим такой интервал вместе с датами отправления рейсов поезда, лежащими в этом интервале. Эти даты можно рассматривать как относительные, относительно начала интервала. Назовем их днями хождения поезда в n -дневном интервале.

Если заданный интервал дат можно покрыть одинаковыми непересекающимися m -дневными интервалами такими, что для каждого из них у поезда есть ровно k рейсов, и дни хождения поезда во всех интервалах одинаковы, то будем говорить, что для этого интервала дат поезд имеет *период хождения* в m дней и k рейсов в периоде.

Дадим определения бита и битовой строки. *Битом* в информатике называют основную единицу хранения информации, двоичный разряд. Бит может содержать только значения ноль и один. Последовательность из n битов, где n - неотрицательное целое число, называют *битовой строкой длины n* . *Круговой сдвиг битовой строки вправо* – сдвиг битовой строки вправо, при котором потерянные биты не выбрасываются, а записываются слева, в начало битовой строки. Подробнее о битовом представлении данных и битовых операциях с данными, такими как битовый сдвиг в ([9]).

При наличии у поезда периода хождения, для поезда можно поставить в соответствие битовую строку, длина которой равна количеству дней в периоде. Единицы в ней стоят на позициях, соответствующих дням, по которым у поезда есть рейсы. Например, “1010” означает, что поезд ходит два раза в четыре дня - в первый и третий день периода. Такую битовую строку будем называть *битовой строкой периодичности отправления поезда* или кратко *битовой строкой отправления*. Количество дней в периоде, количество рейсов поезда в периоде и битовая строка периодичности определяют *тип периодичности поезда* в заданном интервале дат. Если у периодичности курсирования поезда нет периода в заданном интервале дат, то считаем это особым типом периодичности – *без периода*.

Равными будем считать типы периодичности, если у каждого из них либо нет периода, либо он одинаковый и битовые строки отправления равны или одну можно перевести в другую круговым сдвигом вправо. Это также означает, что у равных типов периодичности совпадает число дней в периоде и количество рейсов в периоде. Для примера, поезд, ходящий каждую среду и поезд, ходящий каждый понедельник, имеют одинаковый тип периодичности.

Мы можем определить, что битовые строки совместимы круговым сдвигом вправо, следующим образом. Пусть у нас есть две битовые строки одинаковой длины (если длины разные, то строки нельзя перевести сдвигом). Выберем одну из строк и удвоим ее. Если строка имела вид ”010”, то после удвоения она будет “010010”.

Теперь, если вторая строка совместима с первой сдвигом, то вторая строка должна быть подстрокой удвоенной первой строки. Проверяя, так ли это при помощи алгоритма поиска подстроки в строке, получаем ответ.

Дата отправления поезда с начальной станции и дата прибытия поезда в конечную станцию могут не совпадать. Тогда для дат прибытия поезда на конечную станцию можно, по аналогии с битовой строкой отправления поезда, ввести *битовую строку прибытия поезда*. Ее легко вычислить, зная битовую строку отправления поезда и время, которое поезд провел в пути. Достаточно просто сделать круговой сдвиг битовой строки отправления поезда вправо, взяв за величину сдвига количество дней между датой прибытия поезда в конец и датой отправления поезда из начала.

3.3. Основные понятия модели

Опишем теперь подробнее нашу модель. Для описания будем использовать термины теории расписаний. Рейсы поездов, для которых нам необходимо предоставить составы, будем рассматривать как *работы*. Сами составы будем считать *исполнителями* этих работ. Исполнители в модели имеют единственный режим выполнения работ. Прерывания во время выполнения работы запрещены.

Для задачи построения оборотов пригородных поездов в большинстве случаев каждая работа может выполняться любым исполнителем, все исполнители одинаковы. Для дальних поездов типичной является ситуация, когда у работы есть требования к исполнителю по схеме состава.

Атомарной работой (аналог train service в [3]) будем считать выполнение составом рейса поезда. Она определяется:

- 1) *временами начала работы и конца работы*. Это времена отправления и прибытия поезда;
- 2) *станциями-концами работы* – станции-концы поезда;
- 3) номером поезда;
- 4) *периодичностью начала и конца* – битовыми строками отправления и прибытия поезда.
- 5) Схемой состава поезда.

Переход исполнителя к выполнению работы *Б* после выполнения работы *А* называется *переключением с работы А на работу Б* (сокращенно - *переключением*).

Работу *A* также будем называть *первой работой переключения*, а работу *B* – *второй работой переключения*. Поезда, соответствующие работам *A* и *B*, называются *поездами переключения*.

На возможность переключения между работами накладываются ограничения. *Допустимыми переключениями* называются переключения между работами, удовлетворяющие ограничениям задачи. Переключения, не удовлетворяющие ограничениям, соответственно называются *недопустимыми*.

Чтобы не рассматривать отдельно поезда, для которых мы не меняем план выполнения составами, введем понятие обобщенной работы.

Обобщенной работой будем называть упорядоченную последовательность последовательно выполняемых атомарных работ. Для нее определяем:

- 1) *Время начала работы* – время начала первой атомарной работы;
- 2) *Время конца работы* – время конца последней атомарной работы;
- 3) *Станции-концы работы* – станция-начало первой атомарной работы и станция-конец последней.
- 4) *периодичностью начала и конца* – периодичность начала первой атомарной работы и периодичность конца последней атомарной работы.
- 5) *Схемами начала и конца* – схемы составов первой и последней атомарной работы.

В определении переключения не важно, являются ли работы атомарными или обобщенными. *Поездами переключения между обобщенными работами A и B* будем теперь называть поезд, соответствующей последней атомарной работе в обобщенной работе *A* и поезд, соответствующей первой атомарной работе в обобщенной работе *B*. Атомарную работу можно рассматривать как обобщенную, в которую входит только одна атомарная работа.

Атомарные работы будем считать *одинаковыми*, если им соответствует один и тот же поезд. Обобщенные работы будем считать *одинаковыми*, если они содержат одинаковое число атомарных работ, и их атомарные работы, стоящие на позициях с одним и тем же номером, одинаковы. В противном случае будем говорить, что работы *отличаются*.

Целью введения обобщенных работ является упрощение и оптимизация модели. Благодаря обобщенным работам исключается необходимость рассматривать в модели отдельно такие переключения между атомарными работами, которые точно не изменятся при решении задачи. Обобщенную работу можно понимать как *блок работ*. В качестве простого примера можно привести пару поездов, где только одна из станций-концов пары является станцией увязки. На второй станции мы не разрешаем

менять план работ, поэтому рассматривать оба поезда как отдельные объекты будет ненужным усложнением, всю пару можно рассматривать в качестве одной обобщенной работы. В общем случае в обобщенные работы попадают атомарные работы, поезда которых имеют одинаковый текущий оборот.

В дальнейшем речь будет идти в основном об обобщенных работах, поэтому, говоря работы, будут иметься в виду обобщенные работы, если не указано иное.

Стоимостью переключения с работы A на работу B называется число, сопоставляемое переключению. Оно вычисляется по данным о работах A и B согласно заданным для задачи критериям. Каждый критерий вносит в стоимость свой вклад, она служит оценкой “качества” переключения. При решении задачи мы пытаемся найти решение с минимальной суммарной стоимостью переключения.

Оборот в модели является набором работ для исполнителя, выполняемых последовательно. При выполнении последней работы оборота исполнитель переключается на первую работу.

3.4. Описание модели

Теперь, описав основные используемые термины, перейдем к построению самой модели. Задачу, переформулированную с использованием терминов теории расписаний, можно представить в виде взвешенного ориентированного графа. Каждую работу из множества поданных на вход работ будем считать вершиной этого графа (аналогично модели из [3]). Если переключение между двумя работами является допустимым, то между вершинами графа, представляющими эти работы, есть ориентированное ребро, направленное от первой работы переключения ко второй. В том числе допускаются петли, если переключение работы с самой на себя является допустимым. Примером петли будет простой оборот, который состоит из одной обобщенной работы, переключающейся на саму себя. Вес ребра равен стоимости переключения между работами. Назовем полученный граф графом допустимых переключений. Он и служит моделью для задачи составления графика оборота.

Так же, как и в моделях семейства RSR, оборот является контуром в этом графе, а задача построения графика оборота равносильна задаче нахождения покрытия вершин непересекающимися контурами минимального суммарного веса. Действительно, обороты не могут пересекаться по работам, и каждая работа должна войти в один из оборотов решения, следовательно, решение задачи представляется в виде покрытия вершин графа непересекающимися контурами. Сумма весов ребер,

входящих в контур, равна сумме стоимостей переключений между работами в соответствующем ему обороте. Мы ищем обороты с минимальной суммарной стоимостью переключения, т.е. покрытие минимального суммарного веса.

Решением задачи построения графика оборота назовем вершинное покрытие графа допустимых переключений непересекающимися контурами. Это эквивалентно покрытию множества всех обобщенных работ задачи непересекающимися по работам оборотами. Также можно сказать, что множество текущих оборотов поездов из входных данных задает начальное решение задачи построения графика оборота.

В случае пригородных поездов из-за существующих технологических требований, принятых на железной дороге, задача усложняется. При составлении оборотов пригородных поездов для заранее составленного набора поездов, например, поездов из одного депо, требуется увязать их все в один оборот. Как показывает следующая теорема, это сильно усложняет задачу.

Теорема 6. Задача составления графика оборота составов, требующая, чтобы решение состояло из одного оборота, является NP-трудной.

Доказательство: Покажем, что к данной задаче можно свести задачу коммивояжера – поиск во взвешенном графе цикла через все вершины минимальной стоимости. Рассмотрим некоторый взвешенный ориентированный граф G , задающий ассиметричную задачу коммивояжера, с неотрицательными весами ребер. Будем считать вершины этого графа работами, а ребро между двумя вершинами – переключением между работами, которые они представляют. Вес ребра – стоимость переключения.

Таким образом, G является графом допустимых переключений, задающим задачу составления графика оборота. Нахождение единого оборота минимальной стоимости для этой задачи решит и задачу коммивояжера – оборот задает цикл минимальной стоимости через все вершины. Время сведения задачи коммивояжера к задаче о нахождении оборота полиномиально. Как известно, задача о коммивояжере является NP-трудной, следовательно, по определению NP-трудной задачи, и исходная задача является NP-трудной, что и требовалось доказать.

На практике одним из способов решения описанной выше задачи является поиск покрытия вершин в графе набором контуров минимального веса. Для найденного покрытия затем ищут приближенное решение. Также эта задача решалась для итальянских железных дорог в [3].

3.5. Ограничения модели

Опишем используемые в модели ограничения. Одним из наиболее важных ограничений является ограничение на время переключения между работами.

Временем переключения с работы А на работу В называется длина временного интервала между временем конца работы А и временем начала работы В. *Минимальным временем переключения между работами* назовем минимальную допустимую величину времени переключения. Оно подается на вход задачи в множестве управляемых параметров. Если время переключения между работами меньше минимального времени переключения, то переключение является недопустимым.

Для задачи построения оборотов поездов дальнего следования почти всегда присутствует требование о совпадении концов работ у переключения. Требуется, чтобы при переключении станция-конец первой работы совпадала со станцией-началом второй. Эти станции назовем *станциями переключения*. Если станции переключения не совпадают, то переключение является недопустимым.

Если для задачи разрешено использование засылочных поездов (пустые поезда из модели RSR-E), то переключение будет допустимым, а между станциями переключения вводится засылочный поезд. Параметр, определяющий можно или нет использовать засылочные поезда, подается на вход во множестве управляемых параметров.

Если работы задачи различаются по требованиям к схеме состава, то используется *ограничение на схему составов*. Для этого мы применяем *функцию сравнения схем составов с заданной точностью*. Эта функция принимает на вход схему конца первой работы переключения, схему начала второй работы переключения и заданный параметр, называемый *точностью*, а возвращает ответ - являются ли схемы составов совместимыми с заданной точностью.

Тогда ограничение на схему составов определяется функцией сравнения. Чтобы переключение было допустимым, необходимо, чтобы функция сравнения для заданной (из множества управляющих параметров) точности возвращала для схем составов работ переключения результат “совместимы”.

Такое гибкое ограничение позволяет учитывать ситуации, встречающиеся в реальной жизни, когда состав со схемой состава, слегка отличающейся от схемы, требуемой поездом, используется для выполнения рейсов этого поезда. Простым примером функции сравнения схем составов будет функция, сравнивающая числа вагонов каждого типа в составах. Точностью будет неотрицательное целое число. Если разность количества вагонов какого-то типа у сравниваемых схем составов по абсолютному значению меньше или равна точности, то составы считаются совместимыми, в противном случае они не совместны.

При создании модели предполагалось, что в самой модели заданы только основные ограничения на переключения между работами. Фактическое количество ограничений в реальной задаче достаточно велико, а сами ограничения разнообразны и часто не формализуемы. Пользователь модели знает об этих ограничениях. Также на практике возможны ситуации, когда переключение между работами запрещается основными ограничениями модели, но пользователь все равно считает его допустимым. Поэтому он может задавать для работы множество ее допустимых переключений. Это множество подается на вход задачи как множество пар поездов. Если для первого поезда в переключении задано вручную множество допустимых переключений, и пара поездов переключения принадлежит этому множеству, то переключение является допустимым. Если же не принадлежит, то переключение является недопустимым. Этот процесс можно представить как добавление и удаление ребер пользователем в графе допустимых переключений.

3.6. Критерии модели

Опишем критерии, по которым оценивается эффективность решения задачи. Основным критерием является минимизация числа исполнителей. Поезда, поданные на вход задачи, имеют одинаковые число дней и количество рейсов в периоде. Обороты выполняются циклично, количество используемых составов-исполнителей зависит от количества рейсов в периоде у поездов и от суммарного времени выполнения оборотов. Поскольку количество рейсов фиксировано, то минимизировать количество исполнителей можно сократив время выполнения оборотов. При этом время выполнения работ также фиксировано, меняться может только время переключения

между работами. Потому минимизация числа исполнителей является *минимизацией суммарных времен переключения между работами*. Соответственно, это и будем считать основным критерием в дальнейшем.

Другим критерием является *критерий консервативности решения*. Изменение существующей увязки поездов в обороты – дело непростое, при этом требуется провести ряд административных мероприятий. Решение, которое нашли, исходя только из предыдущего критерия, и которое мало отличается суммарным временем переключений от начального решения, не экономит исполнителей. Желательно, чтобы оно не менялось в случае незначительного улучшения суммарного времени переключения между работами при внесении малых изменений в параметры модели.

Поэтому вводится второй критерий – отклонение от начального решения. Всякий раз, при выборе переключения для заданной работы A , если вторая работа B' в переключении отличается от второй работы B , на которую A переключалась в исходном решении, то мы говорим, что переключение с A на B' отклоняется от начального решения. К стоимости переключений, отклоняющихся от начального решения, добавляется штраф – задаваемая в множестве управляющих параметров константа. Будем называть ее *штрафом за отклонение от начального решения*.

Третий критерий связан с устойчивостью найденных решений к задержкам поездов. Этот критерий задается специальной функцией, выставляющей оценку времени переключения. Эту функцию называют *функцией рейтинга времени переключения между работами*, а саму оценку – *рейтингом времени переключения*:

$$rate : t \in T \rightarrow E(rate) \quad (4)$$

Здесь $E(rate) \subseteq [0, +\infty)$ – область значений функции рейтинга, T – множество допустимых значений времени переключения, $T \subseteq [T_{min}, +\infty)$, T_{min} – минимальное время переключения.

Функция рейтинга – кусочно-линейная, выпуклая, настраиваемая пользователем функция. Она задает рейтинг времени переключения и подается на вход задачи во множестве управляющих параметров. Функция рейтинга обычно задана так, чтобы выставлять плохие оценки слишком маленьким переключениям.

В случае, когда разрешены засылочные поезда, вводится еще один критерий – *штраф за засылочный поезд*. В случае несовпадения соответствующих

концов у работ, переключение между ними все равно может оказаться допустимым, если засылочный поезд между концами работ все разрешен. К стоимости переключения добавляется заданный во множестве управляющих параметров штраф-константа. Кроме того, из-за использования засылочного поезда время переключения сильно возрастет, следовательно, возрастет и вклад в стоимость переключения основного критерия.

Если работы различаются по требованиям к схеме составов, то используется также *штраф за несовпадение схем составов*. Данный критерий применяется только для переключений, совместимых с точки зрения ограничения на схему составов. Суть критерия состоит в том, что если схемы составов работ переключения не являются полностью одинаковыми, то к стоимости переключения добавляется штраф за несовпадение схем составов. Штраф является, функцией, ставящей в соответствие паре схем составов неотрицательное число. Эта функция задается пользователем модели и подается на вход во множестве управляющих параметров. Критерий позволяет выбирать среди переключений с совместимыми схемами составов работ наилучшие.

Каждый критерий имеет свой вес - величину его вклада в итоговую стоимость переключения. Вес критерия задается как масштабирующий множитель, на который умножается вклад критерия. Веса находятся во входных данных, во множестве управляющих параметров задачи. По заданным критериям для вычисления стоимости переключения между работами формируется *функция оценки стоимости переключения*, как сумма критериев:

$$C = \sum_i a_i * c_i \quad (5)$$

Здесь C – стоимость переключения, a_i - вес i -го критерия, c_i - вклад i -го критерия. Целевой функцией задачи является суммарная стоимость переключений в решении.

3.7. Сведение к задаче о назначениях

Дадим сначала формальную постановку *задачи о назначениях* ([7] и [8]):

Даны два множества A и T одного размера. Задана функция стоимости $C: A \times T \rightarrow \mathbf{R}$ как квадратная матрица C , состоящая из вещественных чисел.

Необходимо найти минимум целевой функции:

$$\sum_{i \in A} \sum_{j \in T} c(i,j)x_{ij} \quad (6)$$

с ограничениями:

$$\sum_{j \in T} x_{ij} = 1 \text{ для } i \in A \quad (7)$$

$$\sum_{i \in A} x_{ij} = 1 \text{ для } j \in T \quad (8)$$

$$x_{ij} \geq 0 \text{ для } i, j \in A, T \quad (9)$$

Переменная x_{ij} может принимать значения на отрезке $[0, 1]$. Однако, для задачи о назначениях всегда существует оптимальное решение с целыми значениями ([7] и [8]). В нем x_{ij} представляет назначение исполнителя i на работу j , принимая значение 1, если исполнитель назначен на эту работу, и 0 - в противном случае. Первое ограничение требует, чтобы каждому исполнителю была назначена в точности одна задача, второе требует, чтобы для каждой задачи был назначен один исполнитель. Также можно заметить, что задача о назначениях эквивалентна задаче поиска минимального совершенного паросочетания, применяемой для поиска решения в моделях RSR и RSR-E.

Покажем теперь, что задача поиска покрытия вершин контурами с минимальным суммарным весом сводится к задаче о назначениях. Заметим, что для каждой работы в задаче требуется назначить ровно одного исполнителя, и, после выполнения работы, освобождается один исполнитель.

Преобразуем граф допустимых переключений в двудольный граф. Заменяем каждую вершину-работу на две вершины. Одна вершина представляет источник исполнителей, другая – потребитель исполнителей. Ребра, входившие в исходную вершину, теперь входят в вершину-потребителя, а ребра, исходившие из исходной вершины, теперь выходят из вершины-источника. Если у вершины была петля, то между источником и потребителем также есть ребро. Веса ребер нового графа равны весам соответствующих ребер исходного графа. Мы получили двудольный граф, в одной доле находятся источники, в другой – потребители. Доли одинакового размера. Назовем полученный граф *двудольным графом допустимых переключений*. На этом графе будем решать задачу о назначениях. Пример преобразования на рисунке 6.

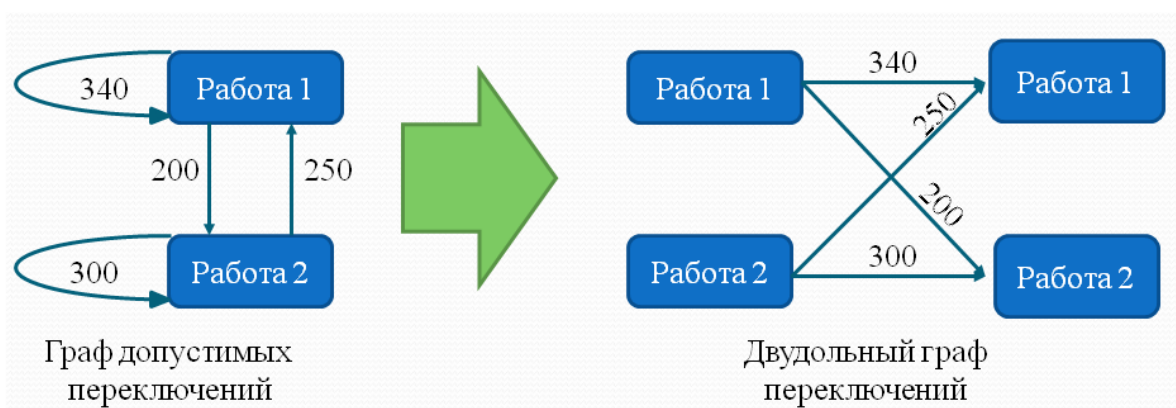


Рисунок 6. Преобразование графа допустимых переключений в двудольный граф для задачи о назначениях.

После решения задачи о назначениях мы получим либо совершенное паросочетание минимального веса для этого графа, либо результат, что решения нет. Ребрам в результате соответствуют ребра в исходном графе допустимых переключений, использовавшемся в модели в терминах теории расписаний. Переходя обратно к нашей модели, совмещаем вершины в одну. В качестве переключений между работами выбираем ребра, входящие в найденное паросочетание. В итоге, каждая работа имеет ровно одно переключение на другую работу, задано разбиение на контуры в исходном графе. Результатом может быть как один контур, так и несколько. Так как паросочетание, полученное при решении задачи о назначениях, имеет минимальную стоимость, то и суммарная стоимость полученных контуров-оборотов минимальная.

Время сведения полиномиально. Для задачи о назначениях существует полиномиальное решение, например, Венгерский алгоритм ([7], [8]). Поэтому и для задачи построения графика оборота также существует полиномиальное решение.

3.8. Модификация фиксированной вершиной до транспортной задачи

Выше было показано, что наша модель может быть сведена к задаче о назначениях. Представленный алгоритм сведения позволяет находить решение для задачи составления графика оборота. К сожалению, в случае отсутствия у задачи решения, мы не сможем получить никакой дополнительной информации, а именно какие поезда удалось увязать в обороты, а какие нет. Далее мы модифицируем

двудольный граф допустимых переключений и переходим к более общей транспортной задаче в сетевой постановке, частным случаем которой является задача о назначениях. Данная модификация позволит нам получить *частичное решение* для задачи составления графика оборота - множество оборотов, которые удалось составить, несмотря на отсутствие у задачи общего решения.

Сформулируем транспортную задачу в сетевой постановке ([7], [12]). Имеется взвешенный ориентированный граф $G = (V, E)$, каждой вершине $v \in V$ которого сопоставлено некоторое число b_i , называемое *интенсивностью вершины*. Граф G , вершинам которого сопоставлены значения интенсивностей b_i , будем называть *сетью*. Если $b_i > 0$, то вершина i называется *источником*, если $b_i < 0$, то — *потребителем*, а если $b_i = 0$, то — *нейтральной* вершиной. Также величину b_i в случае, когда она больше 0, называют *производством*, а когда меньше — *потреблением*.

Для определенной выше сети *поток* называется такая совокупность заданных на множестве дуг величин $X = \{x_d\}_{d \in E}$, что

$$\sum_{d \in E_i^+} x_d - \sum_{d \in E_i^-} x_d = b_i, i \in V \quad (10)$$

$$x_d \geq 0, d \in E \quad (11)$$

где E_i^+ — множество дуг, исходящих из вершины i , а E_i^- — множество дуг, входящих в нее. Величина x_d называется *значением потока* по дуге d и содержательно интерпретируется как количество продукта, пропускаемого по данной дуге.

Соотношение (10) означает, что для любой вершины сети разность выходящего и входящего потоков равна ее интенсивности.

. Для каждой дуги $d \in E$ определим значения $c_d \geq 0$, называемые *стоимостью перемещения единицы продукта* по дуге, тогда суммарная стоимость потока X примет вид

$$f(X) = \sum_{d \in E} c_d x_d \quad (12)$$

Задачу минимизации функции (12) при ограничениях (10) и (11) называют *транспортной задачей в сетевой постановке*.

Перейдем от задачи о назначениях к транспортной задаче. В двудольном графе допустимых переключений, как уже упоминалось ранее, первая доля состоит из вершин-источников исполнителей, а вторая доля — из вершин-потребителей исполнителей. Вершина источник символизирует освобождение исполнителя после выполнения работы, вершина потребитель представляет потребление исполнителя работой. Каждый источник освобождает ровно одного исполнителя, каждый

потребитель также использует ровно одного исполнителя. Припишем каждому источнику производство 1, а каждому потребителю потребление -1. Отметим, что сумма всех потреблений и производств равна нулю, такие транспортные задачи называют *сбалансированными* или *закрытыми*.

Добавим к двудольному графу допустимых переключений еще одну вершину с нулевой интенсивностью, которую назовем *фиктивной*. Проведем к фиктивной вершине ребро от каждого источника и добавим к каждому потребителю ребро от фиктивной вершины. Ребро, инцидентное фиктивной вершине, называем *фиктивным*. Построенный граф с фиктивной вершиной назовем *графом допустимых переключений с фиктивной вершиной*.

Фиктивная вершина и ребра позволяют транспортной задаче всегда иметь решение. За проход потока по фиктивному ребру назначим большой штраф, такой, чтобы поток пропускался через фиктивное ребро тогда и только тогда, когда у исходной задачи о назначениях нет решения. Для того чтобы при решении фиктивное ребро всегда выбиралось в последнюю очередь, вес фиктивного ребра должен быть больше, чем вес максимального из обычных ребер в построенном графе

$$c_{\text{фикт}} > \max_{d \in E} c_d \quad (13)$$

На вход транспортной задачи подается граф допустимых переключений с фиктивной вершиной, на выход выдает список ребер этого графа, входящих в оптимальное решение. Для решения задачи мы применяем метод потенциалов, а начальный план строится методом северо-западного угла ([12], [13]).

У транспортной задачи для графа с фиктивной вершиной решение есть всегда. Если у исходной задачи нет решения, то мы сразу можем определить это по вхождению в результат фиктивных ребер. В случае, когда у задачи о назначениях есть решение, решение транспортной задачи совпадает с ним.

Мы можем выполнить преобразование результата в контуры графа допустимых переключений. Для этого склеим каждую пару вершин, полученных при построении двудольного графа допустимых переключений, в одну вершину. Ребро между вершинами пары превращается в петлю. Фиктивная вершина остается в графе. По ребрам полученного графа, соответствующим ребрам в результате решения транспортной задачи, строим контуры в графе. Если решение содержит фиктивные ребра, то в нем будут контуры, проходящие через фиктивную вершину. Их мы

отбрасываем. Оставшиеся контуры, не проходящие через фиктивную вершину, будут частичным решением задачи.

3.9. Алгоритм решения задачи

Перечислим основные шаги, выполняемые при решении задачи построения графика оборота:

1. Преобразуем данные о поездах, поданные на вход задачи, к представлению в терминах теории расписаний.
2. Построение множества переключений поездов в начальном решении.
3. Строим функцию стоимости переключения для модели, используя множество управляющих параметров из входных данных.
4. Фактическое построение модели. Генерируем двудольный граф допустимых переключений для модели.
5. Модифицируем сгенерированный граф добавлением к нему фиктивной вершины.
6. Решаем для построенного графа транспортную задачу методом потенциалов. Преобразуем решение транспортной задачи в список оборотов поездов.

3.10. Построение множества обобщенных работ и начального решения задачи

На первом этапе решения мы создаем множество обобщенных работ для модели по поданным на вход поездам и станциям увязки. Сначала группируем поезда по их текущим оборотам, затем эти группы преобразуются в обобщенные работы. Далее рассматриваем такую группу поездов и ее текущий оборот. Разбиваем этот оборот на наборы поездов, начинающиеся и заканчивающиеся на станциях увязки. Для каждого из полученных при разбиении наборов создаем обобщенную работу, атомарные работы которой создаются по поездам набора.

Если множество станций увязки не пересекается с множеством станций-концов поездов группы, то мы отбрасываем эти поезда, т.к. менять план работ для составов мы разрешаем только для переключения на станциях увязки. Также отбрасываются поезда, у которых нет текущего оборота.

Типичной ситуацией является группа, состоящая из пары поездов. Если обе станции-конца пары поездов являются станциями увязки, то оборот разбивается на два набора по одному поезду, и из группы получаются две атомарные работы, по одной на каждый поезд. Если же только один из концов пары является станцией увязки, то из группы создается одна обобщенная работа.

Следующим шагом будет построение множества переключений поездов из начального решения задачи. Это множество состоит из пар поездов (tr_1, tr_2) , таких, что в начальном решении состав после выполнения рейса поезда tr_1 приступает к выполнению рейса поезда tr_2 . Для получения этого множества

- 1) Удаляем из начального решения все обороты, отброшенные на первом этапе;
- 2) Для всех оставшихся оборотов начального решения добавляем к множеству все переключения между поездами в оборотах. Если оборот состоит из поездов $(tr_1, tr_2, \dots, tr_n)$, то к множеству переключений добавляются пары $(tr_1, tr_2), (tr_2, tr_3), \dots, (tr_n, tr_1)$.

На практике множество переключений поездов из начального решения удобно представить в виде хеш-таблицы, записями которой будут переключения между поездами.

3.11. Функция оценки стоимости переключения

Третьим шагом будет построение функции оценки стоимости переключений. Согласно (5) эта функция является суммой вкладов различных критериев модели умноженных на их вес. Функцию оценки стоимости принимает на вход две работы, задающие переключение между работами, и возвращает стоимость этого переключения. Отметим, что стоимость переключения имеет смысл только в том случае, если переключение является допустимым. Расширим функцию оценки стоимости так, чтобы для допустимого ограничения она возвращала его стоимость, а для недопустимого – специальное значение, например, бесконечность.

$$C(w_1, w_2) = \begin{cases} \sum_i a_i * c_i, (w_1, w_2) - \text{допустимое переключение} \\ +\infty, (w_1, w_2) - \text{недопустимое переключение} \end{cases} \quad (14)$$

Веса критериев получаем из входных данных. Так как критерии независимы, то каждый критерий можно рассматривать по отдельности.

1) Критерий минимизации числа исполнителей – как уже было показано, минимизация числа исполнителей связана с минимизацией суммарного времени переключения между работами в решении. Критерий возвращает для двух работ время переключения между ними. Время переключения считается с учетом периодичности работ. Сначала вычисляем длину временного интервала между временем отправления второй работы переключения и временем прибытия первой работы переключения. Заметим, что она может быть меньше минимального допустимого времени переключения или вообще отрицательной.

Далее возьмем периодичность конца первой работы переключения и периодичность начала второй работы переключения. Путем анализа этих битовых строк найдем число дней k между датой прибытия на станцию переключения первого рейса первого поезда переключения и датой отправления со станции первого рейса второго поезда переключения. Для этого ищем первую единицу в каждой из битовых строк, а затем находим расстояние между ними.

$$k = \begin{cases} j - i, & j \geq i \\ n + j - i, & j < i < n \end{cases} \quad (15)$$

Здесь i – индекс первой единицы в периодичности конца первой работы переключения, j – индекс первой единицы в периодичности начала второй работы переключения, n – количество дней в периоде хождения поезда, т.е. длина битовых строк.

Теперь добавляем k дней к найденной разности между временами отправления и прибытия работ переключения. В случае, когда $k=0$, но разность между временами отправления и прибытия работ переключения меньше минимального времени переключения, добавляем к разности n дней. Возвращаем длину итогового временного интервала.

2) Критерий консервативности вычисляется следующим образом. На вход подаются две работы, требуется определить, присутствует ли переключение между ними в начальном решении. Для этого используем составленное ранее множество переключений поездов из начального решения. Возьмем поезда переключения – поезд, соответствующий последней атомарной работе первой работы переключения, и поезд, соответствующий первой атомарной работе второй работы переключения. Они задают пару поездов (tr_1, tr_2) . Если эта пара не принадлежит множеству поездов из начального решения, то переключение

между работами отклоняется от начального решения, и критерий возвращает штраф за отклонение. В противном случае критерий возвращает ноль.

- 3) Штраф за несовпадение станций переключения принимает на вход две работы и проверяет, совпадают ли у них станции переключения. Если да, то критерий возвращает ноль. Если нет, то есть два случая. Либо засылочные поезда запрещены, и критерий возвращает $+\infty$, либо они разрешены, тогда критерий возвращает стоимость введения засылочного поезда.
- 4) Критерий рейтинга принимает на вход время переключения между двумя работами, вычисленное по алгоритму, описанному в критерии минимизации числа исполнителей. Это время переключения подается на вход функции рейтинга, получаемой из входных данных. На выход критерий выдает оценку переключения, выданную функцией рейтинга.
- 5) Штраф за несовпадение схем составов. Этот критерий принимает на вход две работы и сравнивает схему конца первой работы и схему начала второй при помощи функции сравнения схем составов с заданной точностью. Функция сравнивает попарно количество вагонов разных типов в схемах и, если оно отличается больше, чем позволяет заданная точность, возвращает $+\infty$. В противном случае, для схем вызывается функция вычисления штрафа за различие в схемах составов, которая содержится во входных данных задачи. Критерий возвращает вычисленный штраф.

В случае, если переключение не является допустимым с точки зрения ограничений задачи, но пользователь указал его как допустимое, функция оценки стоимости переключения игнорирует ограничения задачи и возвращает вычисленную для переключения стоимость, словно ограничение является допустимым.

3.11. Генерация двудольного графа допустимых переключений и добавление фиктивной вершины

Теперь, вычислив для модели набор работ и функцию оценки веса переключения между работами, мы можем построить графа допустимых переключений модели. Чтобы избежать лишних действий и связанных с ними накладных расходов, будем сразу строить двудольный граф допустимых переключений с фиктивной вершиной для транспортной задачи:

- 1) Для каждой работы из множества работ модели добавляем к графу две вершины – источник и потребитель. Присваиваем вершине-источнику производство 1, а вершине потребителю производство -1. После добавления новой пары вершин переходим к шагу 2.
- 2) Для каждой из уже добавленных к графу вершин-источников:
 - а. Проверяем, задано ли пользователем для работы, которой соответствует вершина-источник, множество допустимых переключений. Если нет, то переходим к шагу 2б. В противном случае анализируем переключение между работой, соответствующей вершине-источнику, и работой, соответствующе новому потребителю. Если пара поездов переключения находится во множестве допустимых переключений для работы, которой соответствует вершина источник, переходим к шагу 2б. В противном случае берем следующую вершину источник.
 - б. Вычисляем стоимость переключения между работой, соответствующей вершине-источнику и работой, соответствующей новой вершине-потребителю. Если стоимость переключения не равна $+\infty$, то проводим ребро между соответствующими вершинами. Вес ребра равен стоимости переключения. В противном случае возвращаемся на шаг 2 и переходим к следующей вершине-источнику.
- 3) Для каждой из уже добавленных к графу вершин-потребителей, за исключением новой вершины потребителя (мы уже учли ее на шаге 2) проводим аналогичные шагу 2 действия для построения ребер между новым источником и вершиной-потребителем.

- 4) Переходим на шаг 1 алгоритма и добавляем пару вершин для следующей работы из множества обобщенных работ модели.

Алгоритм строит двудольный граф допустимых переключений. К построенному графу добавляем фиктивную вершину. Интенсивность фиктивной вершины равна 0. Проводим фиктивные ребра от всех источников к фиктивной вершине и от фиктивной вершины ко всем стокам. Вес фиктивных ребер одинаков и равен некоторой константе, для которой выполнено (13).

3.12. Решение транспортной задачи и интерпретация результатов.

На вход транспортной задачи подается граф допустимых переключений с фиктивной вершиной, на выход выдается список ребер этого графа, по которым течет поток. Для решения задачи мы применяем метод потенциалов, а начальный план строится методом северо-западного угла ([12], [13]). Хотя этот алгоритм и не является полиномиальным в общем случае, он работает для большинства наборов данных не хуже полиномиального.

У транспортной задачи для графа с фиктивной вершиной решение есть всегда. Если у исходной задачи нет решения, то мы сразу можем определить это по вхождению в результат фиктивных ребер. Опишем алгоритм интерпретации результатов.

- 1) Результат решения транспортной задачи – список ребер.
- 2) Ищем в результирующем списке ребер контуры. Выбираем произвольное ребро в списке. Запоминаем его вершину-начало. Создаем новый контур и добавляем это ребро к нему.
- 3) Ищем в списке ребро, начало которого является концом ребра, добавленного к контуру на предыдущем шаге. Добавляем его к контуру.
- 4) Повторяем шаг 3, пока не добавим к контуру ребро, концом которого будет вершина начало контура.
- 5) Добавляем контур к результатам, удаляем ребра контура из списка.
- 6) Если список ребер не пустой, выполняем шаг 1. В противном случае завершаем работу и возвращаем множество найденных контуров.

Если задача построения графика оборота не имеет решения, то некоторые контуры будут содержать фиктивные ребра. Удалив из результатов такие контуры, мы

получим частичное решение задачи. В дальнейшем контуры преобразуются в обороты поездов и подаются на выход в качестве итогового решения задачи построения графика оборота.

При построении модели изначально за основу была взята модель RSR-E. Также были использованы идеи из модели [3]. Модель была модифицирована для задачи поиска графика оборота, удовлетворяющих технологическим требованиям РЖД. К модели был добавлен ряд критериев дополнительных критериев, позволяющих более точно оценивать эффективность найденных решений. Были реализованы алгоритмы построения модели, сведения модели к транспортной задаче и решения транспортной задачи. Расширено и адаптировано для модели понятие периодичности хождения поездов. На языке C# написан программный комплекс для работы с моделью.

4. Описание практической части

Описанная в работе модель задачи построения графика оборота составов и алгоритм ее решения были реализованы на языке C# при помощи платформы .Net в рамках АСУ “Компас”, предназначенной для работы с железнодорожными пассажирскими перевозками. Часть реализованной системы, связанная с решением транспортной задачи, написана на языке C++.

4.1. Используемый инструментарий

Язык C# является одним из наиболее распространенных современных языков, сочетающий в себе удобство работы с управляемой памятью и возможности как императивных, так и функциональных языков программирования. Он является одним из основных языков для разработки приложений для ОС Windows. Язык C# был выбран, исходя из этих соображений, а также потому, что АСУ “Компас” написана преимущественно на этом языке программирования.

Для реализации транспортной задачи был выбран язык C++, т.к. данный язык программирования признан как один из языков, эффективных для реализации ресурсоемких задач.

Платформа .Net является универсальной платформой для написания приложений для операционной системы Windows. Основой платформы является общезыковая среда исполнения, которая подходит для разных языков программирования. Функциональные возможности этой среды доступны в любых языках программирования, ее использующих. Она поддерживает ряд популярных языков программирования, таких как C#, F# и Visual Basic. Также платформа содержит множество библиотек, облегчающих разработку различных аспектов приложений, таких как доступ к базам данных, работа с веб-службами, асинхронное программирование и создание пользовательских интерфейсов. Все эти области были затронуты при разработке данной системы. Доступ к данным, хранимым в базе данных, использующей Microsoft SQL Server, организован при помощи библиотеки ADO.NET. В основу АСУ “Компас” положена архитектура типа “клиент-сервер”, данные о поездах хранятся удаленно, доступ к ним организован при помощи фреймворка WCF. Для ускорения работы системы некоторые операции, такие как получение данных, выполнены асинхронно при помощи библиотеки TPL. Пользовательский интерфейс

АСУ написан с помощью фреймворка WPF. Все перечисленные библиотеки являются частью платформы .Net.

4.2. Общая схема работы и структура системы

Весь реализованный программный комплекс состоит из четырех частей. Первая часть – это расчетный модуль, написанный на C++ и решающий транспортную задачу. Второй модуль – независимый АСУ “Компас” компонент, получающий на вход множество работ, начальное решение, множество пар поездов, задающих допустимые переключения, и функцию оценки стоимости переключения. Он строит двудольный граф и передает его в расчетный модуль, а после интерпретирует результат решения. Возвращает множество оборотов.

Третий компонент – посредник между вторым и АСУ “Компас”. Он преобразует данные из АСУ “Компас” в данные для второго компонента, а также конструирует функцию оценки стоимости переключения.

Четвертый компонент тесно интегрирован в АСУ “Компас”. Это пользовательский интерфейс, в котором пользователь производит отбор поездов для задачи и задание множества управляющих параметров.

Сначала система получает данные о поездах, станциях увязки, начальном решении и остальных входных данных из АСУ “Компас”. Затем эти данные передаются посреднику между АСУ и независимым компонентом от АСУ компонентом, решающим задачу. Посредник служит мостом между ними и знает о внутреннем устройстве обоих. По данным из АСУ он создает список обобщенных работ и функцию оценки стоимости переключения, Эти данные затем передаются в компонент для построения графика оборота, которая по ним генерирует граф модели, вызывает для этого графа расчетный модуль, решающий транспортную задачу, и интерпретирует полученное решение. Посреднику возвращается список оборотов, представленных как наборы обобщенных работ, которые он преобразует обратно в поезда. Общую схему архитектуры системы можно увидеть на рисунке 7. Архитектура системы представлена в приложении С. На рисунке С.1. приложения изображена упрощенная схема части приложения, связанной с АСУ “Компас”, на рисунке С.2. изображена схема остальных трех частей системы.



Рисунок 7. Схема общего устройства системы.

4.3. Результаты применения системы

Система была протестирована на нескольких наборах поездов из РЖД. Каждый набор соответствовал технической станции, на которой его поезда обслуживаются. Для поездов этих наборов в РЖД было вручную составлено несколько сложных оборотов.

Было выполнено сравнение, показывающее эффективность решений, найденных системой по сравнению с существующими решениями (таблица 1).

| Название технической станции | Сложные обороты | | Экономия |
|---|-------------------------------------|-------------------------------------|----------|
| | Существующие в РЖД | Рассчитанные в системе | |
| ЛВЧД-1 Москва | 1 оборот, экономит 1 состав | 2 оборота, каждый экономит 1 состав | 1 состав |
| ЛВЧД-8 Санкт-Петербург пасс. Московский | 2 оборота, каждый экономит 1 состав | 2 оборота, экономия 1 и 2 состава | 1 состав |
| ЛВЧ-1 Ярославль | 2 оборота, каждый экономит 1 состав | 3 оборота, каждый экономит 1 состав | 1 состав |

Таблица 1. Результаты сравнения существующих графиков оборота с графиками, рассчитанными системой.

Более подробно распишем вторую строчку таблицы 1. Поезда, обслуживаемые технической станцией ЛВЧД-8 Санкт-Петербург пассажирский Московский, имели график оборота, состоящий почти полностью из простых оборотов. Единственный сложный оборот состоял из двух пар поездов – поезда №49/№50 и поезда №55/56. Этот оборот сэкономил 1 состав. С помощью системы мы нашли сложный оборот из трех пар поездов – поезда №55/№56, поезда №227/№228 и поезда №247/248, что дает экономию два состава, на один состав больше, чем в существующем решении.

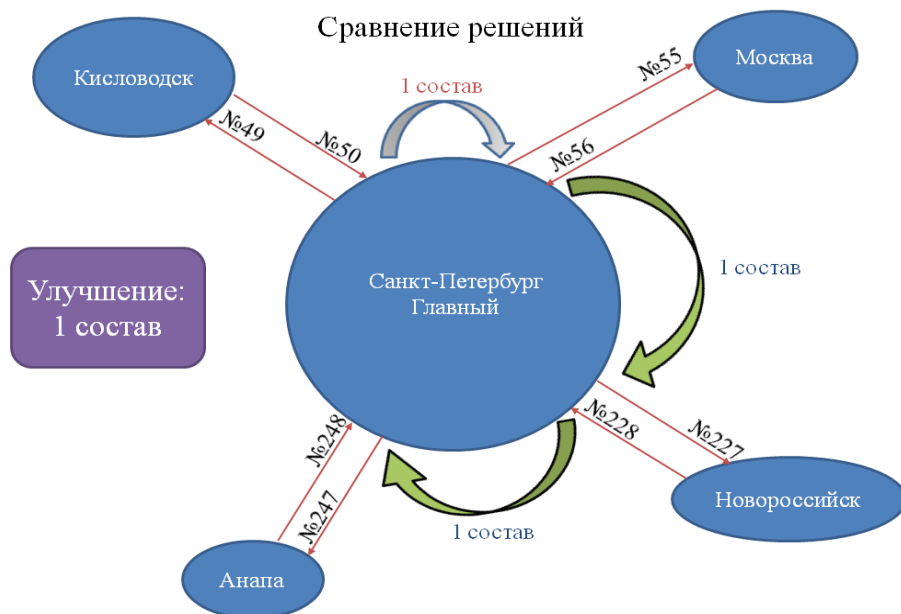


Рисунок 7. Сравнение существующего графика оборота с решением, рассчитанным системой для поездов технической станции ЛВЧД-8 Санкт-Петербург Пассажирский Московский.

5. Заключение

В рамках данной работы были получены следующие результаты:

- 1) Исследованы существующие модели для задачи построения оборотов поездов. Написан обзор, подробно рассматривающий семейство моделей RSR, затронуты также некоторые более сложные модели.
- 2) Построена модель задачи построения плана работ составов, поддерживающая технологические требования РЖД. Для нее было расширено понятие периодичности хождения поездов. К модели добавлены критерии консервативности найденного решения и устойчивости его к задержкам составов, а также гибкая система штрафов и ограничений. Показана сводимость предложенной модели к классической задаче о назначениях. Предложен метод поиска оптимального относительно описанных критериев плана работ составов, а также способ получения частичного плана работ в случае, когда общего не существует.
- 3) Создана программная система для использования модели специалистами РЖД при составлении оборотов дальних пассажирских поездов. Система интегрирована в АСУ “Компас”, но является независимой и поддерживает возможность ее использования в других АСУ.

Список литературы

1. M. Mergner. Rolling Stock Rostering. Seminar on Algorithms and Models for Railway Optimization. University of Constance. SS 2002. [PDF] (<http://algo.uni-konstanz.de/lehre/ss02/rails/mergner.pdf>).
2. T. Erlebach, M. Gantenbein, D. Hürlimann, G. Neyer, A. Pagourtzis, P. Penna, K. Schlude, K. Steinhöfel, D. S. Taylor, P. Widmayer. On the complexity of train assignment problems. In: Eades, P., Takaoka, T. (eds.) ISAAC 2001. LNCS, vol. 2223, pp. 390-402. Springer, Heidelberg (2001).
3. G. L. Giacco, A. D'Ariano, D. Pacciarelli. Rolling stock rostering optimization under maintenance constraints [PDF] (<https://www.mech.kuleuven.be/MT-ITS2011/downloads/Abstracts/079,%20G.%20Giacco%20et%20al.,%20Rolling%20Stock%20Rostering%20Optimization%20under%20Maintenance%20Constraints.pdf>)
4. Основы организации пассажирских перевозок [HTML] (<http://www.tehnoinforma.ru/zheleznajadoroga/67.html>)
5. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы. Построение и анализ. 3-е издание, Вильямс, 2013, с. 811-837.
6. Н. Н. Кузюрин, С. А. Фомин. Эффективные алгоритмы и сложность вычислений, с.63-71, 309-320. [PDF] (http://discopal.ispras.ru/img_auth.php/f/f4/Book-advanced-algorithms.pdf)
7. Н.Кристофидес. Теория графов. Алгоритмический подход, М.: Мир, 1978, 432 с. с. 11-28, 268-411.
8. А. А. Лазарев, Е. Р. Гафаров. Теория расписаний. Задачи и алгоритмы, МГУ, Москва, 2011, с. 76-81 [PDF] (<http://physcontrol.phys.msu.ru/materials/PosobieLazarev/TeorRasp.pdf>)
9. Э. Таненбаум. Архитектура компьютера, 5-е изд. – СПб.: Питер, 2007, ISBN 5-469-01274-3, с. 87-88.
10. J. Miao, Y. Yu, Y. Wang, A heuristic algorithm for the circulation plan of railway electrical motor units // Computers in Railways 2010, vol. 114, pp.877-887 [PDF] (<http://www.witpress.com/Secure/elibrary/papers/CR10/CR10079FU1.pdf>)

11. L. Andereg, S. Eidenbenz, M. Gantenbein, C. Stamm, D.S. Taylor, B. Weber, P. Widmayer. Train Routing Algorithms: Concepts, Design Choices, and Practical Considerations // Proceedings 5th Workshop on Algorithm Engineering and Experiments (ALENEX03), 2003 [PDF] (<http://www.siam.org/meetings/alenex03/Abstracts/LAndereg3.pdf>)
12. Курицын А.Г. Линейное программирование. Транспортная задача (учебное пособие) [PDF] (<http://sa.technolog.edu.ru/files%5Ckuricin%5Ctr-met.pdf>).
13. Симаков Е. Е., Ким Е. Решение транспортных задач с применением программирования в системе MathCAD // Молодой ученый. 2014. №5. С. 8-13.
14. 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.
15. Аничкин А. С., Семенов В. А. Современные модели и методы теории расписаний // Труды ИСП РАН. 2014. №3 С.5-50 [HTML] (<http://cyberleninka.ru/article/n/sovremennye-modeli-i-metody-teorii-raspisaniy>)
16. M. Kuhn. A summary of the international standard date and time notation [HTML] (<https://www.cl.cam.ac.uk/~mgk25/iso-time.html>).
17. Служебное расписание движения пассажирских поездов с 29 мая 2016 года. Курский вокзал. М.: Трансжелдориздат, 2016 г.
18. Служебное расписание движения пассажирских поездов с 13 декабря 2015 года. Ярославский вокзал. М.: Трансжелдориздат, 2015 г.

Приложение А. Данные, связанные с описанием времени и временных интервалов.

При разработке компьютерных приложений данные, связанные с конкретным моментом времени, имеют общее название *DateTime*. Они обычно состоят из двух частей – *времени суток* и *даты*. Под датой понимается тройка натуральных чисел – год, номер месяца в году и номер дня в месяце. Время суток – набор неотрицательных целых чисел: число часов (от 0 до 23), число минут и число секунд (от 0 до 59), число миллисекунд (если требуется большая точность). Для таких объектов естественным образом определяются операции сравнения – посредством сравнения сначала дат, а потом времен суток. Если один объект типа *DateTime* меньше другого, то можно сказать, что он описывает более ранний момент времени. Подробнее о типе данных *DateTime* можно прочитать в [15].

Другой используемый тип данных для работы со временем связан с описанием временных интервалов и часто называется *TimeSpan*. Он описывает длину временного интервала и имеет точно такой же формат, что и время суток в данных типа *DateTime*, за исключением того, что кроме часов, минут, секунд и миллисекунд он также содержит знак и число дней во временном интервале. Также временной интервал можно задать его началом и концом (типа *DateTime*). Тогда разность между концом интервала и началом будет описываться объектом типа *TimeSpan*. В отличие от времени суток объект типа *TimeSpan* может быть отрицательным, например, в случае, если конец интервала меньше его начала. Более подробно о *TimeSpan* в [15]. Также подробная информация о числовом представлении данных, связанных с временем, содержится в стандарте ISO 8601.

Приложение Б. Графики движения и оборота поездов.

В данном приложении можно увидеть, как в РЖД принято изображать графики движения для какого-то участка железной дороги. Это делают с помощью *линейчатого графика*. На линейчатом графике отображаются поезда с маршрутами, проходящими через как минимум две подряд идущие станции участка дороги. По оси X отложено время, сама ось разбита на равные интервалы, каждый длиной в сутки. Также возможно проставление оси X отметок, разбивающих каждые суточные интервалы на часовые и десятиминутные подинтервалы, в зависимости от масштаба линейчатого графика. На оси Y отложены станции участка железной дороги. В каждом суточном интервале чертится рейс поезда в виде кусочно-линейной функции, задаваемой расписанием поезда. Такую функцию называют *ниткой поезда*. Если в какой-то день у поезда нет рейса, то нитка для этого дня чертится пунктирной линией.

Иногда на графике могут также строить *объезды* - отображение кусков маршрутов поездов, проходящих через две не подряд идущие вершины. Их чертят особой заранее оговоренной пунктирной линией. Пример линейчатого графика изображен на рисунке Б.1.

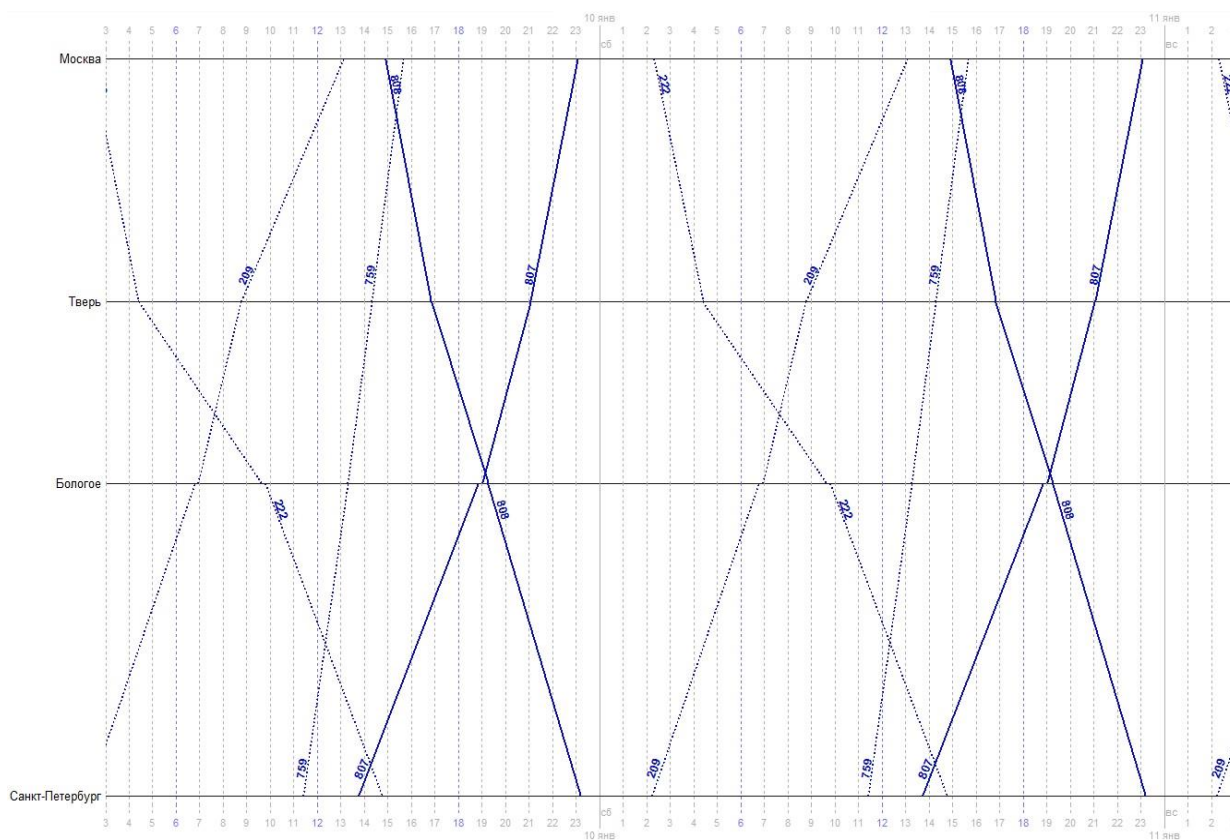


Рисунок Б.1. График движения для участка железной дороги между Москвой и Санкт-Петербургом.

Графически обороты могут отображаться несколькими способами. Первый способ – отображать обороты на линейчатом графике, описанном выше. Это так называемое *развертывание станции*, когда на линейчатом графике рисуют обороты для выбранной станции.

Развертывание бывает подробным и кратким. При *кратком* развертывании на линейчатом графике рисуются все железнодорожные пути, которые есть на станции, в виде горизонтальных прямых линий. На путях проставляются все поезда, проходящие через станцию. Состав отмечается на станционном железнодорожном пути, если идет по нему при проходе через станцию. Для станции составляется график занятия станционных путей. На нем сразу видно, сколько и где стоит состав, и рейс какого поезда он пойдет выполнять дальше. Пример краткого развертывания изображен на рисунке Б.2.

На рисунке подробно развернута станция Москва-Пассажирская-Курская. На станционных путях стоят отметки для соответствующих поездов, проходящих через станцию по этим путям. Стоянки составов на путях показаны зелеными прямоугольниками.

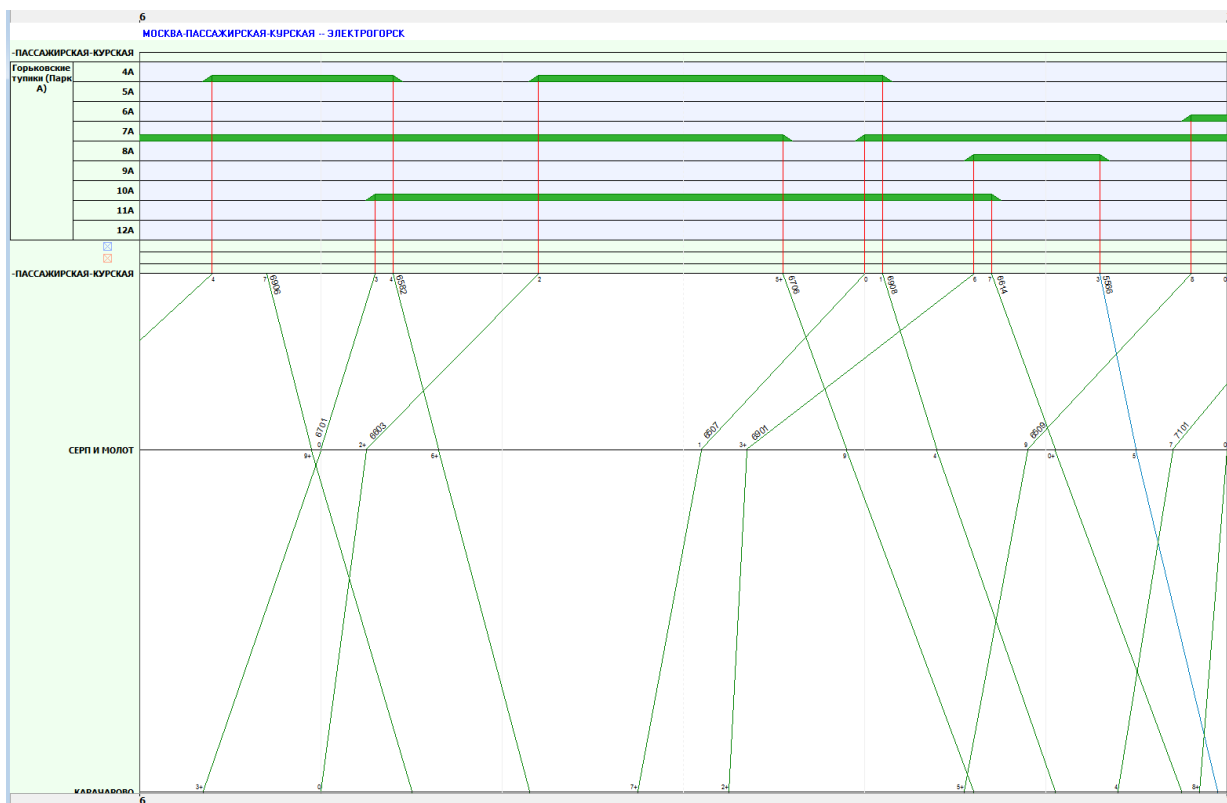
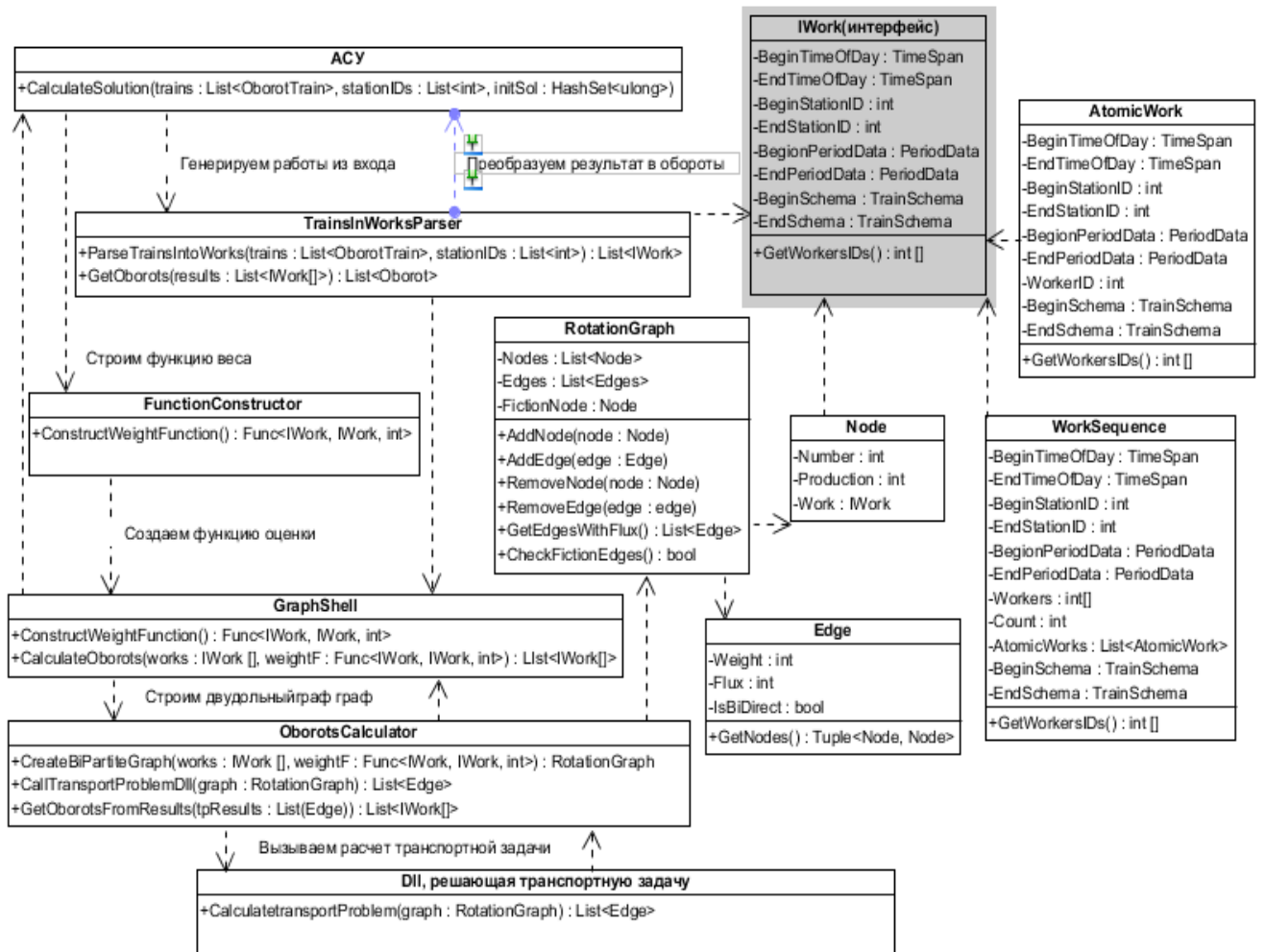


Рисунок Б.2. Развертывание станционных путей для станции Москва-Пассажирская-Курская.

отправляющиеся после трех часов ночи, переносятся на следующую строку, рейсы, проходящие три часа ночи, разбиваются на две части – часть до трех часов и часть больше трех. На последней станции дневного маршрута электричка переживает ночь, здесь находится ее *ночевка*.



С.2. Упрощенная схема архитектуры независимого компонента системы, связанного с решением задачи.