

На правах рукописи

Железов Роман Владимирович

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ ИНФОРМАЦИОННО-
СПРАВОЧНОЙ СИСТЕМЫ ПОИСКА ОПТИМАЛЬНЫХ
ПУТЕЙ ПРОЕЗДА НА ПАССАЖИРСКОМ ТРАНСПОРТЕ**

Специальность 05.12.13 – Системы, сети и устройства
телекоммуникаций.

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Москва – 2009

Работа выполнена на кафедре телекоммуникационных сетей и систем в Московском физико-техническом институте (государственном университете).

Научный руководитель: доктор технических наук, профессор
Вишневский Владимир Миронович

Официальные оппоненты: доктор технических наук, профессор,
академик РАН
Кузнецов Николай Александрович
Институт радиотехники и электроники РАН

кандидат технических наук
Березка Михаил Павлович,
Научно-исследовательский и проектно-
конструкторский институт информатизации,
автоматизации и связи на железнодорожном
транспорте (НИИАС)

Ведущая организация: Институт проблем управления
им. В.А.Трапезникова РАН

Защита состоится 24 марта 2009 г. в 15⁰⁰ на заседании диссертационного совета Д.212.156.04 при Московском физико-техническом институте (ГУ) по адресу:
141700, г. Долгопрудный, Московская обл.,
Институтский пер., д. 9, ауд. 204 Новый корпус.

С диссертацией можно ознакомиться в библиотеке
Московского физико-технического института (ГУ).

Автореферат разослан __ февраля 2009 г.

Ученый секретарь
диссертационного совета Д.212.156.04,
к.т.н., доцент

Л.П.Куклев

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Предоставление справочной информации об оптимальных путях проезда на пассажирском транспорте является необходимым условием для качественного обслуживания пассажиров. Полнота предоставленной информации не только помогает пассажиру, но и повышает эффективность пассажирских перевозок, уменьшает нагрузку на транспортные сети за счет оптимизации пассажиропотока.

Первые электронные справочные системы расписаний транспорта появились в 80-х годах прошлого века. На постсоветском пространстве функционируют системы «ЭКСПРЕСС» – на железнодорожном транспорте, «СИРЕНА» – на воздушном транспорте. В Европе широкое распространение получили системы «NAFAS» и «EFA».

К сожалению, уровень справочно-информационного обслуживания пассажиров далек от совершенства. В настоящее время в России даже в рамках отдельных видов транспорта отсутствует возможность поиска маршрута проезда с пересадкой. Исключение составляют автоматизированные информационные системы на воздушном транспорте, которые способны находить маршруты с пересадкой. Принимая во внимание размер транспортной системы России и то, что количество железнодорожных станций превосходит количество аэропортов в сотни раз, данная задача на железнодорожном транспорте является более сложной, чем на воздушном.

Еще сложнее - задача поиска интермодального пути (пути проезда с использованием нескольких видов транспорта), которая пока в полной мере не решается ни одной из существующих автоматизированных систем. При этом следует отметить, что транспортная сеть в Российской Федерации весьма неоднородна: есть населенные пункты, в которые можно попасть только по воздуху; имеются регионы, лишенные железнодорожного транспорта; есть населенные пункты, до которых можно добраться только автотранспортом. Поэтому создание информационно-справочной системы с использованием современных телекоммуникационных технологий, которая объединит информацию из действующих автоматизированных систем на различных видах транспорта с целью получения наиболее полной справочной информации о возможности проезда с учетом пересадок и наличия мест является актуальной проблемой.

Научные исследования по поиску оптимальных путей проезда на пассажирском транспорте проводили М. Müller-Hannemann, F. Schulz, D. Wagner, C. Zaroliagis, G. S. Brodal, R. Jacob, M. Schnee, K. Weihe, E. Pyrga, B. A. Жожикашвили, В.М. Вишнеvский, В. К. Попков и др. Большую работу по практическому исследованию алгоритмов поиска пути и сравнению их производительности провели D. Wagner и T. Willhalm. Однако недостатком известных работ является слабое внимание к требованиям, неизбежно возникающим при реализации единой информационно-справочной системы с использованием современных телекоммуникационных технологий. Так, в работах упомянутых авторов предлагается проводить длительные предварительные вычисления при обновлении данных, что накладывает ограничения на возможность актуализации информации в реальном времени.

Цель диссертации. Целью настоящей диссертационной работы является построение моделей и методов реализации информационно-справочной системы, объединяющей информацию о разных видах пассажирского транспорта, на базе эффективного алгоритма поиска пути проезда с учетом расписаний, возможных пересадок между видами транспорта и наличия свободных мест. Для достижения поставленных целей в работе решены следующие основные задачи:

1. Проведение сравнительного анализа существующих методов и алгоритмов поиска пути с учетом расписаний движения транспорта.
2. Разработка сетевой модели взаимодействия справочной системы с существующими информационными системами на транспорте.
3. Разработка эффективного алгоритма поиска пути проезда на пассажирском транспорте с учетом пересадок и наличия мест.
4. Разработка базы данных и программного комплекса для обслуживания справочной системы.
5. Исследование производительности предложенных алгоритмов и разработанной информационно-справочной системы.
6. Разработка интернет-портала для доступа к информационно-справочной системе.

Методы исследования. Для достижения поставленной цели в диссертационной работе используются методы теории вероятностей, теории массового обслуживания, теории графов, а также имитационное моделирование.

Научная новизна:

- разработан новый алгоритм поиска оптимального пути проезда на пассажирском транспорте с минимальным количеством пересадок и минимальным временем в пути с учетом наличия свободных мест;
- разработана модель интеграции существующих систем на транспорте в единую информационно-справочную систему;
- предложена модель сети, описывающая функционирование справочной системы с учетом топологии сети и пропускных способностей различных уровней сети;
- предложены эффективные методы ускорения алгоритма поиска пути, позволяющие использовать алгоритм на практике.

Практическая ценность и реализация результатов. Разработанные алгоритмы и методы явились основой для создания справочного интернет-портала <http://transport.marshrutv.ru> . Портал обеспечивает получение справок о возможности проезда с пересадками между двумя любыми пунктами Российской Федерации и ближнего зарубежья с использованием различных видов транспорта. Апробированная технология используется в разработке информационной системы ГУП МО «МОСТРАНСАВТО» и ЗАО «Транспортные Автоматизированные Информационные Системы», что подтверждено соответствующими актами о внедрении. Разработанные алгоритмы и программное обеспечение могут быть использованы и на других видах транспорта. Получено свидетельство о государственной регистрации программы для ЭВМ (№ 2008614887 от 10 октября 2008 года).

Апробация результатов работы. Основные результаты диссертации докладывались и обсуждались на:

- Международном семинаре «Распределенные компьютерные и телекоммуникационные сети. Теория и приложения» (DCCN-2007, Москва);
- Международном семинаре «Распределенные компьютерные и телекоммуникационные сети. Теория и приложения» (DCCN-2005, София, Болгария);

- Научных конференциях «Технологии Microsoft в теории и практике программирования» (2006, 2007, Москва);
- Семинарах ИППИ РАН им. А. А. Харкевича;
- Конференции молодых ученых и специалистов «Информационные технологии и системы» (ИТиС-2007, Звенигород)
- Научных конференциях МФТИ в 2004 и 2005г. (Долгопрудный).

Публикации. По теме диссертации опубликовано 9 научных работ, список которых приведен в конце автореферата, в том числе две статьи в рецензируемых научных журналах, утвержденных в перечне ВАК.

Структура и объем диссертационной работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы, включающего 88 наименований, и приложения. Работа изложена на 145 страницах и содержит 75 рисунков и 7 таблиц.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность диссертационной работы, ее научная новизна и практическая значимость результатов.

В **первой главе** делается обзор отечественных и зарубежных информационных систем на транспорте и существующих алгоритмов поиска путей на графе. Особое внимание уделяется специализированным алгоритмам поиска пути в пространстве. Описаны существующие подходы к поиску маршрута с учетом расписаний движения транспорта, рассматриваются сетевые модели и архитектуры кэширования, которые используются при проектировании справочной системы.

В параграфе 1.2 описана отечественная система резервирования железнодорожных билетов «ЭКСПРЕСС». Первая система электронного резервирования «ЭКСПРЕСС-1» начала работу в Москве в 1972 году. В 1982 году была внедрена новая система – «ЭКСПРЕСС-2». В последующие годы количество региональных центров «ЭКСПРЕСС-2» достигло 29, которые обслуживали большинство государств бывшего СССР. В январе 2002 года в Москве заработала система «ЭКСПРЕСС-3». ЭКСПРЕСС-3 – это комплексная система обслуживания пассажиров, которая включает в себя средства

децентрализованной подготовки и публикации расписаний, а также средства информационного обслуживания пользователей.

Первая автоматизированная система резервирования авиационных билетов «СИРЕНА» была введена в действие в 1972 году. Исчерпав свой технологический ресурс, она была заменена в 1981 году системой «СИРЕНА-2». Спустя несколько лет было разработано несколько альтернативных проектов, на смену которым пришла распределительная система «Сирена-Трэвел», введенная в промышленную эксплуатацию в 2001 году.

Помимо глобальных автоматизированных систем, охватывающих всю территорию Российской Федерации, существуют региональные справочные системы в отдельных городах и регионах. Обычно эти системы обслуживают автобусные компании, осуществляющие перевозки в отдельных регионах. Несмотря на невысокую сложность задачи поиска пути проезда в отдельном регионе, эти системы не имеют возможности поиска с пересадкой. На данный момент даже такая актуальная задача, как справка о проезде с пересадкой на автобусах и электропоездах Московской области, не решается ни одной из систем.

В параграфе 1.3 описываются о зарубежные информационные системы на пассажирском транспорте. В Европе наибольшее распространение получили две системы: «**NAFAS**» – на железнодорожном транспорте, «**EFA**» – на городском и пригородном транспорте. NAFAS - наиболее известная в Европе на сегодняшний день информационная справочная система на наземном транспорте для больших расстояний. Система используется государственными и частными железными дорогами в нескольких странах Европы. Несколько лет назад начала работать новая справочная система «**EU-Spirit**». Система позволяет находить пути проезда между интересующими точками в различных регионах Европы. Она основывается на существующих локальных, региональных и национальных справочных информационных системах, которые соединены посредством разработанных программных и телекоммуникационных интерфейсов. Известно, что результат справки для некоторых запросов в этих системах оказывается не самым оптимальным. Основной причиной такого поведения является то, что поисковый алгоритм справочной системы использует неточные методы, чтобы уменьшить пространство поиска.

В параграфе 1.5 дано описание эффективного алгоритма для поиска оптимальных путей в различных пространствах - **A*** (читается как "А-

звездочка"). Этот эвристический поиск сортирует все узлы по приближению наилучшего маршрута, идущего через этот узел. Типичная формула эвристики выражается в виде:

$$f(n) = g(n) + h(n) ,$$

где $f(n)$ - значение оценки, назначенное узлу n , $g(n)$ - наименьшая стоимость прибытия в узел n из точки старта, $h(n)$ - эвристическое приближение стоимости пути к цели от узла n . A^* гарантированно находит кратчайший путь, до тех пор, пока эвристическое приближение $h(n)$ является допустимым, то есть оно никогда не превышает действительного оставшегося расстояния до цели.

Существует два основных подхода к моделированию **информации о расписаниях** как задачи поиска кратчайшего пути: **время-расширенный** и **время-зависимый**. Эти подходы рассмотрены в параграфе 1.6. Общее свойство подходов в том, что результат вычисляется применением некоторого алгоритма поиска кратчайшего пути к специально построенному графу. Время-расширенный граф конструируется таким образом, что каждый узел соответствует определенному времени (прибытия или отправления) на станцию, а ребра между узлами представляют либо элементарные соединения между двумя событиями, либо время ожидания на станции. На практике это приводит к созданию графа большой размерности. Что касается время-зависимого подхода, то идея состоит в том, чтобы избежать создания узлов графа для каждого события. Вместо этого время-зависимый граф строится так, что каждый узел представляет станцию, и два узла считаются связанными ребром, если соответствующие станции связаны элементарным соединением. Вес ребра присваивается «на лету» и зависит от времени, когда конкретное ребро будет использовано алгоритмом поиска кратчайшего пути. В задаче **о расписаниях** используются следующие понятия: станции (либо остановки, порты и др.), маршруты (поезда, автобусы), курсирующие между станциями, времена отправления и прибытия маршрутов на станции, и дни курсирования. Задача поиска формулируется следующим образом: имеется набор маршрутов Z , набор станций B , и набор элементарных соединений C элементы которого c - кортеж пяти величин в форме $c = (Z, S_1, S_2, t_d, t_a)$. Каждое элементарное соединение понимается как маршрут Z , отправляющийся со станции S_1 в момент времени t_d , и прибывающий на станцию S_2 в момент времени t_a . Длина элементарного соединения c , обозначается $\text{length}(c)$, это время, которое прошло между прибытием и отправлением c . Расписание действует для числа N дней

курсирования. Каждому маршруту соответствует битовое поле из N битов, определяющих, в какие дни маршрут курсирует. На станции $S \in B$ возможна пересадка с одного маршрута на другой, если время между отправлением и прибытием на станцию S больше, либо равно **минимальному времени пересадки** для этой станции. Для станций расположенных близко друг к другу возможно прохождение пешком, которое описывается введением, так называемых **пеших ребер** между станциями. Формально, мы рассматриваем пешее ребро как элементарное соединение c , где маршрут Z , времена отправления и прибытия t_d и t_a не определены, но $\text{length}(c)$ определено и равно времени пешего перехода.

Пусть $P = (c_1, \dots, c_k)$ – последовательность элементарных соединений (включая пешие ребра), с заданными отправлениями $dep_i(P)$ и прибытиями $arr_i(P)$ для каждого элементарного соединения c_i , $1 \leq i \leq k$. Полагаем, что времена прибытия и отправления включают также даты прибытий и отправлений и учитывают время, прошедшее с первого дня поездки. Время $t = a * M + b$, где $a \in [0, 364]$, M – кол-во минут в сутках, $b \in [0, M - 1]$. Последовательность P называется **постоянным соединением** от станции $A = S_1(c_1)$ до станции $B = S_2(c_k)$, если удовлетворяет следующим условиям:

$$\begin{aligned}
 & c_i \text{ действует 1 день } [dep_i(P)/M], \\
 & S_2(c_i) = S_1(c_{i+1}), \\
 & dep_i(P) \equiv t_d(c_i) \pmod{M}, \\
 & arr_i(P) = dep_i(P) + \text{length}(c_i), \\
 & dep_{i+1}(P) - arr_i(P) \geq \begin{cases} 0, & \text{если } Z(c_{i+1}) = Z(c_i) \text{ или } c_i - \text{пешее ребро} \\ \text{transfer}(S_2(c_i)), & \text{в противном случае} \end{cases}
 \end{aligned}$$

Наиболее известная частная задача о расписаниях называется также **задачей наискорейшего прибытия**. Запрос (A, B, t_0) определяет станцию отправления A , станцию прибытия B и время отправления t_0 . Соединение допустимо, если отправление из A не происходит раньше заданного t_0 . Критерий оптимизации – минимизировать разницу между временем прибытия и заданным временем отправления. В другой задаче **минимального количества пересадок** необходимо отыскать путь с наименьшим количеством пересадок для станций A и B , вообще не принимая во внимание времена прибытий и отправлений.

Представленные подходы могут быть использованы для **моделирования реальной задачи**, учитывающей сложные условия, например, время пересадок. Чтобы учесть времена пересадки во время-расширенной модели, на графе

создается копия всех узлов прибытия и отправления со станции, которые мы называем **узлами пересадки**. Для каждого узла прибытия существует два исходящих ребра: одно ребро к отправлению того же маршрута, второе ребро – к узлу пересадки со временем больше или равным сумме времен прибытия в узел и минимальным временем требуемым на пересадку на данной станции.

Часто пассажиры заинтересованы в том, чтобы найти самый дешевый путь из А в В в заданном интервале времени. К сожалению, тарифная система в большинстве стран настолько сложна, что практически невозможно решить такую задачу поиска точно и одновременно быстро. Даже в стандартных тарифах стоимость проезда обычно не аддитивна. Еще хуже обстоит дело, если требуется учесть специальные тарифы. **Многокритериальная оптимизация** по нескольким критериям необходима, когда пассажиру требуется найти путь с минимальным количеством пересадок, который начинается после заданного времени и не заканчивается слишком поздно. Вычисление оптимального пути по нескольким критериям сводится к решению задачи поиска кратчайшего пути по нескольким критериям (MOSP, multi-objective shortest path) – основная задача в области многоцелевой оптимизации. Задача многокритериальной оптимизации обычно NP – полная. Так как даже для очень простых графов (цепь узлов, соединенных ребрами) в задаче с двумя критериями количество решений в наборе Парето растет экспоненциально. Поэтому на практике для нахождения решения многокритериальной задачи используются приближенные подходы.

Исследования известных алгоритмов поиска по расписаниям показали, что без применения специальных **ускоряющих техник** производительность оказывается недостаточной. Средняя производительность особенно важна в системах с центральным сервером, который должен обрабатывать тысячи запросов одновременно, например, в Интернет или на транспортных терминалах (вокзалах, аэропортах). Техника **«ограничения угла»** использует информацию о пространственном расположении географических объектов. На этапе **препроцессинга** применяется алгоритм Дейкстры, чтобы вычислить кратчайшие пути из узла до всех других станций. Результаты вычислений не сохраняются, так как это потребовало бы слишком много места, но сохраняются два значения углов $a(v, w)$ и $b(v, w)$, которые сопоставляются ребру графа. Пусть (v, w) – ребро соединяющее событие v с другим событием w на время-расширенном графе и пусть S_v и S_w – станции, к которым принадлежат эти события. Рассмотрим угловой сектор между углами $a(v, w)$ и $b(v, w)$ с вершиной в точке

S_p . Углы a и b определяются таким образом, что конечные точки кратчайших путей, проходящих через ребро, будут внутри сектора. Позднее, во время работы алгоритма ребра (v, w) могут быть проигнорированы поиском, если целевой узел не входит в угловой сектор. Основная идея другой техники **«выбора станций»** применяется во многих планировщиках маршрута на транспорте. Пусть $G = (V, E)$ – означает маршрутный граф. Пусть V^{\sim} – набор выбранных узлов, из которых строится вспомогательный граф G^{\sim} . Вспомогательный граф G^{\sim} и веса ребер создаются и задаются только один раз на этапе препроцессинга. После этого любой запрос может быть обработан на вспомогательном графе G^{\sim} и путь будет соответствовать кратчайшему пути на графе G . Техника **«поиска в направлении цели»** использует потенциальную функцию на наборе узлов. Веса ребер в очереди изменяются так, чтобы направить поиск по графу в сторону цели. Пусть λ – эта потенциальная функция. Новый вес ребра (v, w) определяется как $l(v, w) := l(v, w) - \lambda(v) + \lambda(w)$. **«Иерархический поиск»** требует препроцессинга на котором исходный граф $G = (V, E)$ разбивается на несколько уровней $l + 1$ и дополняется кратчайшими путями между определенными узлами. Чтобы найти кратчайший путь между узлами, алгоритму Дейкстры достаточно рассмотреть меньший подграф многоуровневого графа. Еще одна интересная техника использует понятие **«радиуса достижения»**. Если v – вершина на кратчайшем пути P из s в t , то радиус достижения по отношению к P , $r(v, P)$ есть минимум от длины префикса P (часть пути от s к v) и длины суффикса (часть пути от v к t). Радиус достижения узла v , $r(v)$ – это максимум $r(v, P)$ по всем кратчайшим путям, которые содержат v . Вычислив предварительно $r(v)$ для всех узлов можно использовать эту информацию во время работы алгоритма Дейкстры. Техника **«прямоугольника кратчайших узлов»** похожа на метод «ограничения угла». На этапе препроцессинга обрабатываются все деревья кратчайших путей. Для каждого ребра $e \in E$, вычисляется набор узлов $S(e)$, кратчайшие пути к которым проходят через ребро e , и для каждого $e \in E$, сохраняется ограничивающий прямоугольник $S(e)$. Во время работы алгоритма можно оперативно определить, по какому ребру необходимо двигаться дальше, исключая из рассмотрения те ребра, которые заведомо не ведут к цели.

В параграфе 1.8 сделан обзор архитектур распределенных систем эширования в Интернет. **Кэширование данных** – фундаментальная технология, позволяющая уменьшить время доступа к объекту данных. Пусть N – общее

число документов. Пусть $P_n(i)$ – условная вероятность того, что пришедший запрос относится к документу i . Расположим документы в порядке их популярности, где документ i – i -ый по популярности документ. Полагаем, что $P_n(i)$, $i = 1, 2, \dots, N$ имеет Zipf распределение

$$P_N(i) = \frac{\Omega}{i},$$

где

$$\Omega = \left(\sum_{i=1}^N \frac{1}{i} \right)^{-1}.$$

В случае бесконечного кэша и конечного числа запросов, количество попаданий как функцию количества запросов можно получить следующим образом: $H(R) = \sum_{i=1}^N P_N(i) (1 - (1 - P_N(i))^R)$.

Асимптотическое поведение коэффициента попаданий $H(R) \approx \Omega \ln R$.

Это поведение можно очевидным образом получить приближением $H(R)$:

$$H(R) = \sum_{i=1}^N \frac{\Omega}{i} f(i) \approx \sum_{i=1}^{R\Omega} \frac{\Omega}{i} \approx \Omega \ln(R\Omega).$$

В случае ограниченного размера кэша объемом C документов, на который поступает бесконечный поток запросов, асимптотическое значение коэффициента $H(C)$ может быть вычислено так:

$$H(C) = \sum_{i=1}^C P_N(i) \approx \Omega \ln C.$$

Вероятностное распределение $d(k)$ того, что следующий запрос к документу придет через k запросов, определяется как

$$d(k) \approx \frac{1}{k \ln N} \left(\left(1 - \frac{1}{N \ln N}\right)^k - \left(1 - \frac{1}{\ln N}\right)^k \right).$$

Эффективность кэша в Интернет сильно зависит от корреляции интересов в группе пользователей. Чтобы повысить эффективность кэширования, несколько кэшей объединяют в систему. Существует два основных подхода к построению больших систем кэширования: **иерархический** и **распределенный**. В иерархическом кэшировании кэши располагаются на различных сетевых

уровнях. В распределенном кэшировании не устанавливается промежуточных кэшей. Сетевая топология моделируется в виде дерева. Пусть O – плотность ветвей дерева. Пусть N – общее число документов и S – размер документа, документы обновляются с периодичностью Δ , запросы к документу i , $1 \leq i \leq N$ в кэше учреждения распределены по закону Пуассона со средней интенсивностью $\lambda_{i,i}$. Тогда общее число запросов к документу i равно $\lambda_{tot,i} = \lambda_{i,i} O^{2H}$ и тоже распределено по Пуассону. Пусть интенсивность запросов к кэшу учреждения для всех N документов, $\beta_I = \sum_{i=1}^N \lambda_{i,i}$. β_I распределена по закону Zipf:

$$\lambda_{i,i} = \beta_I \frac{\sigma}{i^\alpha}$$

где константа α определяет, насколько пологим будет Zipf распределение и σ как:

$$\sigma = \left(\sum_{i=1}^N \frac{1}{i^\alpha} \right)^{-1}$$

В иерархическом кэшировании имеет место меньшее время соединения, чем в распределенном кэшировании. Но распределенное кэширование имеет меньшее время передачи, чем иерархическое кэширование, так как больше трафика проходит через менее загруженный нижний сетевой уровень. Что касается нагрузки на сеть, то при иерархическом кэшировании она суммарно меньше, чем при распределенном, так как иерархическое кэширование использует промежуточные кэши. С другой стороны, распределенное кэширование хорошо распределяет сетевую нагрузку на систему, в отличие от иерархической, где могут возникать перегруженные узлы системы.

Разработке алгоритмов и архитектуры справочной системы посвящена **вторая глава** диссертации. В начале главы предложен оригинальный алгоритм поиска пути. Далее рассмотрены технологии взаимодействия с внешними автоматизированными системами. В третьей части описан цикл обработки клиентского запроса. В четвертой части рассмотрены основные моменты проектирования архитектуры и предложена аналитическая модель информационной системы.

В параграфе 2.1 разработан оригинальный алгоритм поиска, который строго решает поставленную двухкритериальную задачу. На первом этапе проводится поиск по «виртуальному» графу. Этот поиск находит все пути с одинаковым

минимальным количеством пересадок. Найденные пути анализируются на следующем шаге алгоритма, где для каждого из них проверяется допустимость с учетом дней курсирования и наличия мест, и решается задача наискорейшего прибытия. Затем среди допустимых путей выбирается наискорейший. В том случае, если все пути оказались недопустимы, то повторяется первый этап поиска по графу с исключением найденных ранее путей. Основные функциональные части алгоритма: (i) алгоритм оптимистического поиска в графе, (ii) методы выбора наискорейшего пути из оптимистически найденных на графе, (iii) проверка наличия свободных мест. В графе хранится информация обо всех допустимых транспортных маршрутах и решается задача поиска пути с минимальным количеством пересадок. Все узлы графа, между которыми можно проехать без пересадки, соединим ребрами веса 1. Поиск такого пути можно производить с помощью двунаправленного алгоритма Дейкстры. В реализации алгоритма необходимые ребра рассчитываются динамически, и по окончании работы алгоритма память освобождается. В противном случае потребовался бы значительный объем памяти для хранения данных. В оригинальном алгоритме подсчитывается не минимальное число пересадок, а минимальное число маршрутов, которое на единицу больше числа пересадок. Известны два распространенных способа представить граф: как набор смежных вершин или как матрицу смежности. Алгоритм реализован на встроенном языке базы данных и работает с таблицами и записями. Хранение расписаний в таблицах и реализация алгоритма на уровне базы данных оказывается на практике достаточно эффективной и не требует избыточного преобразования данных. Для каждой записи о расписании хранится идентификатор узла, идентификатор маршрута, время прибытия, время отправления и другая информация. Алгоритм Дейкстры, хотя и может быть применим к данной задаче, но не является достаточно быстрым для практической реализации в справочной системе. Существующие техники ускорения не могут быть использованы в информационной системе. В них используется длительный препроцессинг данных и хранение вспомогательной информации для каждого ребра. Препроцессинг может длиться десятки часов даже на современных вычислительных комплексах, а это неприемлемо, когда информация обновляется в режиме реального времени из внешних автоматизированных систем. Кроме того, в оригинальной реализации информация о ребрах графа вообще не хранится в системе, поэтому предложены несколько методов ускорения поиска,

лишенные указанных недостатков. Метод **пересадочного графа** заключается в выделении потенциальных узлов пересадки. В процессе обновления данных о расписаниях, можно анализировать упомянутые в расписании станции и определять потенциальные узлы пересадки. Важно, что список станций-пересадок обновляется в режиме реального времени и не требует масштабных вычислений. В результате создается пересадочный граф меньшего размера. Другой метод - **балансировка двунаправленным поиском** - заключается в выборе прямого или обратного поиска на каждом шаге итерации с учетом количества смежных станций. В работе этого метода используется информация о количестве смежных станций, которая может быстро вычисляться в режиме реального времени. Еще один возможный подход - это **ускорение последней итерации**. Последняя итерация поиска требует больше времени, чем предыдущие, так как поиск охватывает все большее количество станций. Для каждой станции вычисляется ограничивающий прямоугольник станций, достижимых без пересадок. На последнем шаге итерации используется информация об ограничивающем прямоугольнике каждой станции для выяснения, может ли станция быть пересадкой или ее заведомо нужно исключить. Еще один эффективный метод ускорения - это метод **ускорения A^*** . Он основан на эвристическом алгоритме A^* и превосходит алгоритм Дейкстры в быстродействии. Хотя A^* в данной задаче и не является точным подходом, но на практике его можно эффективно применить, и найденное решение с высокой вероятностью оказывается оптимальным. Важно, что все предложенные методы ускорения могут применяться в оригинальном алгоритме поиска совместно.

Данные о расписаниях непрерывно обновляются, и все вспомогательные данные должны вычисляться быстро. Поэтому для построения пересадочного графа анализируются только станции, расписание по которым изменилось. После того, как пути с минимальным количеством пересадок найдены, проводится проверка их допустимости, и выбирается наискорейший путь. Поиск кратчайшего пути оказывается вычислительно менее сложной задачей, чем первый шаг поиска по графу. Данные о наличии свободных мест - быстро обновляющаяся часть транспортной информации. Если пропускная способность канала связи с внешними системами ограничена, то физически невозможно поддерживать актуальность такой информации в справочной системе. Но при большом количестве клиентских запросов перенаправление их в автоматизированную систему приводит к перегрузке канала связи. Если использовать кэширование, то

нет необходимости обращаться к источникам и нагружать канал связи при каждом запросе, но тогда возникает ряд других проблем. Поэтому выбор способа зависит от параметров источника. Когда существует много объектов данных, что является более естественным, то задача существенно усложняется. Для каждого объекта данных выбирается своя стратегия репликации. Полная стоимость всех запросов определяется выражением:

$$P = \sum_{\alpha_1 \dots \alpha_n} p(\alpha) \cdot f(\alpha) .$$

Для решения сложной задачи поиска пути цикл обслуживания запроса включает несколько основных шагов, описанных в параграфе 2.3:

1. Поиск возможных путей на графе между пунктами отправления и прибытия без учета дней курсирования.
2. Первичная фильтрация найденных путей и исключение недопустимых по дате поездки, виду транспорта.
3. Выбор из оставшихся вариантов оптимального пути по заданному критерию.
4. Проверка выбранного пути на наличие свободных мест.

Каждый из этапов характеризуется средним временем обработки и вероятностью «промаха». В диссертации проанализирован цикл обслуживания запроса, что позволяет разработать методы повышения производительности.

Архитектура справочной системы должна обеспечивать высокую производительность при большом количестве пользовательских запросов. Справочная система рассматривается в параграфе 2.4 как распределенная информационная система. Предлагается аналитическая модель системы, которая позволяет оценить время обработки справочных запросов. Одна из особенностей задачи, которая учитывается в модели - географическое распределение запросов от пользователей. Существует два вида источников данных: (i) региональные автоматизированные системы (охватывают транспорт отдельного города или региона) и (ii) глобальные автоматизированные системы на транспорте (охватывают одну или несколько стран). Подавляющее большинство пользователей интересуется справочная информация по тому региону, где они находятся, поэтому нецелесообразно копировать региональную информацию на межнациональный уровень. Что касается глобальных автоматизированных

систем, запросы к ним поступают со всех регионов, и объем данных слишком велик, чтобы поддерживать их актуальную копию в информационной системе. Предполагается, что запросы к данным i , $1 \leq i \leq N$ распределены по закону Пуассона со средней интенсивностью $\lambda_{i,i}$. Считается, что все документы запрашиваются независимо друг от друга, то есть пренебрегается любая корреляция между запросами. Пусть интенсивность запросов к кэшу нижнего уровня для всех N документов, $\beta_i = \sum_{i=1}^N \lambda_{i,i}$. β_i распределена по закону Zipf: $\lambda_{i,i} = \beta_i \frac{\sigma}{i^\alpha}$, где константа α определяет, насколько пологим будет Zipf распределение и σ определяется как:

$$\sigma = \left(\sum_{i=1}^N \frac{1}{i^\alpha} \right)^{-1}.$$

Общая задержка T при получении документа состоит из задержки времени соединения T_c и времени передачи T_t

$$E[T] = E[T_c] + E[T_t],$$

где $E[T]$ – математическое ожидание времени полной передачи документа.

Время соединения зависит от количества сетевых связей, которое необходимо пройти от справочной системы до источника данных. Пусть L_i количество связей, которое прошел запрос к документу i , прежде чем достиг нужного сервера. Время соединения с сервером глобальной системы T_c^h определяется:

$$E[T_c^h] = 3d \cdot \sum_{l \in \{0, H, 2H, 2H+z\}} P(L_i = l) \cdot l.$$

Время соединения с удаленным сервером локальной системы T_d^h определяется выражением:

$$E[T_c^d] = 3d \cdot \sum_{l=0}^{2H} P(L_i = l) \cdot 2l + 4d \cdot P(L_i = 2H + z) \cdot (2H + z).$$

Чтобы достичь удаленного локального сервера, запрос сначала поднимается вверх по сетевой иерархии, а затем опускается вниз к локальному серверу, содержащему документ i , проходя при этом $2l$ связей. Сетевой уровень L_i , до которого поднимается запрос (вероятность того, что количество связей, которое пришлось преодолеть больше или равно l) $P(L_i \geq l)$ – это:

$$P(L_i \geq l) = \frac{1}{\Delta} \int_0^{\Delta} P(L_i \geq l|\tau) d\tau ,$$

где $P(L_i \geq l|\tau)$ – вероятность того, что нет запросов к документу i в поддереве с корнем в $l - 1$ в интервале времени $[0, \tau]$

$$P(L_i \geq l|\tau) = e^{-O^{l-1}\lambda_{l,i} \cdot \tau} .$$

В результате получаем вероятность:

$$P(L_i \geq l) = \frac{1}{O^{l-1}\lambda_{l,i} \cdot \Delta} \left(1 - e^{-O^{l-1}\lambda_{l,i} \cdot \Delta}\right) .$$

Чтобы вычислить времена передачи, сначала определяется суммарная частота запросов на каждом сетевом уровне l . Суммарная частота к глобальной внешней системе β_l^h определяется выражением:

$$\beta_l^h = \begin{cases} \beta_l, & l = 0 \\ O^l \beta_l \cdot (1 - hit_l), & 0 < l < H \\ O^l \beta_l \cdot (1 - hit_R), & H \leq l < 2H \\ O^{2H} \beta_l \cdot (1 - hit_N), & 2H \leq l < 2H + z \end{cases}$$

Вероятность обработки на каждом сетевом уровне $hit_l = \sum_{i=1}^N \left(\frac{\lambda_{l,i}}{\beta_l}\right) \cdot P(L_i \leq l)$.

Суммарная частота запросов к локальной системе β_l^t определяется выражением

$$\beta_l^t = O^l \beta_l \cdot ((1 - hit_l) + (hit_N - hit_l)),$$

для $0 \leq l < 2H$, и $O^{2H} \beta_l \cdot (1 - hit_N)$, для $2H \leq l < 2H + z$.

Время передачи данных из различных систем рассматривается в параграфе 2.4.

Время передачи из глобальной информационной системы $E[T_t^h]$:

$$E[T_t^h] = \sum_{l \in \{0, H, 2H, 2H+z\}} E[T_t^h | L_i = l] \cdot P(L_i = l) ,$$

где $E[T_t^h | L_i = l]$ – время передачи из соответствующего сетевого уровня глобальной информационной системы. Время передачи из региональной информационной системы $E[T_t^d]$ определяется выражением:

$$E[T_t^d] = \sum_{l=0}^{2H+z} E[T_t^d | L_i = l] \cdot P(L_i = l) ,$$

где $E[T_t^d | L_i = l]$ – время передачи через соответствующий сетевой уровень из локальной информационной системы.

Пусть \tilde{S} – средний размер документов. Теория очередей M/D/1 дает для запросов к глобальным системам выражение:

$$E[T_t^h] = \frac{\tilde{S}}{c_l - \beta_l^h \cdot \tilde{S}} \cdot \left(1 - \frac{\beta_l^h \cdot \tilde{S}}{2c_l}\right).$$

Аналогично для запросов к локальным системам:

$$E[T_t^d] = \frac{\tilde{S}}{c_l - \beta_l^d \cdot \tilde{S}} \cdot \left(1 - \frac{\beta_l^d \cdot \tilde{S}}{2c_l}\right).$$

Время полной обработки запроса выводится в параграфе 2.4 и определяется наибольшим из времен взаимодействия с информационными системами. Время установления соединения с удаленной локальной системой оказывается больше этого времени для глобальной системы:

$$E[T_c^d] = 3d \cdot \sum_{l=0}^{2H} \left[\frac{1}{O^{l-1} \lambda_{l,i} \cdot \Delta} \left(1 - e^{-O^{l-1} \lambda_{l,i} \cdot \Delta}\right) - \frac{1}{O^l \lambda_{l,i} \cdot \Delta} \left(1 - e^{-O^l \lambda_{l,i} \cdot \Delta}\right) \right] \cdot 2l + 3d \cdot \frac{1}{O^{2H-1} \lambda_{l,i} \cdot \Delta} \left(1 - e^{-O^{2H-1} \lambda_{l,i} \cdot \Delta}\right) \cdot (2H + z).$$

Программной реализации справочной системы посвящена третья глава диссертации. В главе описан комплекс программ и реализация алгоритмов, которые обеспечивают функционирование информационно-справочной системы. Программный комплекс включает в себя следующие компоненты:

1. Специализированная база данных геоинформации и расписаний.
2. Средства подготовки данных для геоинформационной системы и БД.
3. Средства импорта данных из промежуточного формата XML.
4. Реализация алгоритмов поиска маршрута на транспорте.
5. Службы интеграции с другими информационными системами, включающие:
 - а. интеграционный сервис, функционирующий в режиме on-line;
 - б. эмулятор терминала системы ЭКСПРЕСС.
6. Справочный интернет-портал для доступа к информационной системе.

Специализированная база данных оптимизирована для хранения и анализа информации по транспорту. Она содержит несколько десятков таблиц и реализацию базовых алгоритмов работы с данными, которые запрограммированы в хранимых процедурах на языке T-SQL. Каждый объект в базе данных имеет как координатную привязку, так и привязку к совокупному графу вершин и соединений.

Средства подготовки данных для системы описаны в параграфе 3.3. Особенности информационной системы потребовали создания специального программного инструментария – среды визуального описания транспортного графа. Программа представляет собой Windows-приложение, написанное на языке C++. С ее помощью можно описать фрагменты единого графа. Результаты работы программы сохраняются в файлы формата XML. Для каждого региона создается отдельный XML файл, где указывается расположение региональных транспортных узлов. Таким способом можно последовательно описать все фрагменты железнодорожной и автобусной сети. С помощью разработанной программы было описано большинство железнодорожных станций, вокзалов, платформ на территории всего бывшего СССР и некоторых прилегающих стран, а также основные автобусные станции нескольких регионов России.

После того, как географические данные подготовлены в формате XML, они загружаются в систему с помощью **программы импорта данных**. Программа подключается к базе данных системы, получает на вход данные в формате XML и вносит изменения в БД. Текущая база данных загружена из нескольких десятков файлов XML и образует граф из более чем двадцати тысяч вершин. Процесс загрузки состоит из следующих шагов: считывание данных из файла, анализ данных, устранение неоднозначностей, формирование объектной структуры, сохранение данных. Программа импорта состоит из двух функциональных модулей: (i) загрузчик географических данных, (ii) загрузчик расписаний. Загрузчик географических данных обрабатывает подготовленные оператором файлы данных. Алгоритм импорта географических объектов запрограммирован следующим образом: 1) распознавание названия и типа объекта; 2) поиск объектов в базе данных с похожим названием; 3) добавление объекта в базу данных; 4) обновление топологических связей.

Реализация алгоритма поиска и модуль ядра описаны в параграфе 3.7. Используя идентификаторы пунктов назначения и отправления, модуль собирает необходимую информацию о географических объектах. Если прямой маршрут не

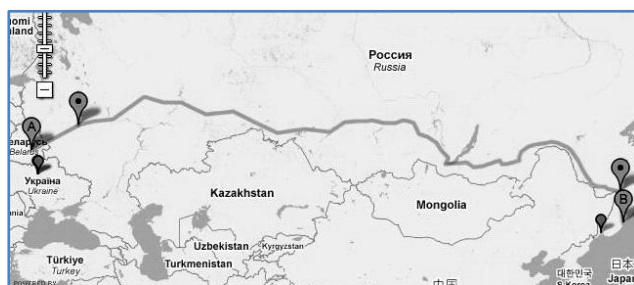
найден, то алгоритм начинает поиск сложного маршрута. Хранимая процедура возвращает таблицу данных, в которой содержится информация о найденных путях проезда. Алгоритм рассматривает все найденные пути и для каждого из них вычисляет наискорейшее время прибытия. После того как из списка удалены недопустимые пути, и для каждого из доступных путей вычислено наискорейшее время прибытия, список путей сортируется по времени наискорейшего прибытия. На последнем этапе лучший из найденных путей отправляется на проверку в модуль взаимодействия с автоматизированными системами.

Взаимодействие с внешними системами представляет отдельную подсистему, в которую входят: модуль интеграции в составе модуля поиска пути; сервис интеграции с внешними системами; эмулятор терминала ЭКСПРЕСС; программа контроля взаимодействия с внешними системами. Подсистема работает следующим образом. Когда пользователь делает запрос, то сначала выполняются основные шаги алгоритма поиска: (i) находятся возможные пути проезда, (ii) проверяются даты курсирования маршрутов. После того как найден предполагаемый оптимальный путь, делается внутренний запрос в подсистему взаимодействия, которая формирует запрос во внешние системы. Способ отправки запроса зависит от особенностей внешней системы. Если запрос **синхронный**, то обслуживающий процесс выполняет его немедленно. Если используется **асинхронный** способ взаимодействия, например, с системой ЭКСПРЕСС, то запрос ставится в очередь. Когда в очереди появляются запросы на обслуживание, к работе подключается программа-сервис взаимодействия. Запрос берется из очереди и трансформируется сервисом в запрос по протоколу, поддерживаемому автоматизированной системой. Одновременно могут работать несколько сервисов взаимодействия, запущенные на нескольких вычислительных машинах и просматривающие единую очередь запросов. Сервисы могут специализироваться на обслуживании запросов к определенным внешним системам. Взаимодействие с системой ЭКСПРЕСС осуществляется через специальный сервис-эмулятор терминала. При получении запроса, для которого необходимо получить информацию из системы ЭКСПРЕСС, эмулятор терминала преобразует запросы системы в пакеты BSC-3 сети ЭКСПРЕСС, инкапсулирует их в пакеты TCP/IP и передает на шлюз доступа в ЭКСПРЕСС.

Далее шлюз обменивается информацией с ХОСТ-ЭВМ и возвращает ответ эмулятору. Процесс протекает асинхронно: один запрос системы может отображаться в серию запросов-ответов между эмулятором и ХОСТ-ЭВМ.

Для получения справочной информации о возможности проезда через сеть Интернет разработан справочный портал. Портал обеспечивает пользователей справочной информацией о маршрутах пассажирского транспорта России, ближнего и дальнего зарубежья. Сайт предоставляет следующую информацию:

- поиск пунктов пересадки, когда нет прямого пути проезда;
- поиск путей только на указанном виде транспорта (автобусы, поезда);
- поиск пути проезда между двумя пунктами с пересадкой в третьем явном указанном пункте;
- поиск интермодального пути проезда (на разных видах транспорта);
- поиск пути проезда в заданную дату (маршруты транспорта, которые не удовлетворяют указанной дате, не отображаются в результатах);
- поиск пути проезда со всех вокзалов города, или явное указание с какого вокзала искать путь;
- отображение найденных путей проезда на интерактивной карте;
- отображение схемы всех прямых маршрутов транспортного узла.



Откуда-->Куда	N	Время
Бобруйск		отпр. 15:20
Москва		приб. 06:18
Москва		время на пересадку - 15 ч 7 мин Расстояние 9 км. Используйте городской транспорт
МОСКВА ЯРОСЛАВСКАЯ		
МОСКВА ЯРОСЛАВСКАЯ		отпр. 21:25
Хабаровск 1		приб. 11:00
Хабаровск 1		время на пересадку - 2 ч 50 мин Расстояние 3 км. Используйте городской транспорт
Хабаровск		
Хабаровск		отпр. 13:50
Дальнегорск		приб. 00:00

Рисунок 1. Результат поиска пути проезда Бобруйск-Дальнегорск

В главе 4 продемонстрированы результаты использования информационно-справочной системы, и проведено исследование ее работы. В параграфе 4.1 приводятся примеры результатов поиска по различным видам запросов:

- поиск маршрута с пересадкой на одном виде транспорта;
- поиск маршрута с одной пересадкой на нескольких видах транспорта;
- поиск маршрута с одной пересадкой в крупном транспортном узле;
- поиска маршрута с несколькими пересадками на одном виде транспорта.

В параграфе 4.2 проводится статистический анализ транспортного графа. Проводится три серии вычислений по исследованию плотности графа вокруг Москвы, Калининграда и Владивостока. Результаты исследования приведены в таблицах и на графиках. Исследования зависимости количества станций как функции удаленности от исследуемой станции подтверждают высокую неоднородность транспортного графа. Например, на расстоянии 500 км от Москвы расположено более 3000 станций, а на таком же расстоянии от Владивостока - только 195 станций. Аналогичный

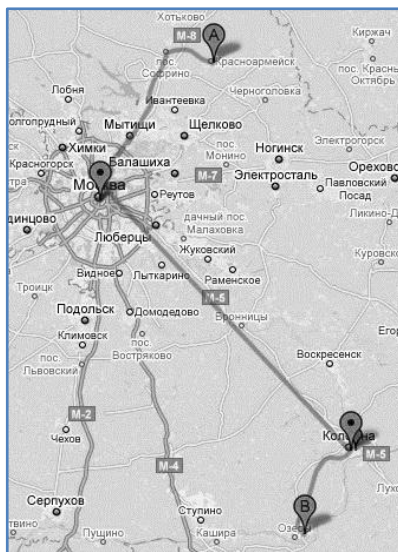


Рисунок 2. Результат поиска пути проезда по Московской области

статистический анализ проводится для пересадочного графа. Вид графиков и данные таблиц показывают аналогичную картину неоднородности для пересадочного графа. Далее в главе исследуются транспортные маршруты поездов дальнего следования.

Далее находится зависимость количества достижимых пунктов назначения от минимального числа необходимых пересадок. Проводится серия вычислений для пунктов Москва, Калининград и Владивосток. Как и следовало ожидать, для проезда из Москвы в другие пункты в среднем требуется меньше пересадок, чем для проезда из Владивостока. Приводится анализ зависимости количества пунктов назначения от расстояния и заданного количества пересадок. Из графиков видно, что наиболее вероятное расстояние до пункта назначения, до

которого можно добраться из Москвы минимум с одной пересадкой, составляет 800 км. Для Калининграда это расстояние составляет уже 1200 км, а для Владивостока - 6500 км.

В параграфе 4.2 проводится анализ справочных запросов, сделанных пользователями на сайте. За несколько месяцев пользователями выполнено около полумиллиона запросов. Анализ этих данных позволяет получить информацию о поездках, которые интересуют пользователей. Показано, что чем меньше расстояние между пунктами, тем больше пользователей интересует возможностью проезда. На основе вышеупомянутых исследования проводится анализ суммарного времени

обработки запросов информационно-справочной системой в зависимости от расстояния между пунктами. Исследования показали, что наибольшее

количество ресурсов система расходует на обработку запросов на средние расстояния. В параграфе 4.2 также исследуется зависимость времени обработки запроса от расстояния между пунктами. Математическое исследование дает хорошую аппроксимацию логарифмическим приближением. Таким образом, эмпирически получено, что среднее время поиска пропорционально логарифму от расстояния между пунктами. В параграфе 4.3 проводится анализ производительности методов ускорения и анализ особенностей алгоритма A*. Показано, как изменение параметра эвристики влияет на точность алгоритма A*. Приведены результаты сравнений производительности методов ускорения. Результаты показывают, что наиболее эффективным методом ускорения оказывается метод балансировки поиском. Наибольшую производительность дает суперпозиция всех предложенных методов ускорения.

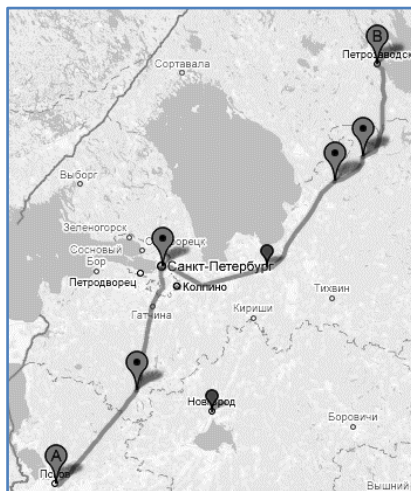


Рисунок 3. Результат поиска пути Псков-Петрозаводск только на электричках. Отмечены возможные места пересадок.

В **заключении** приводятся основные результаты и выводы диссертационной работы.

В **приложение** вынесены акты о внедрении, описания форматов используемых данных, описания вспомогательных программ, технические требования к оборудованию.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

- В диссертации проведен сравнительный анализ методов и алгоритмов поиска пути с учетом расписаний движения транспорта, а также выявлены недостатки существующих алгоритмов.
- Разработана сетевая модель взаимодействия информационно-справочной системы с существующими информационными системами на транспорте, позволяющая спроектировать оптимальную архитектуру справочной системы.
- Разработан оригинальный алгоритм поиска оптимальных путей проезда на пассажирском транспорте с учетом пересадок и наличия мест.
- Разработана база данных и комплекс программ для обслуживания справочной системы.
- Проведены испытания разработанной справочной системы и сделаны оценки эффективности разработанных алгоритмов.
- Разработан интернет-портал <http://transport.marshruty.ru> для доступа к информационно-справочной системе.
- Созданная система предоставляет следующую справочную информацию о возможности проезда:
 - расписания для нескольких видов транспорта;
 - выбор оптимальных путей по критерию минимального количества пересадок и минимальному времени в пути на железнодорожном и автобусном транспорте;
 - учет наличия свободных мест на поезда при поиске пути проезда;
 - пригородные маршруты подъезда к железнодорожным вокзалам.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ ПО ТЕМЕ ДИССЕРТАЦИИ

- [1] Вишнеvский В. М., Железов Р. В. Принципы построения и реализация автоматизированной информационно-справочной системы поиска оптимальных путей проезда на пассажирском транспорте // Проблемы Управления. – 2009. – №1. – С. 33 – 37.
- [2] Железов Р. В. Новые технологии справочно-информационного обслуживания пассажиров // Железнодорожный транспорт. – 2008. – №8. – С. 34 – 36.
- [3] Вишнеvский В. М., Железов Р. В., Атанасова Т. А. Единая справочная информационная система на пассажирском транспорте // Труды междунар. сем. “Распределенные компьютерные и телекоммуникационные сети” DCCN`2005. – М.: Техносфера, 2005. – С. 165 – 172.
- [4] Железов Р. В. Поиск маршрута с учетом расписаний движения транспорта // Труды конф. “Информационные технологии и системы” ИТиС`07. - М.: ИППИ РАН, 2007. – С. 171 – 173
- [5] Железов Р. В., Особенности алгоритма поиска маршрута на транспорте с учетом расписаний движения // Труды междунар. сем. “Распределенные компьютерные и телекоммуникационные сети” DCCN`2007. – М.: ИППИ РАН, 2007. – С. 28 – 32.
- [6] Железов Р. В. Особенности архитектуры распределенной справочной системы на транспорте // Труды всеросс. конф. «Технологии Microsoft в теории и практике программирования». – М.: Вузовская книга, 2007. – С. 16 – 18.
- [7] Железов Р. В. Реализация единой справочной системы пассажирского транспорта на базе технологий Microsoft // Труды всеросс. конф. «Технологии Microsoft в теории и практике программирования». – М.: МГТУ им. Н.Э. Баумана, 2006. – С. 10 - 13.
- [8] Железов Р. В., Оригинальный алгоритм поиска кратчайшего маршрута с пересадкой // Труды XLVII конф. МФТИ «Современные проблемы фундаментальных и прикладных наук». – М.: МФТИ, 2004. – С. 29 – 30.
- [9] Железов Р. В. Проблема репликации в контексте «Единой справочной системы на транспорте» // Труды XLVIII конф. МФТИ «Современные проблемы фундаментальных и прикладных наук». – М.: МФТИ, 2005. – С. 33 – 35.