МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Т. Ф. ГОРБАЧЕВА»

Институт информационных технологий, машиностроения и автотранспорта

Кафедра прикладных информационных технологий

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Зав. кафедрой, доцент, к.т.н.

И.А. Соколов

«

»

2015 г.

Гарченко Елизавета Владимировна

**РАЗРАБОТКА, АНАЛИЗ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОИСКА ОПТИМАЛЬНЫХ МАРШРУТОВ ДВИЖЕНИЯ ТРАНСПОРТА В УЛИЧНО-ДОРОЖНОЙ СЕТИ ГОРОДА**

Выпускная работа бакалавра

по направлению подготовки 09.03.03 «Прикладная информатика»

|  |  |  |
| --- | --- | --- |
| Научный руководитель, |  |  |
| ст. преподаватель |  | В.С. Дороганов |
|  |  |  |
| Исполнитель, |  |  |
| студ. гр. ПИб-111 |  | Е.В. Гарченко |
|  | | | | |

Электронная версия выпускной работы бакалавра помещена в электронную библиотеку

Файл:

Администратор:

Кемерово – 2015

**РЕФЕРАТ**

Бакалаврская работа, 28 страниц, 24 рисунка, 26 источников, 2 приложения.

АЛГОРИТМЫ НА ГРАФАХ, ПОИСК ПУТЕЙ, ОПТИМИЗАЦИЯ, БАЗА ДАННЫХ, ПРОГРАММИРОВАНИЕ, ГЕОИНФОРМАЦИОННАЯ СИСТЕМА, АНАЛИЗ.

**Объект исследования:** улично-дорожная сеть города.

**Предмет исследования:** алгоритмы поиска оптимальных маршрутов движения.

**Цель работы:** исследование эффективности алгоритмов поиска оптимальных маршрутов движения транспорта в улично-дорожной сети города.

**Методы и технологии разработки:** анализ алгоритмов поиска на графах; технология автоматизированных баз данных; визуальное и объектно-ориентированное программирование. Написание клиент-серверного программного продукта было произведено в интегрированной среде разработки Visual Studio 2012, исходный код написан на объектно-ориентированном языке программирования Visual C#. Для хранения информации, используемой при работе с системой, используется база данных Microsoft SQL Server.

**Результаты работы:** проведен анализ алгоритмов поиска на графах. Разработана геоинформационная система анализа эффективности работы алгоритмов поиска оптимального пути в улично-транспортной сети города.

**Область применения:** применение в отделах логистики предприятий.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc423520098)

[1. ЗАДАЧА ЛОГИСТИКИ В УСЛОВИЯХ ГОРОДА 5](#_Toc423520099)

[1.1. Основы транспортной логистики 5](#_Toc423520100)

[1.2. Проблематика в условиях города 5](#_Toc423520101)

[1.3. Задача оптимального пути в условиях города 6](#_Toc423520102)

[2. АЛГОРИТМЫ 8](#_Toc423520103)

[2.1. Основы теории графов 8](#_Toc423520104)

[2.2. Поиск на графах 9](#_Toc423520105)

[2.2.1. Поиск в глубину 9](#_Toc423520106)

[2.2.2. Поиск в ширину 10](#_Toc423520107)

[2.2.3. Алгоритм Дейкстры 11](#_Toc423520108)

[2.2.4. Алгоритм Беллмана-Форда 12](#_Toc423520109)

[2.2.5. Алгоритм А\* 12](#_Toc423520110)

[2.2.6. Муравьиный алгоритм 13](#_Toc423520111)

[3. РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ АНАЛИЗА ЭФФЕКТИВНОСТИ РАБОТЫ АЛГОРИТМОВ 15](#_Toc423520112)

[3.1. Среда и средства разработки 15](#_Toc423520113)

[3.2. Информационная система 16](#_Toc423520114)

[3.2.1. Структура системы и базы данных 16](#_Toc423520115)

[3.2.2. Интерфейс и функциональные возможности 17](#_Toc423520116)

[3.2.3. Методики поиска оптимальных путей 18](#_Toc423520117)

[ЗАКЛЮЧЕНИЕ 24](#_Toc423520118)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc423520119)

[ПРИЛОЖЕНИЕ А АКТ ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ 27](#_Toc423520120)

[ПРИЛОЖЕНИЕ Б СПИСОК ФАЙЛОВ НА КОМПАКТ-ДИСКЕ 28](#_Toc423520121)

ВВЕДЕНИЕ

Дороги являются важной частью инфраструктуры государства. Изначально они сооружались для военных целей, обеспечивая мобильность и быстроту реагирования армии. Данные плюсы не остались незамеченными купцами, и вскоре торговля стала постоянным пользователем данных путей сообщения.

В современных реалиях большую социально-экономическую роль играют именно автомобильные дороги. На сегодня в России понятие автомобильных дорог закреплено на законодательном уровне. Согласно Федеральному Закону №257-ФЗ ст.3: «Автомобильная дорога - объект транспортной инфраструктуры, предназначенный для движения транспортных средств и включающий в себя земельные участки в границах полосы отвода автомобильной дороги и расположенные на них или под ними конструктивные элементы (дорожное полотно, дорожное покрытие и подобные элементы) и дорожные сооружения, являющиеся ее технологической частью, - защитные дорожные сооружения, искусственные дорожные сооружения, производственные объекты, элементы обустройства автомобильных дорог.» [1].

По данным 2014 года Россия занимает пятое место по протяженности дорог (1 396 000 км) [2]. При этом территория страны составляет 17 125 407 км². Таким образом, на 1 км² приходится 0,08 км дорожного полотна, в то время как в Бразилии, находящейся на четвертом месте в данном рейтинге, этот коэффициент составляет 4,86. Хорошо это или плохо, но как бы то ни было в России имеет место быть низкий уровень развития дорожной сети. Строительство дорог идет недостаточно высокими темпами за счет высокой стоимости и малой длительности сезона, пригодного для проведения работ, и дорожно-транспортные сети города не справляются с постоянно возрастающей на них нагрузкой. Помимо этого, в рамках города дорожная сеть не всегда подлежит расширению из-за его архитектуры. Поэтому высока актуальность анализа, разработки и программной реализации алгоритмов, способных в заданной улично-дорожной сети города определять оптимальные с различных точек зрения маршруты движения.

Таким образом, *объектом* исследования данной работы является улично-дорожная сеть города, а *целью* – исследование эффективности алгоритмов поиска оптимальных маршрутов движения транспорта в улично-дорожной сети города. Для достижения этой цели были поставлены следующие задачи:

1. изучить проблемы логистики в условиях города
2. проанализировать алгоритмы поиска на графах
3. выбрать средства реализации
4. разработать информационную систему, способную осуществлять поиск, оптимизацию и визуализацию маршрутов, а также предоставлять информацию о работе алгоритмов.

Бакалаврская работа состоит из введения, трех глав, заключения, списка использованных источников и приложения. Во введении обоснована актуальность, обозначены объект и предмет и сформулированы цель и задачи исследования. Первая глава посвящена задаче логистики в условиях города Кемерово. Во второй главе рассмотрены теоретические основы работы: проанализированы алгоритмы поиска на графах, описаны методики. В третьей главе содержится информация о средствах реализации информационной системы, а так же описание разработанной информационной системы и базы данных. В заключении обобщаются результаты проведенного исследования.

1. ЗАДАЧА ЛОГИСТИКИ В УСЛОВИЯХ ГОРОДА
   1. Основы транспортной логистики

В России сложилась мощная транспортная система, в которую входят различные виды транспорта: железнодорожный, морской, речной, автомобильный, воздушный и трубопроводный. Согласно современному экономическому словарю: «Логистика - часть экономической науки и область деятельности, предмет которых заключается в организации и регулировании процессов продвижения товаров от производителей к потребителям, функционирования сферы обращения продукции, товаров, услуг, управления товарными запасами, создания инфраструктуры товародвижения» [3]. Большая часть логистических операций осуществляется с помощью различных транспортных средств. Транспорт же является ключевым связующим звеном между элементами логистических систем.

Таким образом, транспортная логистика - это перемещение нужного количества товара в конечную точку, оптимальным маршрутом за требуемое время и с наименьшими издержками. Одной из задач транспортной логистики является создание транспортных систем, в частности транспортных коридоров.

* 1. Проблематика в условиях города

Рассмотрим современные реалии города Кемерово: ежедневно среднестатистический житель совершает не менее двух поездок по городу. И если раньше это не было такой большой проблемой и занимало не так много времени, то теперь ситуация сложилась иначе. По данным Федеральной службы государственной статистики, за последние 13 лет автопарк собственных автомобилей в России вырос более чем в два раза (Рис. 1), при этом количество эксплуатационных автобусов общего пользования уменьшилось примерно в этой же пропорции (Рис. 2) [4].

Рис. - Динамика собственных легковых автомобилей на 1000 населения

Рис. - Динамика эксплуатационных автобусов общего пользования на 100 000 населения

Можно наблюдать определенную зависимость, часть людей пересаживается с автобусов общего пользования на личные автомобили, и, как следствие, наблюдается тенденция спада популярности пользования общественным транспортом. Вроде бы все закономерно, но дороги в рамках населенного пункта по большей части остались теми же, с той же пропускной способностью, что является причиной такого явления как автомобильные пробки в час пик движения. Таким образом, на примере региона можно сделать вывод, что основной проблемой развития транспортной логистики в России, касательно автомобильных путей, является слаборазвитая инфраструктура автомобильных дорог, в частности их низкий технико-технологический уровень [5].

* 1. Задача оптимального пути в условиях города

С транспортной логистикой неотъемлемо связана одна из важнейших классических задач теории графов – задача о кратчайшем пути. В данной задаче минимизируется сумма весов ребер, которые составляют путь. Любую транспортную систему можно представить в виде ориентированного графа, так же следует отметить, что в роли веса ребра в графе может выступать не только расстояние, но и цена перевозки, время, или любой другой критерий, являющийся целевым для пользователя. Современный темп жизни сделал время еще более ценным ресурсом, чем прежде. Таким образом, наиболее актуальными задачами с большой аудиторией пользователей являются поиск оптимального пути по критерию расстояния и по критерию времени.

Критерий расстояния является статическим, что позволяет, зная координаты начальной и конечной точки, а так же формулу гаверсинусов, найти расстояние с очень низкой погрешностью, которое и будет являться весом ребер графа. Для решения задачи по критерию времени в роли веса ребра будет выступать время, затраченное на перемещение из начальной точки в конечную.

Согласно приказу Минтранса РФ от 9 марта 2010 года №55 «Об утверждении перечня видов автомобильных транспортных средств, используемых для перевозки пассажиров и опасных грузов, подлежащих оснащению аппаратурой спутниковой навигации ГЛОНАСС или ГЛОНАСС/GPS» все транспортные средства (ТС) отвечающие за перевозку пассажиров должны быть оснащены навигационным оборудованием [6]. Информация о движении пассажирского транспорта в г. Кемерово предоставляется ООО «Кемгортранс» и открыта для просмотра. Именно эта информация позволяет собирать информацию о скорости движения на отдельных участках дорожно-транспортной сети города. Для данной работы были выбраны следующие маршруты общественного транспорта: 03, 13т, 35т, 40т, 43т. Эти маршруты имеют насыщенное расписание, приемлемую скорость движения и значительную зону покрытия целевого участка дорожно-транспортной сети города, что позволяет проанализировать загруженность основных магистралей г. Кемерово (Рис. 3).

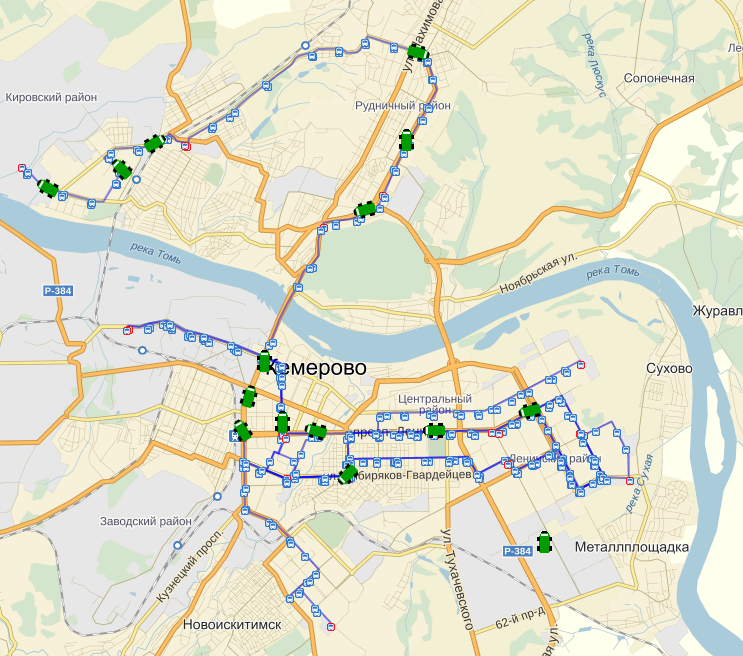
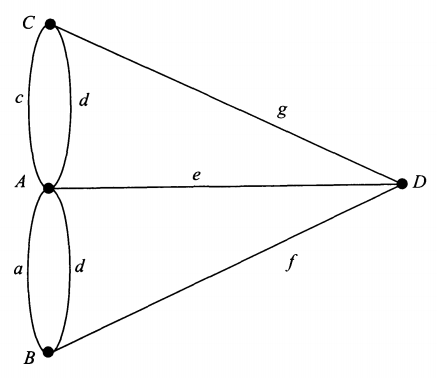


Рис. - Схема движения выбранных маршрутов с отображением местоположения транспортных средств в системе Кемгортранс

Таким образом, зная маршрут движения и осуществляя запрос о местоположении целевых транспортных средств с периодичностью 30 секунд, можно вычислить скорость движения транспортного средства на определенном участке с высокой точностью, что позволяет осуществлять анализ состояния большей части дорожно-транспортной сети города на наличие автомобильных заторов [7].

1. АЛГОРИТМЫ
   1. Основы теории графов

Теория графов является разделом дискретной математики, изучающим свойства графов. Впервые графы появились в XVIII веке, когда математик Леонард Эйлер пытался решить задачу о Кенигсбергских мостах, которая стала в последствии классической задачей теории графов. Задача состоит в том, чтобы во время прогулки по городу пройти по каждому мосту только один раз и в конце вернуться в исходную точку. Эйлер изобразил Кенигсберг в виде графа, приняв части города за вершины, а мосты – за ребра (Рис. 4). Таким образом Эйлеру удалось доказать, что оптимального маршрута обхода города не существует [8].



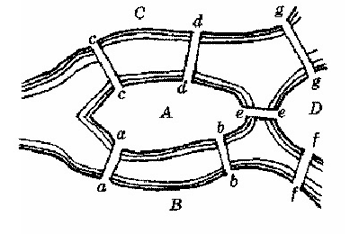


Рис. - Схема Кенигсберга XVIII века и модель задачи

*Графом* *G* называется пара (*V(G)*, *E(G)*), где *V(G)* – непустое конечное множество элементов, называемых вершинами, а *E(G)* – конечное множество неупорядоченных пар элементов из *V(G)*, называемых ребрами (Рис. 5, а) [9]*.*

Граф называется *планарным*, если его можно уложить на плоскости так, чтобы никакие два его ребра не пересекались. *Плоский* *граф* – это граф, уже уложенный на плоскости [10].

*Ориентированным графом* или *орграфом D* называется пара (*V(D)*, *A(D)*), где *V(D)* – непустое конечное множество вершин, а *A(D)* – конечное множество упорядоченных пар элементов из *V(D),* называемых дугами (Рис. 5, б) [9].

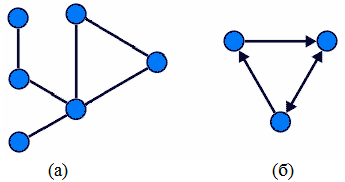


Рис. – а) - граф, б) - ориентированный граф

Две вершины *j* и *k* графа называются *смежными*, если они соединяются ребром {*j, k*}, при этом вершины будут называться *инцидентными* этому ребру, а ребро-этим вершинам. *Степенью* вершины называется число ребер, инцидентных этой вершине. Вершину степени *0* называют *изолированной* [9].

*Путь* в ориентированном графе – это последовательность вершин, в которой существует ребро, указывающее из каждой вершины на следующую вершину последовательности. *Длина* пути равна количеству ребер в нем [11].

Теория графов находит свое место в различных областях науки. В химии это молекулярный граф, используемый для описания структур, в информатике и программировании – граф-схема алгоритма. В схемотехнике топология межсоединений на плате или микросхеме представляет собой граф. Так же теория графов находит свое применение в экономике, в виде сетевого планирования и управления, а так же в транспортной логистике.

Для представления орграфов можно использовать различные структуры данных. Выбор структуры зависит от действий, которые будут совершаться над вершинами и дугами графа.

Одной из основных структур является матрица смежности. *Матрица смежности* для орграфа *G* - это матрица *A* размера *n* x *n* со значениями булевого типа, где *A[i, j] = true*, если существует дуга из вершины *i* в вершину *j*. Так же в ячейках матрицы смежности могут находиться метки дуг, тогда *A[i, j] = 3* будет означать, что дуга *i → j* имеет метку *3.* Если же таковой дуги не существует, то ячейка *A[i,* *j]* может рассматриваться как пустая. Основной недостаток матриц смежности заключается в том, что они требуют *ϴ(V2)* памяти, где *V* – количество вершин графа [12].

Чтобы уменьшить количество памяти, необходимой для хранения данных, применяют представление посредством *списков смежности*. Списком смежности для вершины *i* называется список всех вершин, смежных с данной вершиной и упорядоченный определённым образом. Такое представление орграфа требует для хранения объем памяти, пропорциональный сумме количества вершин и дуг: *ϴ(V+E)* [12].

* 1. Поиск на графах

В задаче о кратчайшем пути из одной вершины в другую дан ориентированный граф *G(V, E),* весовая функция которого сопоставляет каждому ребру графа некоторый вес. Весом пути же будет являться сумма весов ребер, входящих в этот путь. Задача состоит в том, чтобы среди всех возможных путей, соединяющих две вершины, найти такой путь, вес которого будет минимален – это и будет являться кратчайшим путём [13].

* + 1. Поиск в глубину

Поиск в глубину (depth-first search, DFS, бэктрекинг) позволяет определить, какие вершины могут быть достигнуты из заданной. Идея алгоритма состоит в том, чтобы идти вперед, пока это возможно. Алгоритм можно описать рекурсивно, без явного использования стека: при поиске в глубину исследуются все ребра, выходящие из последней открытой вершины, а возврат происходит в тот узел, из которого была открыта текущая вершина. Этот процесс повторяется до тех пор, пока не будут открыты все вершины графа, или же не найдена целевая вершина. Тем не менее, на больших графах поиск в глубину серьезно нагружает стек вызовов. Что бы избежать переполнения стека, используют нерекурсивный вариант поиска (Рис. 6). Существуют три состояния вершин: белая (новая), серая (открытая) и черная (закрытая). Эти условия позволяют избежать повторного исследования узлов процедурой *dfs* [13].



Рис. - Алгоритм поиска в глубину

Следует помнить, что поиск в глубину не дает кратчайшего пути на графе, поскольку список элементов организован по принципу LIFO (last in – first out “последним пришел – первым вышел”) и не предполагает какой-либо сортировки по критерию расстояния. Такая структура хранения вершин называется стеком. Алгоритм поиска в глубину применяется в топологической сортировке, для задачи поиска пути в лабиринте, осмотра директорий в файловой системе.

Время работы данного алгоритма с использованием списков смежности составляет *O(V+E)*, где *V* – количество вершин, а *E –* количество ребер, а с использованием матрицы смежности – *O(V2)*.

* + 1. Поиск в ширину

Поиск в ширину (breadth-first search, BFS) – один из простейших алгоритмов обхода графа. Является основой для многих других, например алгоритм Дейкстры использует схожую концепцию. Поиск в ширину отличается от поиска в глубину порядком исследования вершин, который зависит от структуры, которая используется для хранения открытых вершин. В отличии от поиска в глубину, данные хранятся не в стеке, а в очереди. Таким образом, при обходе в ширину мы исследуем сначала самые старые отрытые вершины, а затем плавно расширяем область исследования по направлению от стартовой вершины [14]. Алгоритм (Рис. 7) систематически обходит все ребра графа до открытия всех вершин, достижимых из заданной начальной вершины. Поиск в ширину заслужил свое название тем, что при обходе графа, прежде чем приступить к исследованию вершин, находящихся от целевой на расстоянии *k+1*, нужно выполнить обход всех вершин, находящихся на расстоянии *k*. Так же, как и при поиске в глубину, во время поиска в ширину вершины имеют три состояния белая (новая), серая (открытая) и черная (закрытая).



Рис. - Алгоритм поиска в ширину

Поиск в ширину позволяет найти путь, содержащий в себе наименьшее количество ребер.

Обход в ширину с использованием списков смежности исполняется за время *O(V+E)*, где *V* – количество вершин, а *E –* количество ребер, а с использованием матрицы смежности – *O(V2)*.

* + 1. Алгоритм Дейкстры

Алгоритм Дейкстры является «жадным» алгоритмом, способным находить кратчайший путь от заданной вершины ко всем другим вершинам графа, включая требуемую вершину *t*. Алгоритм работает поэтапно, находя на каждой итерации кратчайший путь из начальной вершины к новой. Существует множество вершин *S*, для которых уже вычислено минимальное расстояние из начальной вершины. В начале работы алгоритма расстояние до всех вершин из начальной принимается за положительную бесконечность, а затем поочередно выбирается вершина *uϵV-S,* которая имеет минимальную метку на данном этапе. Затем вершина добавляется в множество *S* и производится релаксация всех исходящих из нее ребер. Такие действия продолжаются, пока не будет найдена целевая вершина [15]. Следует помнить, что алгоритм Дейкстры применим для ориентированных графов, в которых отсутствуют отрицательные веса. Данный алгоритм широко применяется в программировании, например в протоколе динамической маршрутизации OSPF и протоколе внутренних шлюзов IS-IS, который используется в основном в крупных сетях провайдеров или особо крупных корпоративных сетях.

Задача для данного алгоритма может быть сформулирована следующим образом: имеется сеть автомобильных дорог Кемеровской области. Найти все кратчайшие пути от Кемерово до каждого населенного пункта. Движение возможно только по дорогам.

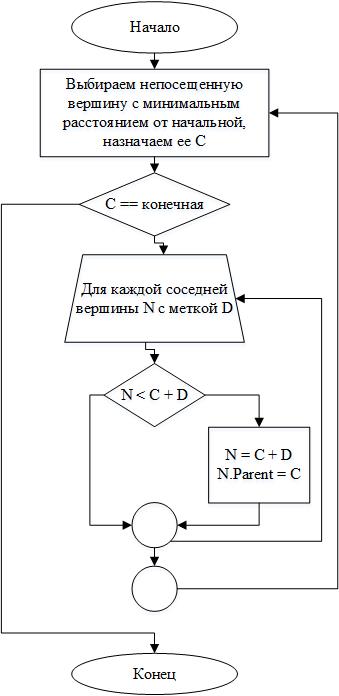


Рис. - Алгоритм Дейкстры

Время работы алгоритма при использовании неубывающей очереди с приоритетами равняется *O(V2+E) = O(V2).*

* + 1. Алгоритм Беллмана-Форда

Если в графе имеют место быть ребра с отрицательными весами, то алгоритм Дейкстры может дать неверный результат. Подобно алгоритму Дейкстры, имеет место процедура релаксации. Но в алгоритме Беллмана-Форда в данной процедуре происходит однократное ослабление каждого ребра графа. Для решения задачи нахождения кратчайшего пути из одной вершины во все, данная процедура производится *V-1* раз [15].

Алгоритм маршрутизации RIP (алгоритм Беллмана–Форда) был впервые разработан в 1969 году, как основной для сети ARPANET. Так же, нужно заметить, что кратчайших путей может и вовсе не существовать. Например, в графе, содержащим цикл с отрицательным суммарным весом, существует короткий путь от одной вершины этого графа до другой, так как каждый обход отрицательного цикла уменьшает длину пути.

Время работы алгоритма составляет *O(VE).* Для выяснения наличия отрицательного цикла в графе, нужно производить релаксацию всех ребер до тех пор, пока все они не будут ослаблены. Это занимает время *O(V).* Если же цикл с отрицательным весом имеет место быть, то он не может состоять более чем из *E* ребер, так, время его трассировки - *O(E)* [16].

Так как в рамках данной работы используется граф, в котором заведомо отсутствуют отрицательные ребра, данный алгоритм не был программно реализован.

* + 1. Алгоритм А\*

Одной из разновидностей поиска по первому наилучшему совпадению является поиск А\* (A star, А звезда). В нем применяется оценка узлов, объединяющая в себе стоимость достижения данного узла *g(n)* и стоимость прохождения из текущего узла до целевого *h(n)* : *f(n) = g(n) + h(n).* Таким образом, можно сказать что *f(n)* является оценкой наименее дорогостоящего пути, проходящего через узел *n*. Работа алгоритма построена таким образом, что из всех доступных на текущем шаге узлов первым проверяется тот, который обладает наименьшей оценкой *f(n)* [17].

В качестве *h(n)* может выступать любая формула по определению расстояния между двумя точками по прямой как на плоскости, так и на эллипсоиде. Например, формула гаверсинусов (1), которая позволяет получать расстояние между точками с низкой погрешностью на небольших расстояниях.

(1)

*∆σ* –угловая разница;

*φ1, α1 ;φ2 α2* – широта и долгота двух точек в радианах;

*∆α* – разница координат по долготе.

В простейшем же варианте задачи на плоскости, в качестве *h(n)* может выступать формула длины вектора (2).

(2)

*A(a1; b1), B(a2; b2)* – координаты начала и конца вектора.

Временная сложность алгоритма А\* зависит от эвристики. Доказано, что экспоненциальный рост происходит в том случае, если ошибка эвристической функции *h(n)* растет не быстрее логарифма фактической стоимости пути. В математических обозначениях данное условие будет выглядеть так: *|h(n) – h\*(n)| ≤ O(log h\*(n))* [17]*.*

* + 1. Муравьиный алгоритм

Основная идея муравьиного алгоритма состоит в том, чтобы сымитировать поведение муравьев. Данный алгоритм относят к так называемым «естественным алгоритмам», к которым так же можно отнести алгоритм имитации отжига, алгоритм поведения роя пчел.

Что бы понять идею данного алгоритма, нужно немного углубиться в биологию и понять, чем же так примечательны муравьи. Муравьи для общения используют два способа: прямой и непрямой. К прямому способу можно отнести обмен пищей, визуальный контакт. К непрямому относится стигмержи, тип взаимодействия, когда субъект изменяет часть окружающей среды, а остальные используют информацию о ее состоянии. У муравьев этот тип взаимодействия осуществляется посредством специального секрета, выделяемого насекомым при передвижении – феромона. Феромон - достаточно стойкое вещество, и именно он является своеобразным дорожным полотном муравьиных троп. Иными словами, чем больше муравьёв ходит по одной тропе, тем сильнее на ней плотность феромона (Рис. 9) [18].

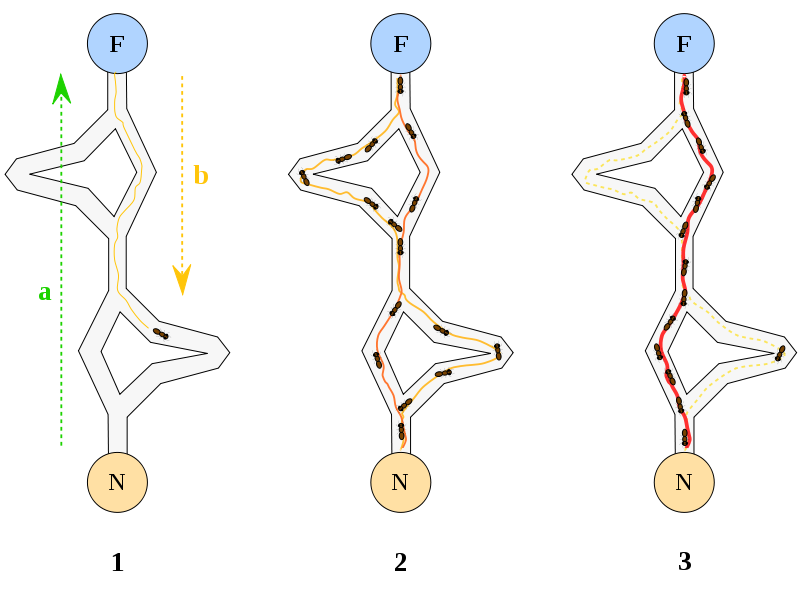


Рис. - Схематическое представление поведения муравьев

С математической точки зрения, в этом алгоритме выбором следующей вершины к посещению управляет случайная функция, которая определяет с большей вероятностью такую вершину *j*, в которой будет наибольшее значение функции *Pij,k* (где *i* – вершина в которой мы находимся, *j* – вершина, которую рассматриваем для посещения, а *k* – номер муравья) (3) [19].

(3)

– количество феромона на ребре *i, j*;

α – параметр, контролирующий влияние количества феромонов;

*–* привлекательность ребра *i, j* (начальное значение обычно *1/di,j*, где *d* - расстояние);

*–* параметр, контролирующий влияние привлекательности ребра*.*

Так же, как и настоящий феромон, феромон в алгоритме должен «испаряться», чтобы не пропустить оптимальное решение. Для этого вводится коэффициент *p* в следующую итеративную формулу, применяемую после каждого обхода графа (4) [19].

(4)

– количество феромона на ребре *i, j*;

*p* – скорость испарения феромона.

1. РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ АНАЛИЗА ЭФФЕКТИВНОСТИ РАБОТЫ АЛГОРИТМОВ
   1. Среда и средства разработки

Для создания информационной системы был использован язык **Visual C#**, который относится к пакетам интегрированной среды разработки (ИСР) **Microsoft Visual Studio 2012**. К преимуществам использования данного языка можно отнести:

* объектная ориентированность;
* большое количество библиотек и насыщенная документация;
* встроенный обработчик событий;
* возможность работать с памятью напрямую;
* работа в исполняемой среде .NET;
* удобный пользовательский интерфейс конструкторов;
* автоматическая «сборка мусора» [20].

Visual C# позволяет создавать обычные приложения Windows, XML-веб-службы, клиент-серверные приложения, приложения по работе с базами данных и т.д.

Использовалась платформа пользовательского интерфейса **Windows Presentation Foundation** (WPF). WPF имеет множество возможностей разработки: элементы управления, ресурсы, графику, привязки данных и т.д. С помощью WPF можно создавать как автономные приложения, так и приложения, запускаемые в браузере. В основе платформы лежит система векторной визуализации, не зависящая от разрешения устройства вывода. В отличии от Windows Forms, которая использует GDI/GDI+, графической технологией, лежащей в основе WPF, является DirectX [21].

**DirectX** – это набор API, разработанных для решения задач, связанных с программированием, и наиболее широко используется при разработке компьютерных игр. Так как DirectX передает большую часть работы GPU, будь то визуализация текста или сложной модели, наблюдается повышение производительности за счет аппаратного ускорения. Все это позволяет создавать более продвинутые пользовательские интерфейсы, которые не были возможны в Windows Forms, например видеофайлы, 3D содержимое [22].

Декларативное программирование подразумевает собой описание данных, а не написание кода с целью заставить компонент выполнить какие-либо действия. Для обеспечения декларативной модели WPF использует язык XAML [23].

**XAML** (*e****X****tensible* ***A****pplication* ***M****arkup* ***L****anguage*) – это расширяемый язык разметки для приложений, разработанный Microsoft. Позволяет создать видимые элементы интерфейса в разметке XAML, а затем отделить пользовательский интерфейс от логики, используя файлы программной части. Язык позволяет нескольким участникам разрабатывать пользовательский интерфейс и логику приложения, используя различные средства [24].

В качестве источника данных используется система анализа и управления реляционными базами данных **Microsoft SQL Server,** котораяявляется частью программного комплекса. MS SQL Server является надежной базой данных, подходящая для множества задач, включая многопользовательские. Обеспечивается защита от несанкционированного доступа за счет интеграции сетевой безопасности с сервером безопасности. Имеет такие возможности как: создание баз данных, работа с базой данных, резервное копирование и восстановление, система репликации, система транзакций, реляционную подсистему анализа, оптимизации и выполнения запросов клиенту [25].

* 1. Информационная система
     1. Структура системы и базы данных

Система представляет собой двухуровневую клиент-серверную архитектуру, так называемый «толстый клиент» (Рис. 10).

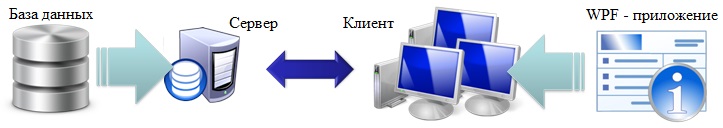


Рис. - Двухуровневая архитектура

**Клиент** – первый уровень, содержащий в себе само WPF-приложение. Уровень напрямую связан с базой данных и содержит всю логику информационной системы.

**Сервер баз данных MS SQL** – обеспечивает хранение данных и является вторым уровнем. Это объектно-реляционная система управления базами данных. Здесь хранится вся информация. База данных состоит из 17 таблиц (Рис. 11). Некоторые поля и таблицы не используются в данной работе, так как эта система является частью программного комплекса.

К группе 1 относятся таблицы, содержащие информацию о точках и локациях. В группе 2 присутствуют таблицы, содержащие статистические данные. В группе 3 находятся таблицы, содержащие информацию о транспортных средствах и остановках.

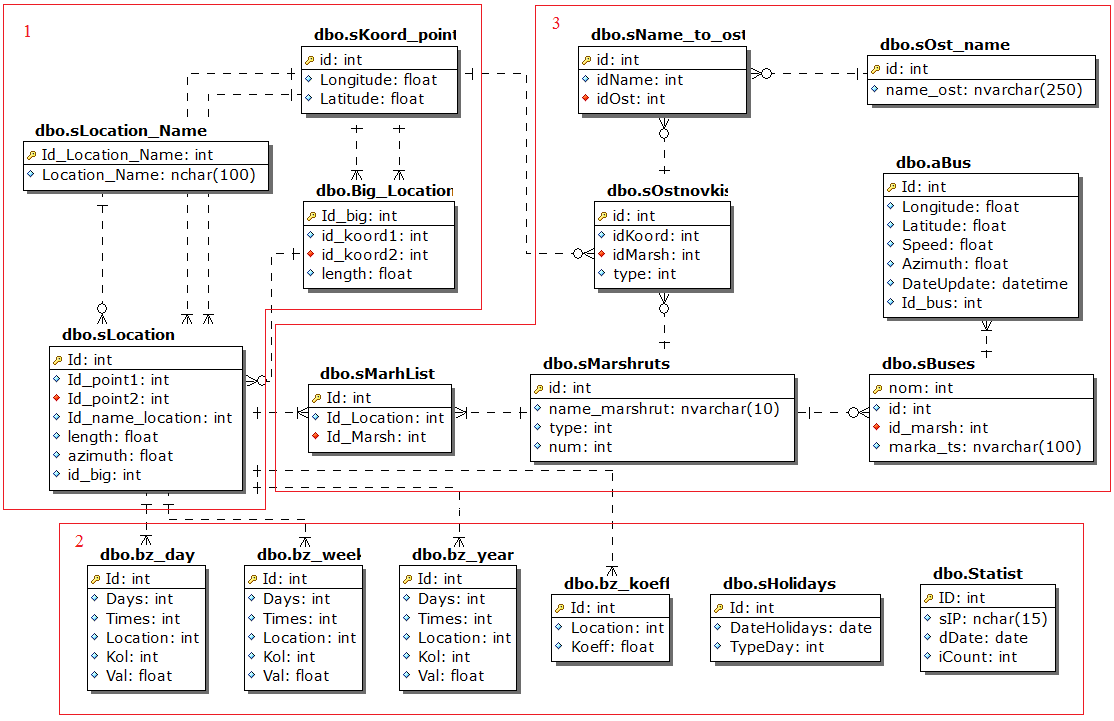


Рис. - Схема базы данных

* + 1. Интерфейс и функциональные возможности

Интерфейс информационной системы построен с использованием возможностей WPF. Преимуществами данной платформы является аппаратное ускорение, позволяющее реализовать функционал работы с картами без потери производительности.

*Локация* – вектор на карте, соединяющий две соседние точки.

*Большая локация* – вектор на карте, содержащий в себе все локации между двумя соседними перекрестками.

При запуске программы появляется основное окно, содержащее строку состояния, карту города Кемерово в виде плана, а так же функциональную панель (Рис. 12).

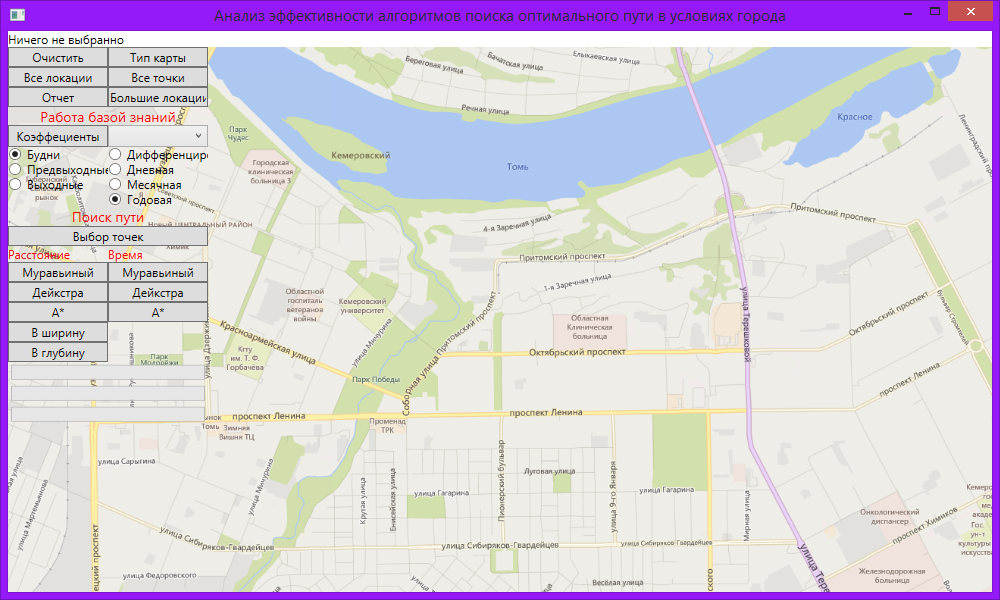


Рис. - Запуск программы

Первый блок функциональной панели содержит следующие функции:

* «Очистить» - удаляет все точки и локации на карте;
* «Тип карты» - позволяет сменить представление карты: в виде схемы или изображения со спутника;
* «Все локации» - отображает все малые локации, имеющиеся в базе данных;
* «Все точки» - отображает все точки, имеющиеся в базе данных;
* «Отчет» - выводит окно отчета о результате работы алгоритмов, запущенных за сеанс, имеет возможность построить выбранные пути на карте;
* «Большие локации» - отображает все большие локации, имеющиеся в базе данных.

Второй блок функциональной панели содержит функции по работе с накопленной статистикой о состоянии дорог в интересующий момент времени:

* ComboBox позволяет выбрать временной срез. Предложено ХХ вариантов с ХХ утра до ХХ вечера с интервалом 10 минут;
* Первый столбец RadioButton позволяет выбрать интересующий тип дня;
* Второй столбец RadioButton позволяет выбрать тип статистики для отображения.

Последний блок содержит функции для работы с маршрутами:

* «Выбор точек» - позволяет выбрать на карте две точки, между которыми необходим найти путь;
* Два столбца функциональных кнопок позволяют найти маршрут по выбранной методике и критерию;
* Три прогресс-бара для отображения состояния системы.
  + 1. Методики поиска оптимальных путей

Система обладает функционалом по поиску оптимальных путей как с точки зрения расстояния, так и с точки зрения времени. На Рис. 13 представлен отчёт о работе алгоритмов по критерию расстояния. По итогам отчета можно сделать вывод, что для нахождения оптимального маршрута при текущих условиях, алгоритм А\* (см. гл. 2.2.5) даёт наилучшее решение. Хотя алгоритм Дейкстры, описанный в главе 2.2.3 и завершил свою работу с аналогичным результатом, время, затраченное им на поиск пути намного превосходит результат А\* и даже муравьиного алгоритма (см. гл. 2.2.6), который по результатам тестирования в большей части имел худший результат работы. Поиск в глубину (см. гл. 2.2.1) показал наименьшее время и не самый худший путь. Но данный показатель не стабилен, так как порядок дочерних вершин может меняться от эксперимента к эксперименту, и, как следствие, при прочих равных условиях путь от А до В может меняться из раза в раз. Муравьиный же алгоритм сделал петлю. В данном опыте применялись следующие параметры:

* + - 3 - начальное количество феромона;
    - 0,3 – скорость испарения феромона;
    - 0,05 – след, оставляемый муравьем.

Всего тестировалось 10 колоний по 300 муравьев с памятью на 200 ребер.

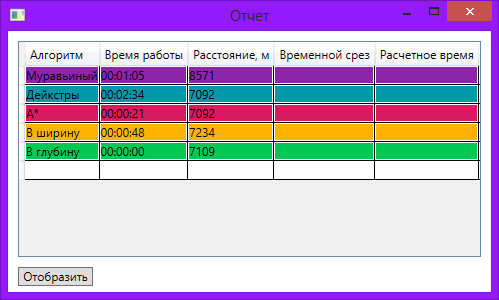


Рис. - Результаты работы алгоритмов по критерию расстояния

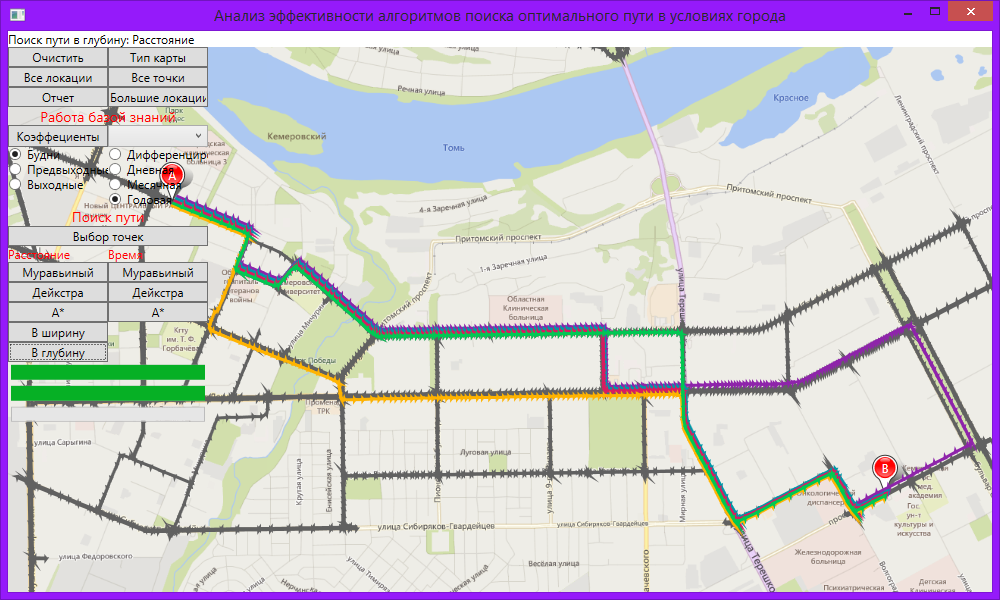


Рис. – Найденные маршруты алгоритмов по критерию расстояния

На Рис. 15 отображён результат работы во временном срезе 6:40 муравьиного алгоритма по критерию расстояния (фиолетовый цвет) и по критерию времени (синий цвет). Визуально, синий путь проходит по локациям с более высокой скоростью движения, результаты отчета это подтверждают (Рис. 16).

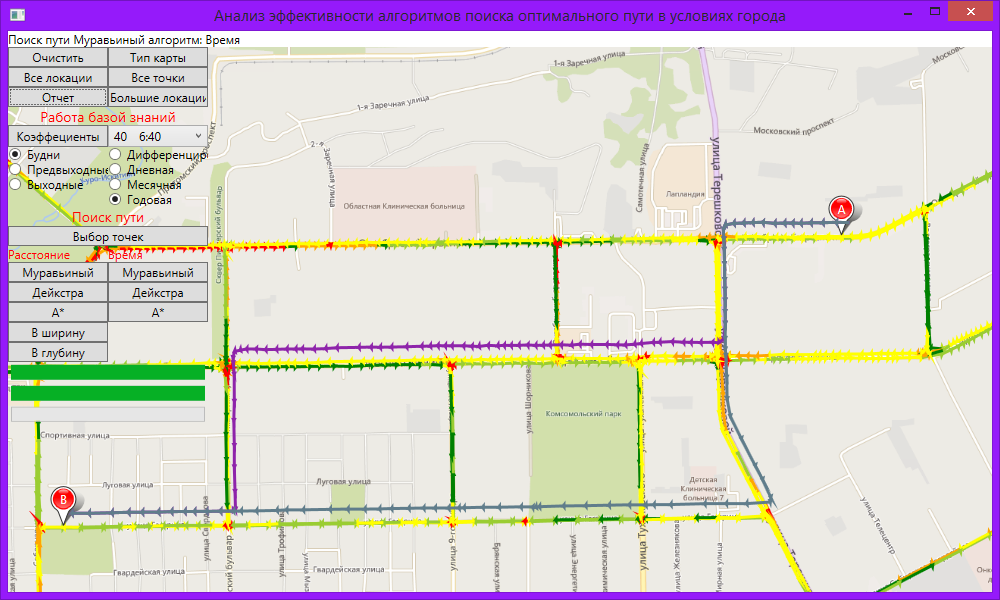


Рис. - Маршруты муравьиных алгоритмов

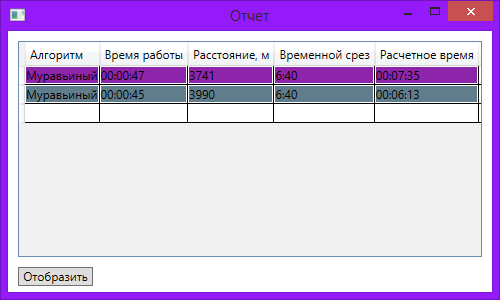


Рис. - Результаты работы муравьиных алгоритмов

Можно заметить, что муравьиный алгоритм по критерию потратил меньшее время на поиск пути, при этом выиграв более минуты в расчетном времени прибытия в конечную точку.

На Рис. 17 результат работы алгоритма Дейкстры, на Рис. 18 – отчёт. При равных временных зтратах на поиск пути, алгоритм по критерию времени выиграл в расчетном времени в текущем срезе. Данный факт подтверждает корректность работы системы.

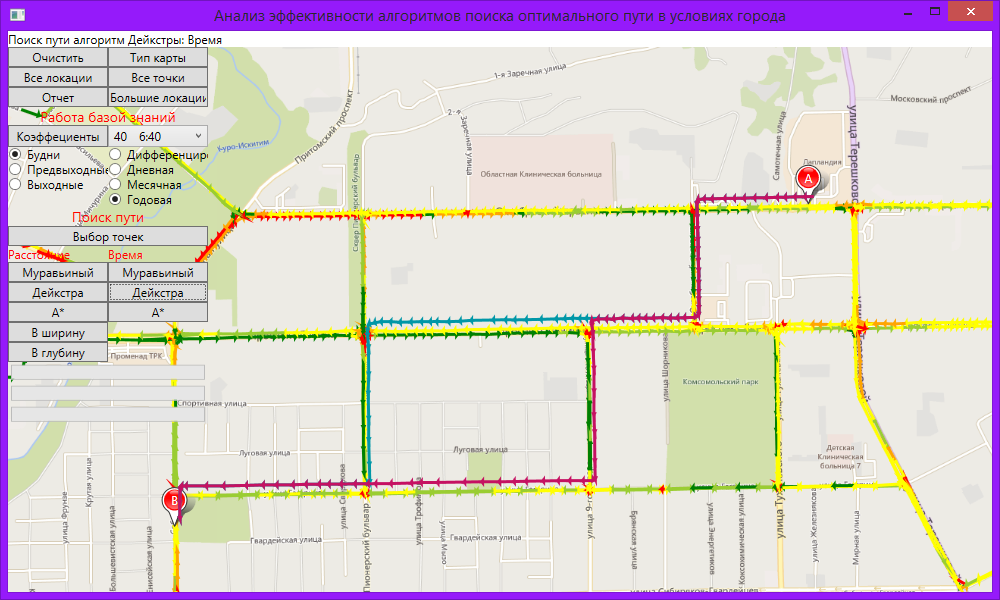


Рис. - Маршруты алгоритмов Дейкстры

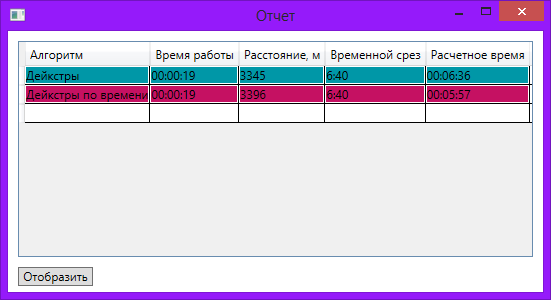


Рис. - Результаты работы алгоритмов Дейкстры

Алгоритм А\* показал похожую схему движения (Рис. 19) с муравьиным алгоритмом, но гораздо с меньшими временными затратами на нахождение пути (Рис. 20). Это объясняется тем, что скорость работы муравьиного алгоритма во многом зависит от начальных параметров. Возможно, уже третья или даже первая колония муравьев нашла конечный путь, но условиями эксперимента жестко задано количество колоний для перепроверки найденного решения, что существенно увеличивает время работы.

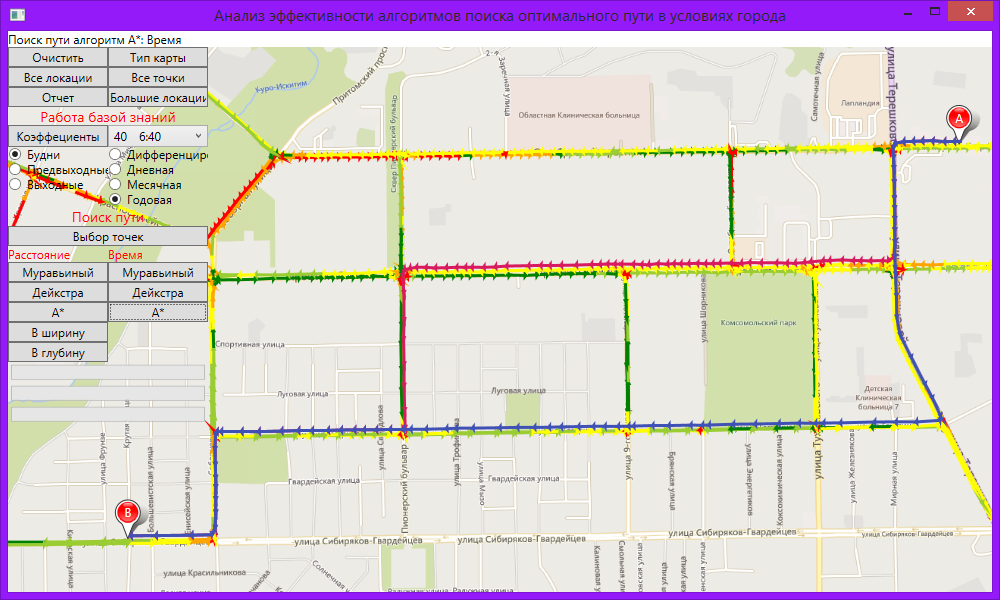


Рис. - Маршруты алгоритмов А\*

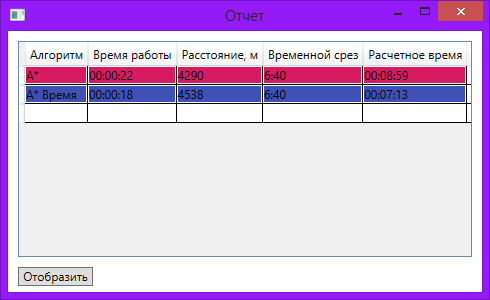


Рис. - Результаты работы алгоритмов А\*

Для более наглядного анализа построим все алгоритмы по критерию времени (Рис. 21). Муравьиный алгоритм и алгоритм Дейкстры показали одинаковое расчётное время, но алгоритм А\* построил другой путь. Это объясняется тем, что алгоритм А\* определяет порядок обхода эвристической функцией «расстояние + стоимость», в случае со временем «время достижения текущей вершины + примерное время достижения конечной вершины». Эвристическая оценка времени достижения конечной точки от рассматриваемой во многом зависит от скоростей на предыдущих локациях, поскольку мы предполагаем, что будем двигаться с такой же средней скоростью. При рассмотрении карты, можно предположить, что локация за перекрестком, выбранная для дальнейшего прохождения алгоритмом А\* имела более высокую среднюю скорость, что улучшило эвристическую оценку времени достижения конечной точки и сделало локацию более привлекательной для дальнейшей работы алгоритма. При этом, в текущих условиях, алгоритм А\* нашел оптимальный путь по соотношению время/расстояние.

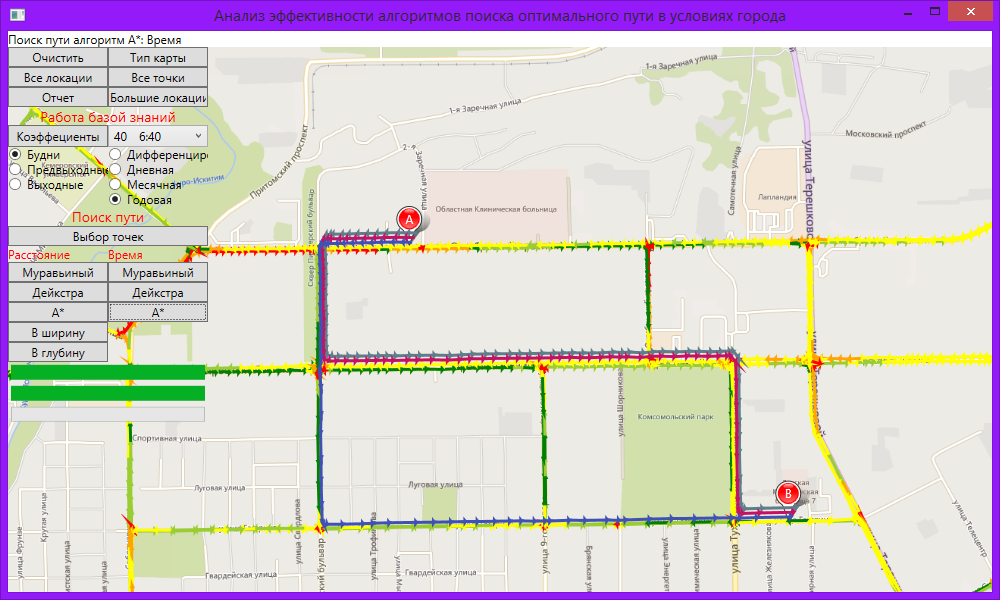


Рис. - Маршруты всех алгоритмов по критерию времени

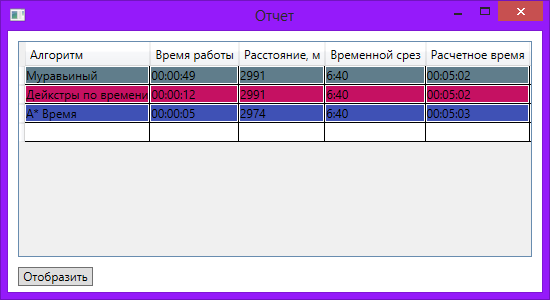


Рис. - Результаты работы всех алгоритмов по критерию времени

Сравним поведение алгоритмов на другом участке дорожной сети. На рисунке 23 отображен результат работы алгоритмов по критерию времени в сравнении с А\* по критерию расстояния. Очевидно, что несмотря на более короткий маршрут, найденный А\* по критерию расстояния, его расчетное время прибытия в конечную точку более чем в три раза превосходит маршруты, найденные алгоритмами по критерию времени. Это объясняется сильным затором в указанное время (7 утра) на одной из улиц, которая составляет участок кратчайшего пути между точками. На рисунке 24 представлен визуализированный результат работы.

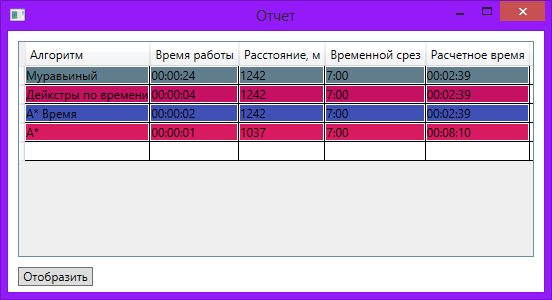


Рис. - Результат работы всех алгоритмов по критерию времени и А\* по критерию расстояния

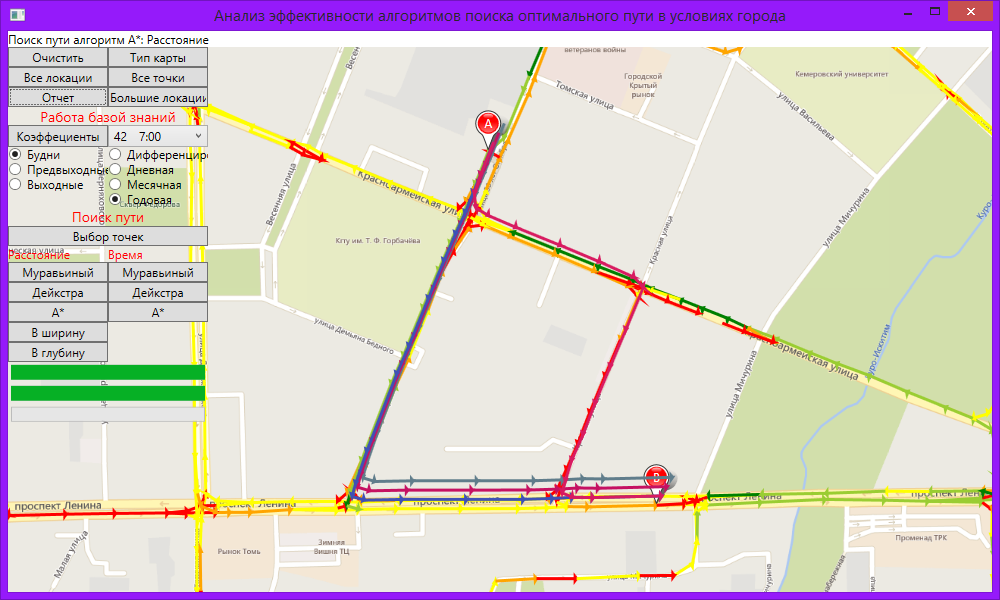


Рис. - Маршруты всех алгоритмов по критерию времени в сравнении с А\* по критерию расстояния

По результатам проведенного исследования по критерию расстояния лучше всего себя показал алгоритм А\*. По критерию времени же алгоритмы показывают примерно одинаковые результаты, но наименьшее время работы так же имеет алгоритм А\*. Можно сделать вывод, что в условиях города найденный кратчайший путь по критерию расстояния не всегда является самым быстрым.

ЗАКЛЮЧЕНИЕ

В рамках выпускной работы бакалавра были решены следующие задачи:

1. изучены проблемы логистики в условиях города
2. проанализированы алгоритмы поиска на графах
3. выбраны средства реализации
4. разработана информационная система, способную осуществлять поиск, оптимизацию и визуализацию маршрутов, а также предоставлять информацию о работе алгоритмов.

Система позволяет находить маршруты в улично-дорожной сети города по различным критериям: расстоянию или времени. Критерий времени учитывает текущее положение на дорогах. Текущее состояние дорожной сети определяется с помощью датчиков ГЛОНАСС или ГЛОНАСС/GPS, установленных на общественном транспорте. Статистика собирается по временным срезам длиной 10 минут с учетом будничных, праздничных и предпраздничных дней. По результатам работы программа предоставляет отчет содержащий в себе: название алгоритма, время работы, длину пути, временной срез, выбранный для анализа и расчетное время прибытия в конечную точку.

Отдельные результаты работы были опубликованы в сборнике VII Всероссийской 60 научно-практической конференции молодых ученых с международным участием «Россия молодая» [26]. Результаты бакалаврского исследования использованы в ООО «ККМ – Торг Сервис» (Приложение А).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Об автомобильных дорогах и о дорожной деятельности в Российской Федерации и о внесении изменений в отдельные законодательные акты Российской Федерации // Федеральный закон Российской Федерации от 08.11.2007 г. № 257-ФЗ.
2. The World Factbook. Central Asia:: Russia [Электронный ресурс] // Central Intelligence Agency. - <https://www.cia.gov/library/publications/resources/the-world-factbook/geos/rs.html>.
3. Райзберг Б.А. Лозовский Л.Ш., Стародубцева Е.Б. Современный экономический словарь [Книга]. - : ИНФРА-М, 2006. - стр. 27.
4. Наличие транспортных средств и происшествия с подвижным составом [Электронный ресурс] // Федеральная служба государственной статистики. - [http://www.gks.ru/wps/wcm/connect/rosstat\_main/rosstat/ru/statistics/enterprise/transport/#](http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/enterprise/transport/).
5. Валентин Никифоров Логистика. Транспорт и склад в цепи поставок [Книга].  -  ГроссМедиа, РОСБУХ, 2008.
6. Об утверждении перечня видов автомобильных транспортных средств, используемых для перевозки пассажиров и опасных грузов, подлежащих оснащению аппаратурой спутниковой навигации ГЛОНАСС или ГЛОНАСС/GPS // Министерство транспорта Российской Федерации, Приказ от 09.03.2010 г. №55.
7. В.С. Дороганов Расчет времени прохождения локаций на основе местоположения общественного транспорта [Конференция].
8. Р. Хаггарти Дискретная математика для программистов [Книга]. - [б.м.] : Техносфера, 2003. - стр. 141.
9. Р. Уилсон Введение в теорию графов [Книга]. - [б.м.] : Мир, 1977.
10. Ф. Харари Теория графов [Книга]. - [б.м.] : УРСС, 2003.
11. Р. Седжвик, К. Уэйн Алгоритмы на Java [Книга]. - [б.м.] : Вильямс, 2013.
12. А. Ахо, Д. Хопкрофт, Д. Ульман Структуры данных и алгоритмы [Книга]. - [б.м.] : Вильямс, 2003.
13. И.В. Красиков, И.Е. Красикова Алгоритмы. Просто как дважды два [Книга]. - [б.м.] : Эскимо, 2007.
14. Стивен С. Скиена Алгоритмы. Руководство по разработке [Книга]. - [б.м.] : БХВ-Петербург, 2011.
15. Т. Кормен Ч. Лейзерсон, Р. Ривест, К. Штайн Алгоритмы. Построение и анализ. [Книга]. - [б.м.] : Вильямс, 2013.
16. Томас Х. Кормен Алгоритмы. Вводный курс [Книга]. - [б.м.] : Вильямс, 2014.
17. С. Рассел, П. Норвинг Искусственный интеллект. Современный подход [Книга]. - [б.м.] : Вильямс, 2014.
18. С.Д. Штовба Муравьиные алгоритмы, Exponenta Pro. Математика в приложениях. 2004. № 4
19. М.Н. Кирсанов Графы в Maple. М.: Физматлит, 2007.
20. Деструкторы (Руководство по программированию в C#) [Электронный ресурс] // Microsoft Developer Network. - <https://msdn.microsoft.com/ru-ru/library/66x5fx1b.aspx>
21. Windows Presentation Foundation [Электронный ресурс] // Microsoft Developer Network. - <https://msdn.microsoft.com/ru-ru/library/ms754130(v=vs.110).aspx>
22. Джим Адамс DirectX Продвинутая анимация [Книга]. - [б.м.] : КУДИЦ-ОБРАЗ, 2004
23. Мэтью Мак-Дональд WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов, 4-е издание [Книга]. - [б.м.] : Вильямс, 2013.
24. Общие сведения о языке XAML (WPF) [Электронный ресурс] // Microsoft Developer Network. - <https://msdn.microsoft.com/ru-ru/library/ms752059.aspx>
25. Microsoft SQL Server [Электронный ресурс] // Microsoft Developer Network. - https://msdn.microsoft.com/ru-ru/library/bb545450.aspx
26. Е.В. Гарченко, И.С. Павлова Алгоритмы решения зедечи о кратчайшем пути [Конференция] Всероссийская научно-практическая конференция Россия Молодая 2015.

ПРИЛОЖЕНИЕ А АКТ ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ



ПРИЛОЖЕНИЕ Б СПИСОК ФАЙЛОВ НА КОМПАКТ-ДИСКЕ

Данное приложение содержит список файлов, находящихся на компакт-диске, прилагаемом к дипломной работе. На диске располагаются:

1. Пояснительная записка – файл «Пояснительная записка.docx»
2. Презентация – файл «Презентация.pptx»
3. Файл базы данных – файл «bdBus\_Data.mdf»
4. Папка «Project» содержит исходные файлы информационной системы.