УДК 004.78:025.4.036

ПРИНЦИПЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИИ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННО-СПРАВОЧНОЙ СИСТЕМЫ ПОИСКА ОПТИМАЛЬНЫХ ПУТЕЙ ПРОЕЗДА НА ПАССАЖИРСКОМ ТРАНСПОРТЕ

В.М. Вишневский, Р.В. Железов

Рассмотрена архитектура разработанной информационно-справочной системы поиска оптимальных путей проезда на пассажирском транспорте. Представлен оригинальный алгоритм поиска оптимальных путей с учетом расписаний пассажирского транспорта.

Ключевые слова: информационная система, оптимальный путь, пассажирский транспорт, расписание, Интернет.

ВВЕДЕНИЕ

Уровень справочно-информационного обслуживания пассажиров в Российской Федерации далек от совершенства. Даже в рамках отдельных видов транспорта (за исключением воздушного) отсутствует возможность поиска маршрута проезда с пересадкой. Например, на запрос пассажира о возможности проезда из Калининграда во Владивосток, действующая автоматизированная система «Экспресс-3» выдает ответ: «Прямое сообщение отсутствует. Подготовьте ответ вручную». Подобная задача на железнодорожном транспорте гораздо сложнее, чем на воздушном, так как число железнодорожных станций превосходит число аэропортов в сотни раз.

Еще более сложная задача — поиск интермодального маршрута (пути проезда с несколькими видами транспорта) [1—3]. Отметим, что транспортная сеть в Российской Федерации весьма неоднородна. Есть населенные пункты, куда можно попасть только по воздуху. Имеются регионы, лишенные железнодорожного транспорта; есть населенные пункты, до которых можно добраться только поездом. Поэтому разработка информационной системы, которая позволит объединить информацию из действующих автоматизированных систем на различных видах транспорта в целях получения наиболее полной справочной информации о возможности проезда с учетом пересадок и наличия мест на разных видах транспорта, весьма актуальна.

В Европе наибольшее распространение получила система HAFAS [4], которая используется на железнодорожном транспорте в нескольких странах. Она выдает удовлетворительные результаты поиска, хотя они не всегда оказываются оптимальными, поскольку алгоритм поиска реализован на базе эвристических методов с целью уменьшения пространства поиска. Применяется статический алгоритм без учета информации о времени проезда, что может приводить к потере оптимального пути.

Разработанная отечественная автоматизированная система лишена недостатков известных систем, в ней реализован оригинальный поисковый алгоритм и используется актуальная, динамически обновляемая информация о расписаниях разных видов пассажирского транспорта.

1. АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННО-СПРАВОЧНАЯ СИСТЕМА НА ПАССАЖИРСКОМ ТРАНСПОРТЕ

Разработанная информационно-справочная система осуществляет выдачу справочной информации через Интернет об оптимальном пути проезда на пассажирском транспорте между любыми пунктами России. В справке учитывается возможность пересадки между разными видами транспорта, расписания его движения и наличие свободных мест. Система представляет собой комплекс программ и программных модулей, которые обеспечивают ее функционирование [2]. Программный комплекс включает в себя:

- специализированную базу данных геоинформации и расписаний;
- средства подготовки данных для информационно-справочной системы;
- средства импорта данных из промежуточного формата XML;
- средства администрирования системы;
- модуль ядра с реализаций алгоритмов поиска пути проезда;
- службы интеграции с другими информационными системами:
  — интеграционный сервис, функционирующий в режиме on-line;
  — эмулятор терминала системы «Экспресс»;
- справочный Интернет-портал для доступа к информационной системе.

Специализированная база данных хранит геоинформацию, регулярно обновляемые расписания движения пассажирского транспорта и реализацию некоторых алгоритмов работы с данными, которые запрограммированы в виде хранимых процедур на языке T-SQL. Особенности информационно-справочной системы потребовали разработки специальных *средств подготовки данных* — программы визуального описания транспортного графа. Программа представляет собой Windows-приложение, с помощью которого описываются фрагменты транспортного графа. Программа сохраняет результат работы в файлы формата XML. Для каждого региона или транспортной организации может быть создан отдельный XML-файл, где указано расположение транспортных узлов. Таким способом по частям описываются фрагменты железнодорожной, автобусной или другой транспортной сети.

После того, как географические данные подготовлены в промежуточном XML-формате, они загружаются в систему с помощью *программы импорта данных*. Программа обрабатывает данные XML-формата, подключается к базе данных системы и добавляет информацию об узлах транспортного графа. Более подробно, 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: загрузчика географических данных и загрузчика расписаний.

Алгоритм импорта географических объектов запрограммирован следующим образом:

1) распознавание названия и типа объекта;

2) поиск объектов в базе данных со схожим названием;

3) добавление объекта в базу данных;

4) обновление топологических связей.

Алгоритм импорта расписаний отличается от импорта объектов:

1) определение структуры данных;

2) считывание данных в промежуточную таблицу;

3) объектное структурирование данных;

4) сохранение структур в базу данных;

5) корректировка вспомогательных данных.

Если в маршрут следования поезда входит станция, принадлежащая гиперузлу, то в базу данных добавляются дополнительные записи о расписании. В процессе импорта расписания маршрута специальным образом обновляется пересадочный подграф.

С помощью программы создания графов в базу данных загружено множество фрагментов транспортного графа с общей численностью более двадцати тысяч узлов. Сюда вошло большинство железнодорожных станций, вокзалов, остановочных пунктов на территории Российской Федерации, ближнего и дальнего зарубежья, а также основные автобусные станции нескольких регионов России.

Осуществление справочного запроса начинается с указания параметров (пункты отправления и назначения, виды транспорта, дата поездки и др.). После того, как пользователь указал параметры запроса и нажал кнопку «Поиск», параметры передаются в модуль поиска пути. Используя идентификаторы пунктов назначения и отправления, модуль получает необходимую информацию о географических объектах. Сначала алгоритм пытается отыскать беспересадочный маршрут между заданными пунктами. Если прямой маршрут не найден, то применяется оригинальный алгоритм поиска пути с пересадками. После того, как найдены возможные пути проезда с минимальным числом пересадок, проверяются дополнительные условия по каждому найденному пути и учитываются параметры поиска. На этом шаге работы алгоритма принимаются во внимание дни курсирования, пересечение государственных границ, длительность пересадок. Для каждого возможного пути определяется время отправления и прибытия наискорейшего маршрута.

На последнем шаге наиболее предпочтительный из найденных путей передается в подсистему взаимодействия с внешними автоматизированными системами. Подсистема предназначена для получения информации о наличии свободных мест. Данные о наличии свободных мест — наиболее быстро обновляющаяся часть транспортной информации. Если пропускная способность канала связи с автоматизированными системами ограничена, а объем данных велик, то физически невозможно импортировать актуальную информацию о наличии свободных мест из автоматизированных систем. Поэтому наличие мест проверяется на последнем шаге работы алгоритма, в тот момент, когда уже найден оптимальный путь проезда по остальным критериям. В подсистему взаимодействия входят:

— модуль интеграции в составе модуля поиска пути;

— сервис интеграции с внешними системами;

— эмулятор терминала «Экспресс»;

— программа контроля взаимодействия с внешними системами.

В подсистеме сначала производится поиск данных в кэше выполненных запросов. В кэш попадают результаты предыдущих запросов к внешним системам и сохраняются там установленное время. Если данные не найдены в кэше, подсистема обращается с запросом во внешние автоматизированные системы. Способ отправки запроса зависит от особенностей конкретной автоматизированной системы. Запрос может быть синхронным или асинхронным. Если запрос *синхронный*, то обслуживающий процесс выполняет его немедленно. Если применяется *асинхронный* способ взаимодействия, то запрос ставится в очередь на обслуживание. Когда в очереди появляются запросы, к работе подключается программа-сервис взаимодействия. Запрос из очереди трансформируется сервисом в запрос по протоколу, понятному внешней автоматизированной системе, и отправляется на ее сервер. Например, для взаимодействия с системой «Экспресс» используется специальный сервис — эмулятор терминала «Экспресс». Сервис преобразует запрос в пакеты BSC-3, инкапсулирует их в пакеты TCP/IP и передает на шлюз доступа в системе «Экспресс». Шлюз обменивается информацией с Хост-ЭВМ и возвращает ответ эмулятору терминала. Процесс протекает асинхронно: один запрос системы может отображаться в серию запросов-ответов между эмулятором и Хост-ЭВМ. Одновременно могут работать несколько сервисов взаимодействия, и они могут быть запущены на нескольких вычислительных машинах, просматривая единую очередь запросов.

Для получения справочной информации о возможности проезда через Интернет разработан портал доступа к информационно-справочной системе. Портал предоставляет пользователям информацию о пути проезда на железнодорожном и автобусном транспорте России, ближнего и дальнего зарубежья с учетом пересадок, осуществляя:

- поиск пунктов пересадки, когда прямого пути между пунктами нет;
- поиск маршрутов указанного вида транспорта;
- поиск путей проезда между двумя пунктами с пересадкой в указанном пункте;
- поиск интермодального пути проезда (разные виды транспорта);
- поиск пути проезда со всех возможных вокзалов города или с одного указанного;
- предоставление информации о расписании движения по станции;
- отображение найденных путей проезда на интерактивной карте;
- отображение интерактивной схемы беспересадочных маршрутов для указанной станции.

2. АЛГОРИТМЫ ПОИСКА ОПТИМАЛЬНЫХ ПУТЕЙ ПРОЕЗДА НА ПАССАЖИРСКОМ ТРАНСПОРТЕ

В разработанной информационно-справочной системе применяется алгоритм поиска оптимального пути, отличающийся от известных. Существуют два основных подхода к моделированию информации о расписаниях как задачи поиска кратчайшего пути: время-расширенный и время-зависимый [5, 6]. 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 очень большого (хотя и разреженного) графа. 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 расписании* 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 формулируется следующим образом: имеется набор маршрутов Z, набор станций B, и набор элементарных соединений C, элементы которого c — кортежи 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 во время t_d; следующая станция маршрута — S_2, прибытие — t_a. Длина элементарного соединения c обозначается $length(c)$ — время между прибытием и отправлением. Расписание действует для 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 только в том случае, если время между отправлением и прибытием на станцию S больше либо равно *минимальному времени пересадки* для этой станции (S). 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 длина $length(c)$ определена и равна времени пешего перехода.

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 от станции A не происходит раньше задан-

ного времени t_0. Критерий оптимизации — минимизировать разницу между временем прибытия и заданным временем отправления. В другой задаче *минимального числа пересадок* необходимо отыскать путь с наименьшим числом пересадок для станций A и B, вообще не принимая во внимание времена прибытий и отправлений.

Представленные подходы могут быть применены для *моделирования реальной задачи*, например, учитывающей время пересадок. Чтобы учесть времена пересадки во время-расширенной модели, на графе создается копия всех узлов прибытия и отправления со станций, 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, большим 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 пункта A в пункт B 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 числом 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) — основной задачи в области многоцелевой оптимизации [7]. Задача многокритериальной оптимизации обычно *NP*-полная, так как размер набора Парето обычно экспоненциально возрастает. Даже для очень простых графов (цепь узлов, соединенных ребрами) в задаче с двумя критериями число решений в наборе Парето растет экспоненциально. Поэтому на практике для нахождения решения многокритериальной задачи применяются приближенные подходы. Один из способов выбора Парето-оптимального решения — *лексикографический порядок*. В упрощенной версии время-расширенного подхода лексикографически первое решение может быть посчитано для любого d-значения как вес ребер. Например, если $d = 2$ и первый элемент — время в пути, а второй — число пересадок, то среди всех быстрых путей находится один с минимальным числом пересадок. В практической версии время-расширенного подхода могут быть использованы только те критерии, где первый критерий — время в пути.

Подходы, описанные для решения упрощенной задачи наискорейшего прибытия, были хорошо изучены для время-расширенной и время-зависимой моделей. Нахождение всех подходящих путей с учетом времени пути, стоимости, числа пересадок, естественно, гораздо сложнее, чем просто поиск наискорейшего пути. Исследования известных алгоритмов поиска по расписаниям показали, что без применения специальных ускоряющих техник их производительность обычно оказывается недостаточной. Средняя производительность особенно важна в системах с центральным сервером, который должен обрабатывать сотни запросов одновременно, например, в Интернете или на транспортных терминалах (вокзалах, аэропортах).

Перечислим некоторые ускоряющие техники, которые позволяют повысить производительность алгоритмов поиска. Техника *ограничения угла* использует информацию о пространственном расположении географических объектов. На этапе *препроцессинга* (предварительной обработки данных) применяется алгоритм Дийкстры, чтобы вычислить кратчайшие пути из узла до всех других станций [9]. 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 каждому ребру графа. Углы вычисляются следующим образом. 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. Нарисуем угловой сектор между углами $a(v, w)$ и $b(v, w)$ с вершиной в точке S_v. 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 [6]. В результате число рассматриваемых узлов может уменьшаться в несколько раз. Пусть $G = (V, E)$ — маршрутный граф и $V^{\sim}$ — набор выбранных узлов, из которых строится вспомогательный граф $G^{\sim}$. Вспомогательный граф $G^{\sim}$ создается и веса его ребер 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 — 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 дополнительными кратчайшими путями между определенными узлами [10]. Чтобы найти кратчайший путь между узлами, алгоритму Дийкстры достаточно рассмот-

реть 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 $r(v, P)$ достижения по отношению к пути P есть минимум от длины префикса пути P (часть пути от s к v) и длины суффикса (часть пути от v к t). Радиус $r(v)$ достижения узла v — это максимум $r(v, P)$ по всем кратчайшим путям, которые содержат вершину v. Вычислив предварительно радиус $r(v)$ для всех узлов, во время работы алгоритма можно быстро узнать, по какому ребру необходимо двигаться дальше, на ходу исключая те ребра, которые не ведут к цели.

3. ОРИГИНАЛЬНЫЙ АЛГОРИТМ ПОИСКА ОПТИМАЛЬНОГО ПУТИ НА ПАССАЖИРСКОМ ТРАНСПОРТЕ

Оригинальный алгоритм поиска, который применяется в информационно-справочной системе, позволяет строго решать двухкритериальную задачу поиска пути с минимальным числом пересадок и с наискорейшим временем прибытия [1, 2]. Алгоритм оперирует со специально преобразованными исходными данными и позволяет решить задачу точно, не сводя ее к эвристическим методам. При этом поиск производится не по реальному графу, а по «виртуальному», ребра которого вычисляются во время поиска. Преимущество состоит в том, что в оригинальной реализации не требуется полный препроцессинг при частичном изменении данных о расписании движения.

Для ускорения оригинального алгоритма известные техники, например, ограничения угла, не могут быть применены, так как информация о ребрах графа вообще не хранится в системе. А алгоритм Дийкстры, хотя и может быть применен, но недостаточно быстрый для практической реализации в информационно-справочной системе. Поэтому для ускорения работы алгоритма предложены следующие специальные техники.

Метод *пересадочного подграфа* заключается в выделении потенциальных узлов пересадки. В процессе загрузки исходных данных о расписаниях можно анализировать указанные в расписании маршрута станции и определять потенциальные узлы пересадки. Разработаны несколько критериев, на основании которых станция добавляется в пересадочный подграф. Техника пересадочного подграфа схожа с техникой иерархического поиска, но, в отличие от последней, пересчет пересадочного подграфа производится быстро. Еще один эффективный метод — *балансировка двунаправленным поиском* — заключается в особом выборе прямого или обратного поиска на каждом шаге итерации. Разработан также метод *ускорения последней итерации*. 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 можно исключить. Еще один эффективный метод ускорения, хотя и не точный — это метод *ускорения* A^*, основанный на эвристическом алгоритме A^*. Хотя в данной задаче он не является методом, сохраняющим расстояния, но на практике почти всегда позволяет находить оптимальный путь.

ЗАКЛЮЧЕНИЕ

Рассмотренные подход и техники ускорения положены в основу разработанной информационно-справочной системы. Ее внедрение позволит резко повысить качество информационного обслуживания пассажиров, оптимизировать пассажиропотоки и загрузку пассажирских транспортных средств.

ЛИТЕРАТУРА

1. *Вишневский В.М., Железов Р.В., Атанасова Т.Н.* «Единая справочная система на пассажирском транспорте Российской Федерации», Distributed Computer and Communication Networks. — М.: Техносфера, 2005. — С. 165—172.
2. *Железов Р.В.* Реализация единой справочной системы пассажирского транспорта на базе технологий Microsoft // Сб. тр. конф. «Технологии Microsoft в теории и практике программирования». — М.: МГТУ им. Н.Э. Баумана, 2006. — С. 10—13.
3. *Вишневский В.М.* Теоретические основы проектирования компьютерных сетей. — М.: Техносфера, 2003. — 512 с.
4. *HAFAS.* A timetable information system by HaCon Ingenieurgesellschaft mbH, Hannover, Germany [Электронный ресурс]. — Режим доступа: http://www.hacon.de/hafas/
5. *Timetable* Information: Models and Algorithms in Algorithmic Methods for Railway Optimization / Muller-Hannemann M., et al. Berlin / Heidelberg: Springer, 2007.
6. *Brodal G.S. and Jacob R.* Time-dependent networks as models to achieve fast exact time-table queries. — In Proc. of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003). — Vol. 92 of Electronic Notes in Theoretical Computer Science. Elsevier, 2004.
7. *Cooke K. L. and Halsey E.* The shortest route through a network with time-dependent internodal transit times // Journal of Math. Analysis and Appl. — 1966. — Vol. 14. — P. 493—498.
8. *Wagner D. and Willhalm T.* Geometric speed-up techniques for finding shortest paths in large sparse graphs. — Proc. of the 11th European Symposium on Algorithms (ESA 2003). — Springer, 2003. — Vol. 2832 of LNCS. — P. 776—787.
9. *Wagner D. and Willhalm T.* Speed-up techniques for shortest path computations // Algorithmic Methods for Railway Optimization, LNCS. Springer.
10. *Schulz F., Wagner D., and Zaroliagis C.* Using multi-level graphs for timetable information in railway systems / Proc. 4th Workshop on Algorithm Engineering and Experiments (ALENEX). — Springer, 2002. — Vol. 2409 of LNCS. — P. 43—59.
11. *Железов Р.В.* Особенности алгоритма поиска маршрута на транспорте с учетом расписаний движения // Тр. междунар. сем. «Распределенные компьютерные и телекоммуникационные сети (DCCN`2007)» / ИППИ РАН. — Москва, РАН, 2007. — С. 28— 32.

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

Вишневский Владимир Миронович — д-р техн. наук, зав. отделом, Институт проблем передачи информации им. А.А. Харкевича РАН, ☎ (495) 699-29-04, e-mail: vishn@iitp.ru,

Железов Роман Владимирович — аспирант, Московский физико-технический институт, г. Долгопрудный, e-mail: ironromeo@mail.ru.