ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения заочная

«К ЗАЩИТЕ ДОПУЩЕН(А)»
Заведующий кафедрой

_____
(подпись)

Черняева С. Н.
(ФИО)

_____ 20___

Выпускная квалификационная работа

Обучающегося Кудинова Дениса Романовича
(фамилия, имя, отчество)

Вид работы выпускная квалификационная работа бакалавра
(выпускная квалификационная работа бакалавра, специалиста, магистра)

Пояснительная записка

Тема Разработка АИС поддержки транспортных перевозок организации (на примере ООО «Ника»)
(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы доцент, к.т.н., Паринов А. Б..
(должность, подпись, фамилия, инициалы, дата)

Консультант _____
(при наличии) (должность, подпись, фамилия, инициалы, дата)

Консультант _____
(должность, подпись, фамилия, инициалы, дата)

Обучающийся Кудинов Д. Р.
(подпись, фамилия, инициалы, дата)

Воронеж
2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал

Кафедра математики, информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)
Форма обучения заочная

УТВЕРЖДАЮ
Заведующий кафедрой
_____
(подпись)
Черняева С. Н.
(ФИО)
_____ 20__

Задание
на выпускную квалификационную работу

Вид работы ВКР бакалавра
(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Кудинову Денису Романовичу
(фамилия, имя, отчество)

Тема Разработка АИС поддержки транспортных перевозок организации (на примере ООО «Ника»)

Утверждена приказом ректора университета от _____ 20___, № _____

Срок сдачи законченной работы_____ 20___

Исходные данные (или цель ВКР):
разработать автоматизированную информационную систему взаимодействия с клиентами и партнёрами организации

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):
(актуальность темы, цели и задачи ВКР;аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)
– Введение. Актуальность выбранной темы, цель и задачи ВКР

– Исследовательский раздел. Применение автоматизированных информационных систем для поддержки транспортных перевозок.Анализ деятельности организации ООО «Ника». Анализ существующих программных решений

(наименование вопроса, раздела и его краткое содержание)

– Проектный раздел. Выбор средств реализации.Разработка автоматизированной информационной системы поддержки транспортных перевозок.Анализ эффективности работы системы. Руководство пользователя.

(наименование вопроса, раздела и его краткое содержание)

– Заключение.Выводы по работе в целом. Оценка степени решения поставленных задач

(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Оперативное планирование перевозок
4. Структура и анализ предприятия
5. Анализ существующих аналогов
6. Выбор средств реализации
7. Постановка задачи оптимального выбора маршрута
8. Генетический алгоритм
9. Внедрение дополнительной логики
10. Интерфейс пользователя
11. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

2. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

3. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: _____ 20___

Задание согласовано и принято к исполнению: _____ 20___

Руководитель ВКР: доцент, к.т.н., Паринов А. Б. _____

(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: Кудинов Д. Р. _____

(учебная группа, ФИО) (подпись)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность исследования. Логистические задачи, которые в массе своей лежат в основе всех автоматизированных информационных систем транспортных компаний – это задачи, которые, с одной стороны, позволяют оптимизировать процессы доставки товаров, сырьё или материалы до конечных пользователей или в пункты их последующей переработки, тем самым уменьшая время ожидания потребителей или производств. А с другой стороны, задачи данного класса позволяют минимизировать издержки самой транспортной компании, связанные с затратами на топливо, оплату рабочего времени персонала. То есть, можно смело говорить, что оптимальное решение логистических задач принесёт пользу всем сторонам.

Именно поэтому постоянно разрабатываются и актуализируются алгоритмы решения задач данной категории, а сами транспортные компании стараются пользоваться результатами улучшения ранее разработанных алгоритмов, внедряя современные программные продукты в свою повседневную деятельность.

Целью данной выпускной квалификационной работы является разработка автоматизированной информационной системы поддержки транспортных перевозок организации. Эффективное построение маршрутов с учётом особенностей доставки – это нетривиальная задача, которая стоит перед многими транспортными компаниями. Решение её средствами автоматизированной системы позволит организации существенно улучшить своё положение на рынке.

Предметом исследования является разработкасоответствующей автоматизированной информационной системы.

Объектом исследования в рамках данной выпускной квалификационной работы являются процессы управления перевозками транспортной компании.

Целями выпускной квалификационной работы являются:

— изучение особенностей автоматизированных информационных систем, которые применяются в деятельности транспортных компаний;

— определение основных критериев и параметров, влияющих на процесс планирования и составления маршрута;

— оценка необходимости создания соответствующей автоматизированной информационной системы;

— анализ и оценка существующих программных решений, их особенностей и нюансов работы;

— анализ языков программирования, которые могут быть использованы для создания автоматизированной информационной системы, определение их особенностей и оценка недостатков;

— изучение основных алгоритмов выбора оптимального пути;

— изучение генетического алгоритма;

— расширение генетического алгоритма, добавление в него требуемой логики;

— реализация разработанной автоматизированной информационной системы на основе выбранных технологий и инструментов программирования;

— оценка результатов работы системы.

Практическое значение данной выпускной квалификационной работы заключается в том, что её результаты были апробированы и планируются к внедрению в компании ООО «Ника», которая специализируется на перевозках сельскохозяйственных культур

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1Применение автоматизированных информационных систем для поддержки транспортных перевозок

1.1.1. Определение автоматизированной информационной системы и описание её подсистем

В соответствии с классическим определением под автоматизированной информационной системой (АИС) понимается система, которая состоит из персонала и комплекса средств автоматизации его деятельности, выполняющая заранее определённые функции и решающая описанные ранее задачи.

Компоненты АИС можно быть представлена как объединение обеспечивающих подсистем.

В общем виде компоненты (или подсистемы АИС) представлены на рис. 1.1.

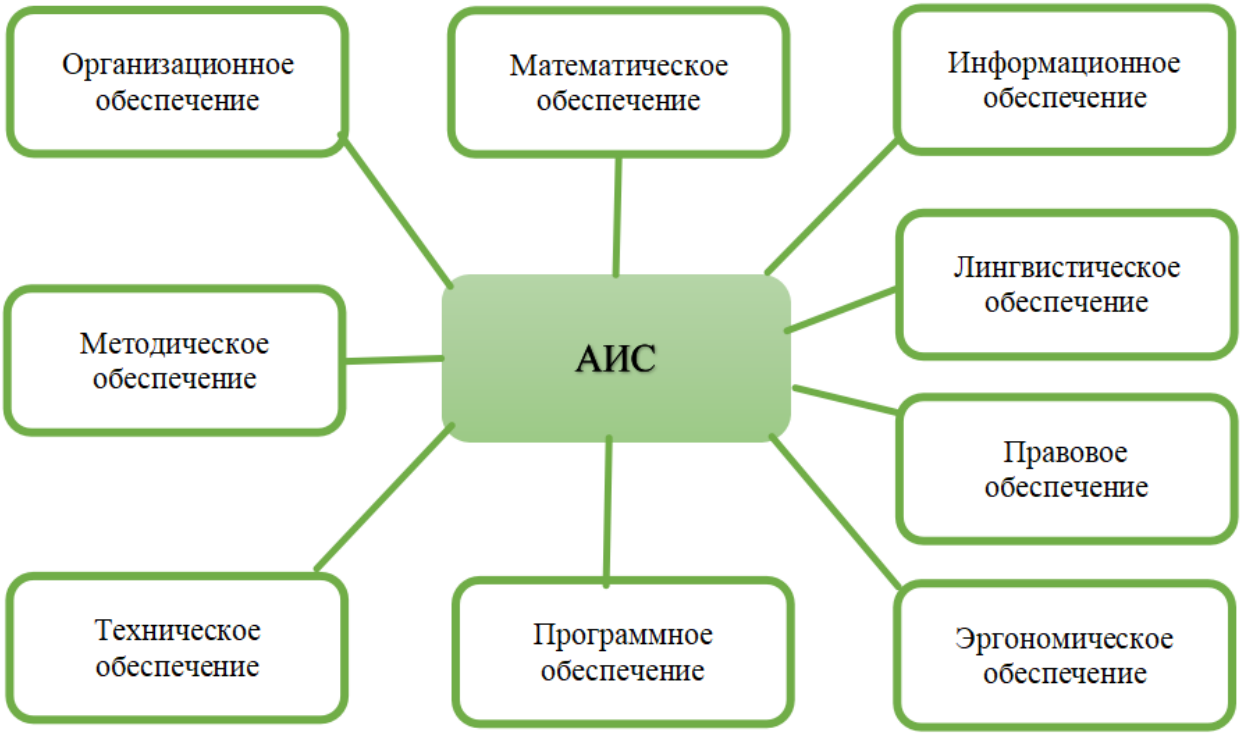


Рисунок 1.1 – Обеспечивающие подсистемы АИС

Под организационным обеспечением автоматизированной информационной системы будем понимать совокупность документов, которые определяют с устанавливают организационную структуру, права и

обязанностипользователей и персонала, отвечающего за эксплуатацию АИС для обеспечения её работоспособности.

Методическое обеспечение АИС–это те документы, которые описывают технологии функционирования АИС, методы и способы полученияконкретных результатов, которые могут применять пользователи АИС. В состав методического обеспечения входят все методические указания, пособия, учебные курсы, обеспечивающие подготовку будущих пользователей автоматизированных информационных систем.

Техническое обеспечение – это весь комплекс технических средств, которые используются для работы АИС, а также вся необходимая документация, описывающая эти средства и технологические процессы.

Под техническими средствами понимаются:

— компьютеры любых моделей;

— все устройства, используемые для сбора, хранения, обработки, передачи и вывода полученной информации;

— каналы передачи данных и иные линии связи;

— любая организационная техника и устройства автоматического получения информации;

— иные комплектующие.

Под математическим обеспечениемавтоматизированных информационных систем далее будем понимать всё множество математическихметодов, моделей и алгоритмов, применяемых в работе АИС. К этим моделям и методам можно отнести средства моделирования процессов управления, методы математического программирования, математическойстатистики,алгоритмы теории графов и комбинаторики и др.

Программное обеспечение АИС — это объединение программ и программных документов, которые используются для отладки, корректного функционирования автоматизированной информационной системы.

В состав программного обеспечения АИС входят:

— непосредственно программные продукты;

— лицензии на использование программных продуктов;

— вся техническая документация.

Под информационным обеспечением автоматизированной информационной системы понимается совокупность форм документов, актуально к данному моменту нормативной базы, применяемой в АИС при её функционировании. Подсистема информационного обеспечения позволяет гарантировать, что информация, которая будет использована АИС является достоверной и может быть использована для принятия управленческих решений.

Лингвистическое обеспечение АИС представляет собой всё множествометодов и правил, которые используются для формализации естественного языка. В состав лингвистического обеспечения входят, как правило, словари терминов, касающиеся конкретной АИС и словари сокращений.

Под правовым обеспечением автоматизированных информационных систем понимаются все правовые нормы, которы регламентируют возникающие при функционировании АИС правовые отношения, а также последующий юридический статус результатов её функционирования.

В состав правового обеспечения входят:

— законы, постановления государственных органов власти, инструкции и иные нормативные документы, начиная от министерств и ведомств, заканчивая отдельными организациями;

— всё правовое обеспечение этапа разработки, включающее в себя нормативные акты, регламентирующие договорные отношения между разработчиком и заказчиком, а также штрафные санкции, предусмотренные возникающими отклонениями от договора;

— правовое обеспечение этапа функционирования автоматизированной информационной системы, включая непосредственно статус системы, права и обязанности персонала, порядок создания и

последующего использования полученной в ходе функционирования АИС информации и др.

Эргономическое обеспечение АИС представляет собой объединение реализованных в АИС требований к психологическим, физиологическим характеристикам и возможностям потенциальных пользователей определённой АИС, а также описание требований к организации рабочих мест для пользователей АИС.

1.1.2 Использование автоматизированных информационных систем в сфере транспортных перевозок

Подсистема управления перевозками представляет собой часть автоматизированной информационной системы, основными задачами которой являются планирование, организация, контроль, регулирование, а также учёта перевозочного процесс. Длярешения задачи управления перевозками используются экономико-математические методы и алгоритмы работы на графах.

Основная цель разработки АИС управления транспортными перевозками– повышение эффективности работы подвижногосостава за счётоптимизации планирования перевозок и эффективного управления всем процессом.

Эффективность внедрения подобной автоматизированной информационной системы связана с оптимизацией использования автотранспортных средств предприятия иснижения затрат на перевозки за счёт:

— минимизации потерь рабочего времени;

— уменьшения доли порожних пробегов;

— уменьшение времени простоя автомобилей;

— повышение коэффициента использования грузоподъёмности транспортныхсредств;

— сокращения расстояний перевозок за счёт оптимизации маршрутов.

Автоматизированная информационная система поддержки перевозок позволяет для каждого клиента и маршрута точно рассчитать необходимое число единиц техники для выполнения заданного объёмазадач более точно, чем это может сделать оператор.

Схема оперативного планирования перевозок представлена на рис.1.2.

Рисунок 1.2 – Схема оперативного планирования перевозок

Для эффективного планирования перевозок требуется:

— банк исходных данных, который содержит информацию о кратчайших расстояниях между пунктами транспортной сети,

— данные о транспортных средствах предприятия и их характеристиках предприятия,

— информацию о составе водителей.

Тогда диспетчер автопредприятия на основе выданного программой решения может спроектировать маршруты сточки зрения наибольшей эффективности перевозочного процесса.

Послесредствами АИС может быть формирован план сменно-суточных задания для водителей с формированием требуемой путевой документации.

За счёт возможностей АИС планирования перевозок, обработка путевых листов также может быть автоматизирована.

Все автоматизированные информационные системы поддержки транспортных перевозок строятся на следующем принципе организации работы с транспортом организации: сначала стоится план, ориентированный на централизованное планирование, а при выявлении ситуаций сбоев – на перераспределение транспортных средств вне зависимости от ранее выполненного закрепления автомобилей за определёнными на данный день маршрутами.

Рассматривая систему транспортировки товаров и грузов, к входящим информационным потокам можно отнести:

— информацию о наличии грузов на складах грузовладельцев, а также её количество, сроки хранения, транспортные свойства и т.д.;

— информацию о наличии свободных автотранспортных средств, с а именно их количество, тип, характеристики, а также условия работы или оплаты;

— детали заказов, которые включают в себя адреса пунктов завоза, количество завозимого груза, время завоза и иные детали;

— данные о состоянии транспортной сети региона.

К исходящим информационным потокам можно отнести, во-первых, составленный на данные сутки план перевозок, к которому относятся система маршрутов и маршрутная сеть компании, передаваемая конечным исполнителям.

Также, к исходящим информационным потокам относятся и рассчитанные графики движения автомобилей, их государственные номера и иные данные, которые передаются получателям грузов.

Особый вид возникающих информационных потоков – это обратная связь о ходе перевозочного процесса и обо всех возникающих сбоях. Наличие и доступ к этой информации позволяет диспетчеру, используя ресурсы АИС, производить изменения в маршрутах, задействовать при

необходимости резервный транспорт и проинформировать клиентов об изменении деталей доставки, а также находиться в постоянной связи с водителями и давать им требуемые указания.

Схема информационных потоков по доставке грузов представлена на рис. 1.3.

Рисунок 1.3 – Схема информационных потоков по доставке грузов

В связи с большим разнообразием в задачах планирования и управления работы транспортных средств идея представления данных в виде единой информационной системы транспортного обслуживания клиентов становится все более и более актуальной. Наличие единой системы, которая удовлетворяла бы требованиям разных категорий пользователей, к которым относят грузоотправителей, заказчиков, перевозчиков, позволила бы существенно облегчить работу по проектированию маршрутов и схем доставки, а также позволила бы использовать сходные подходы и единые

алгоритмы планирования транспортного процесса. Это, в свою очередь, может существенно упростить процедуру взаимодействия всех его участников.

1.2 Анализ деятельности организации ООО «Ника»

Общество с ограниченной ответственностью «Ника» является одним из предприятий-долгожителей в Воронежской области. Свою деятельность ООО «Ника» начало более двадцати трёх лет назад.

Основной сферой деятельности ООО «Ника» является выращивание зерновых, зернобобовых культур и семян масличных культур. Предприятие зарекомендовало себя как одного из наиболее стабильных участников рынка с очень хорошей репутацией.

Помимо основных сфер деятельности, ООО «Ника» оказывает ряд дополнительных услуг, которые отражены на рис. 1.4.

Рисунок 1.4 – Дополнительная деятельность ООО «Ника»

Представленная схема позволяет увидеть, что оказание транспортных услуг – это один из основных дополнительных источников, помимо основной деятельности организации.

Причём, следует заметить, что ООО «Ника» активно использует имеющийся в его распоряжении штат автотранспорта для доставки

выращенных культур, т.е. для эффективного осуществления и основой деятельности организации.

Как было сказано выше, деятельность ООО «Ника» началась более 20 лет назад, когда автоматизация и использование компьютерных решений ещё были не очень широко распространены. Это нашло отпечаток в деятельности ООО «Ника»: многие действия и процессы на данный момент там ещё не автоматизированы, хотя руководство постоянно задумывается о внедрении программных решений для повышения эффективности деятельности компании на рынке.

Одной из сфер деятельности ООО «Ника», которую было решено автоматизировать, была выбрана сфера грузоперевозок. На данный момент составление маршрута будущих перевозок – это задача сотрудника, который отвечает за логистику. Именно в его обязанности входит изучение заявок и составление маршрутного листа для водителей.

Для иллюстрации деятельности ООО «Ника» в сфере грузоперевозок и обозначения места разрабатываемой АИС, используем диаграмму IDEF0. Данные диаграммы используются для моделирования и анализа систем, протекающих в них процессов и операций. IDEF0 диаграммы помогают визуализироватьфункциональные элементы и их взаимодействия в системе.

Диаграмма процесса «Перевозка груза» приведена на рис. 1.5.

Рисунок 1.5 – Диаграмма «Перевозка груза»

Составленная диаграмма позволила выделить входящий информационный поток: заявку, исходящий поток: доставку груза, а также управляющие потоки и исполнительные механизмы.

В число управляющих потоков входят:

— нормативные документы и законы;

— параметры перевозки грузов.

Исполнительные механизмы, выделенные с помощью данной модели:

— транспортные средства;

— диспетчер;

— механик;

— медработник;

— водитель.

Декомпозиция диаграммы позволяет определить подфункции, которые имеются у разрабатываемой системы (рис. 1.6).

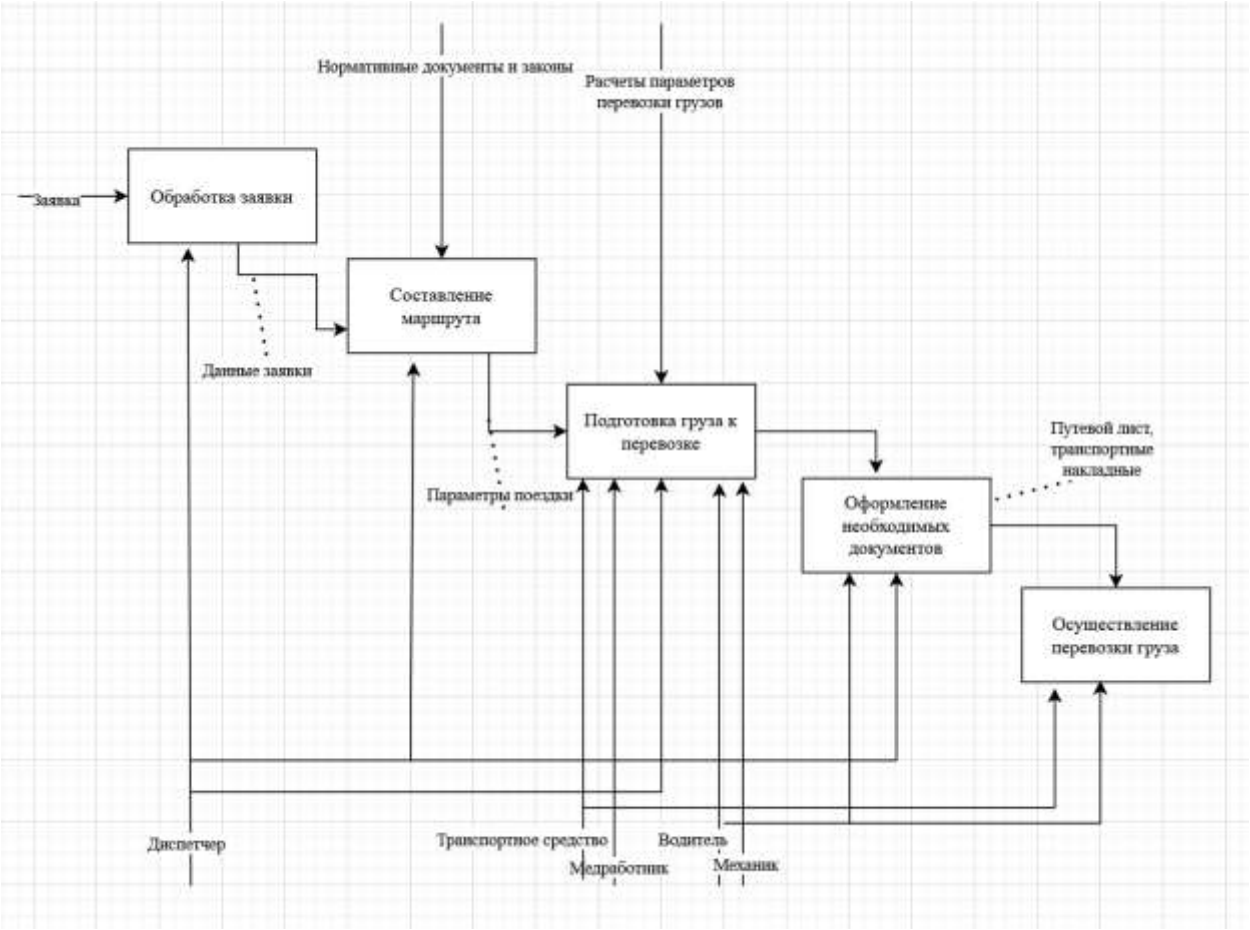


Рисунок 1.6 – Декомпозиция функции «Перевозка груза»

Декомпозиция основной функции позволяет определить подфункции:

— обработка заявки;

— составление маршрута;

— подготовка груза к перевозке;

— оформление необходимых документов;

— осуществление перевозки груза.

Ввиду поставленной в рамках данной выпускной квалификационной работы задачей, наибольший интерес будет представлять процесс «Составление маршрута». Выполним его декомпозицию (рис. 1.7).

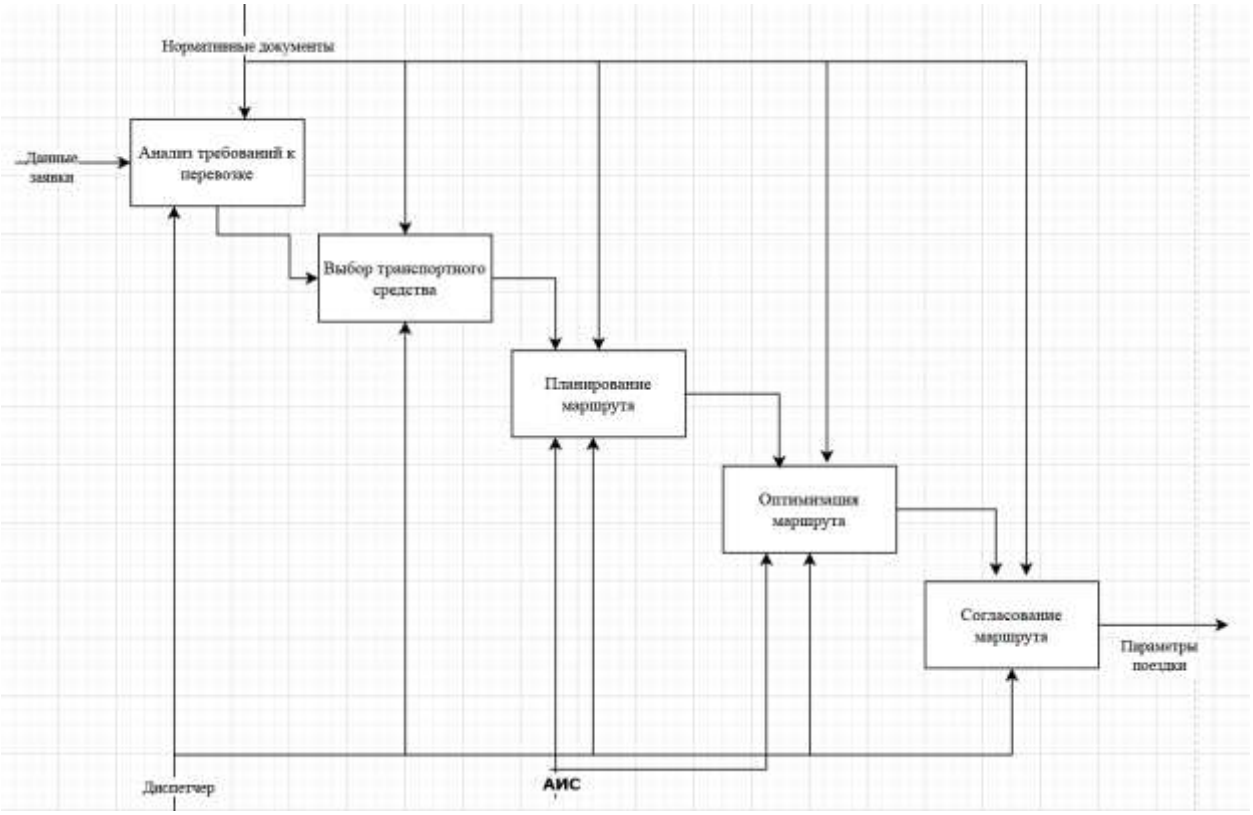


Рисунок 1.7 – Декомпозиция функции «Составление маршрута»

В результате декомпозиции получена диаграмма, в которой уже отражено место разрабатываемой АИС поддержки транспортных перевозок ООО «Ника».

1.3 Анализ существующих программных решений

В рамках написания данной выпускной квалификационной работы были изучены возможные программные аналоги, которые решают поставленные задачи. Следует отметить, что российский рынок имеет много готовых программных решений, каждое из которых обладает своими особенностями и нюансами.

Основными программными продуктами, которые используются в Российской Федерации для решения логистических задач, являются:

— «Яндекс.Маршрутизация» – это сервис автоматического построения маршрутов для городской доставки. Сервис позволяет учитывать достаточно большое число параметров, например, брать в расчёт габарит грузо, вместимость транспортного средства, график работы складов и др.

— RELOG – это популярный облачный сервис для координирования работы внутригородской логистики. Построение маршрутов происходит с помощью алгоритмов, использующих машинное обучение. При построении маршрута можно уточнить около 40 параметров доставки, таких как временные окна, особенности и характеристики товара, показатели грузоподъемности машин и т.д.

— Maxoptra – это ещё одна известная система транспортного планирования, которая позволяет распределять заказы между водителями в автоматическом режиме. Данный сервис использует в своей работе систему ГЛОНАСС и GPS-мониторинг.

— «1С:Предприятие 8. TMS Логистика. Управление перевозками» – это программа для планирования грузоперевозок компании с использованием собственного и привлечённого транспорта.

Есть и другие программные продукты, однако, в процессе проведения анализа было принято решение остановиться на четырёх наиболее используемых системах.

Проведённый анализ готовых систем показал, что, на данный момент ООО «Ника» не готово нести затраты по внедрению готовых систем, а также получать временные и прочие издержки, связанные с затратами на интеграцию готовых систем и обучением пользователей работе в этих системах.

Также анализ показал, что большинство готовых решений оптимизировано под внутригородскую доставку грузов, что не совсем совпадает с областью деятельности ООО «Ника».

Поэтому наилучшим решением на данный момент будет разработка небольшой автоматизированной системы поддержки транспортных перевозок организации.

Результаты сравнения приведены в табл. 1.1.

Таблица 1.1– Сравнение существующих программных решений

Критерий	«Яндекс. Маршрутизация»	RELOG	Maxoptra	«1С»
Кому подходит?	дистрибьюторам, ретейлерам, курьерским службам, финансовым организациям. сервисным компаниям	ретейлерам, дистрибьюторам фармацевтики, компаниям e-commerce, мебельным компаниям, производителям хлебобулочных изделий, компаниям по доставке воды	дистрибьюторам, компаниям; курьерским службам, сервисам, работающим в доставке готовых блюд, транспортным компаниям.	компаниям, занимающимся большими перевозками, компаниям с доставкой по регионам и внутри городов
Доступные функции	быстрое построение маршрутов для большого количества точек, учёт прогноза пробок и данных «Яндекс. Карты», расчёт времени приезда водителя на каждую точку, отслеживание в реальном времени движения водителей по построенным маршрутам	автоматическое создание эффективных маршрутов с помощью искусственного интеллекта, создание маршрутов в соответствии с конкретными транспортными задачами компаний, мониторинг движения водителей и их пребывания в разных точках	автоматическое планирование маршрутов и возможность вручную их корректировать, планирование затрат на топливо, контроль работы водителей, сокращение суммарного пробега автопарка и рабочего времени сотрудников	автоматическое планирование региональной/местной доставки для большого количества заявок, выбор исполнителя перевозки по каждому звену перевозки, контроль за выполнением рейсов
Минусы	— внедрение может быть дорогим и требовать значительных ресурсов; — точность и эффективность зависят от качества входных данных и обучения системы; — требуют интеграции с существующими системами компании, что может быть сложным процессом			
Стоимость внедрения	зависит от количества машин, выбранной периодичности платежа и набора сервисов			

В рамках данной выпускной квалификационной работы требуется разработать автоматизированную информационную систему, которая позволяла бы:

— планировать маршруты движения транспорта;

— вносить корректировки в маршруты;

— вносить в проектируемые маршруты дополнительные ограничения;

— оценить качество полученного прогноза;

— предоставить удобное для диспетчера хранение полученных результатов.

2. ПРОЕКТНЫЙ РАЗДЕЛ

2.1 Выбор средств реализации

Для решения задачи, поставленной в рамках данной выпускной квалификационной работы необходимо подобрать наиболее подходящий для этого язык разработки.

Поскольку, большинство из известных языков программирования могут быть использованы для написания автоматизированной информационной системы поддержки работы транспортной компании, при выборе языка программирования сравним три наиболее популярных языка по итогам 2023 года.

Данные об использовании тех или иных языков приведены на рис. 2.1.

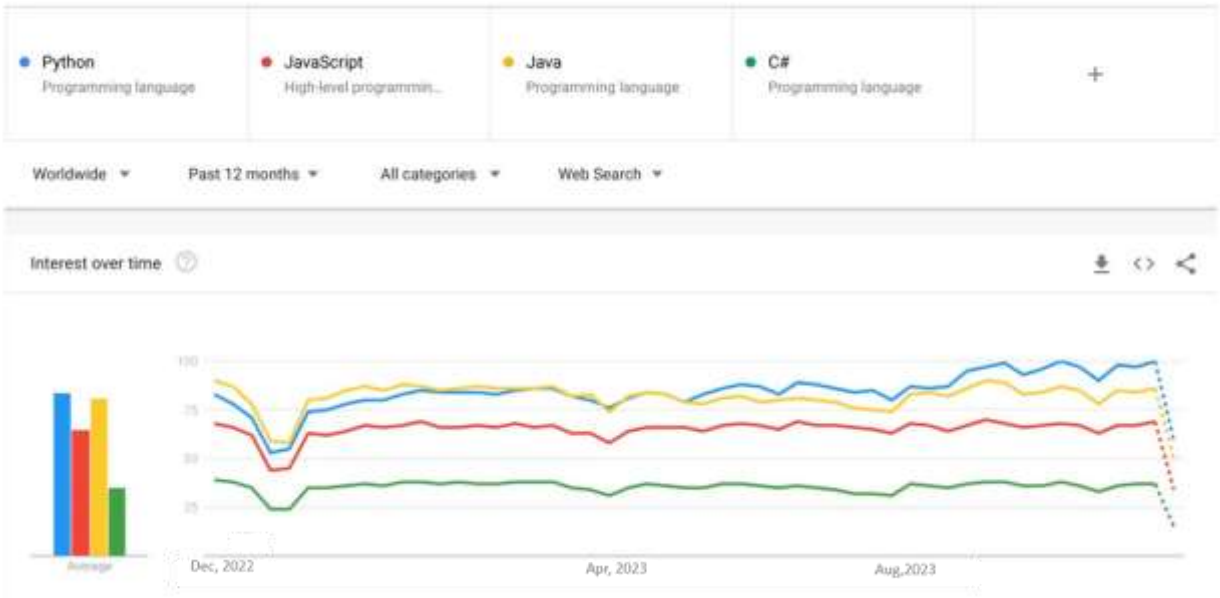


Рисунок 2.1 – Статистика, показывающая популярность языков программирования по итогам 2023 года

Лидирующую позицию занимает язык программирования Python. Это язык программирования, который получил широкую популярность из-за своего простого синтаксиса. Он не имеет какой-то ограниченной сферы применения и подходит как для создания сайтов или анализа данных,так и для работы над проектами связанными с искусственным интеллектом.

Второй в списке наиболее популярных языков разработки – это язык программирования Java. Этот язык не теряет своей популярности многие

годы, поскольку он работает на всех доступных платформах, постоянно обновляется и модернизируется, а также не имеет какой-либо ограниченной сферы применения и может быть использован в разноплановых проектах.

На третьей строчке по популярности находится язык программирования JavaScript.Этот язык программирования широко используется для веб-разработки, создания интерактивных элементов на веб-страницах. Особенности языка делают его наилучшим выбором для решения задач данной категории.

Более подробное сравнение трёх особенностей наиболее распространённых языков программирования приведено в табл. 2.1.

Таблица 2.1 – Сравнение основных особенностей трёх наиболее популярных языков программирования

Критерий	Python	Java	JavaScript
Синтаксис	понятный синтаксис, лёгкость изучения	не является самым простым среди всех языков программирования, однако он считается достаточно стройным и понятным для большинства разработчиков	считается относительно лёгким и простым для изучения
Соответствие парадигмам программирования	соответствует объектно-ориентированным парадигме	соответствует объектно-ориентированным парадигме	соответствует мультипарадигменному подходу
Frontend /backend	может использоваться для создания frontend и backend приложений	преимущественно используется в backend разработке	является основным языком для frontend разработки
Безопасность и управление памятью	автоматическое управление памятью, что упрощает разработку, но может привести к неэффективности при работе с большими объёмами данных	безопасность и управление памятью осуществляются через механизмы сборки мусора	предоставляет автоматическое управление памятью

Однако, помимо аргументов «За» в выборе языка разработки, следует уделить внимание и контраргументам, иными словами рассмотреть и недостатки выбранных для анализа языков программирования.

Сравнение недостатков популярных языков разработки приведено в табл. 2.2.

Таблица 2.2 – Сравнение недостатков трёх наиболее популярных языков программирования

Язык программирования	Недостатки		
Python	редко используется для мобильной разработки из-за доступности различных фреймворков	не так широко используется в разработке корпоративного программного обеспечения в крупных корпорациях	сложность перехода на другой язык программирования из-за простоты и допущений синтаксиса
Java	это старый язык программирования, многие разработчики отдают предпочтение изучению более новых языков программирования	приложения на Java обычно требуют большого объёма оперативной памяти из-за особенностей работы виртуальной машины	невысокая производительность в некоторых задачах, связанных с работой в реальном времени
JavaScript	использование функциональной области видимости, что может привести к непредсказуемым результатам при работе с переменными и функциями	слишком быстрые изменения языка и его фреймворков	код выполняется на стороне клиента, что может приводить к проблемам с безопасности

Сравнение трёх наиболее популярных в 2023 году языков разработки позволило сделать выбор в пользу языка программирования Python. Этот язык выбран из-за понятного и простого синтаксиса, большой и дружелюбной экосистемы разработчиков, а также за наличие огромного числа открытых библиотек, которые могут ускорить процесс разработки.

2.2 Разработка автоматизированной информационной системы поддержки транспортных перевозок

2.2.1 Задача выбора оптимального маршрута и алгоритмы её решения

Задача об оптимальном пути состоит в поиске наилучшего по определенным критериям пути между двумя точками на графе, в которой минимизируется сумма весов рёбер, составляющих путь. Это одна из важнейших классических задач теории графов.

К сегодняшнему дню существует много алгоритмов для её решения. Не смотря на это, задача сохраняет высокую значимость из-за различных практических сфер её применения. Классический пример возникновения этой задачи в реальной жизни – это, например, поиск кратчайшего пути в GPS-навигаторах или построение маршрута с учётом пробок.

В рамках данной задачи все пункты назначения становятся вершинами графа и образуют множество вершин V, а все дороги выступают в качестве дуг между этими вершинами и образуют множетсво E.

Тогда можно задать граф G(V,E), где каждой дуге (u,v) ставится в соответствие число L(u,v) – длина дуги от вершины u до вершины v.

В различных постановках задачи в качестве длины дуги могут выступать не только расстояние от вершины u до вершины v, но и, к примеру, время движения, стоимость перемещения от вершины u до вершины v, или другие характеристики, связанные с прохождением каждой дуги.

Существуют различные постановки данной задачи:

— задача об оптимальном пути из начальной точки в заданный пункт назначения. В данном случае требуется найти наиболее оптимальный путь в заданную вершину назначения t, из начальной вершины графа, в роли которой может выступать любая из вершин графа, кроме t.

— Задача об оптимальном пути между заданной парой вершин. В данном случает требуется найти наиболее оптимальны путь из заданной вершины u в заданную вершину v.

— Задача об оптимальном пути между всеми парами вершин графа. Требуется найти самый оптимальный путь из каждой вершины u в каждую вершину v.

Задача решается алгоритмами из разных типов. Сюда можно отнести как классические алгоритмы на графах, так и алгоритмы комбинаторной оптимизации или эволюционные алгоритмы.

Наиболее известным алгоритмом на графах поиску оптимального пути является алгоритм Дейкстры. Данный алгоритм позволяет найти самый короткий путь между между двумя вершинами, постепенно исследуя все возможные маршруты и обновляя минимальные расстояния до каждой из вершин.

Решение задачи начинается со стартовой вершины, а далее анализируются все возможные пути к соседним вершинам. Этот процесс продолжается, пока не будет найден самый короткий маршрут до конечной вершины.

Логика работы алгоритма может быть проиллюстрирована с помощью схемы, представленной на рис. 2.2

Рисунок 2.2 – Алгоритм Дейкстры

В классе задач комбинаторной оптимизации также есть алгоритм поиска оптимального пути. Он известен, как задача коммивояжёра.

Классическая постановка задачи коммивояжёра имеет следующий вид: "У коммивояжёра есть список городов, которые он должен посетить ровно один раз, и вернуться в исходный город. Необходимо найти такой маршрут, чтобы общая пройденная дистанция была минимальной."

Основные алгоритмы комбинаторной оптимизации, которые могут быть использованы для решения поставленной задачи, приведены на рис. 2.3.

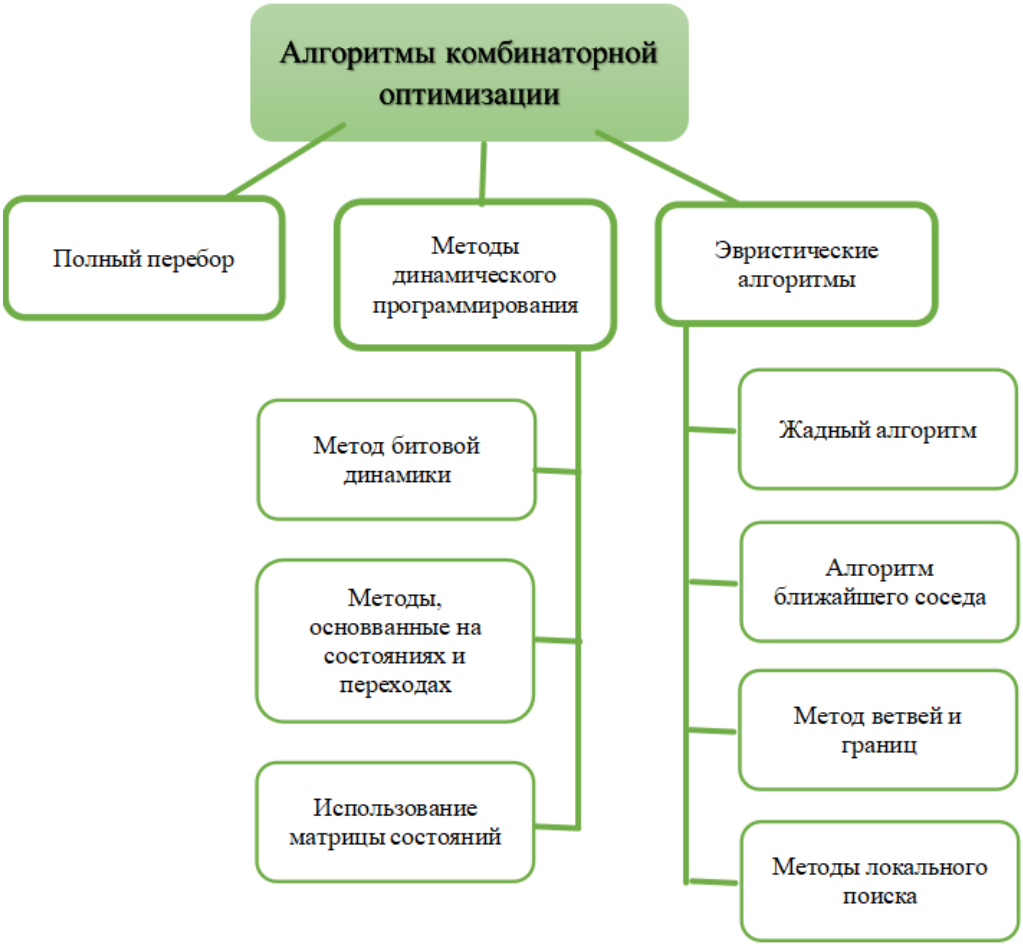


Рисунок 2.3 – Алгоритмы комбинаторной оптимизации

Методами решения задачи коммивояжёра являются:

— алгоритм полного перебора, в ходе которого проверяются все возможные маршруты и из них выбирается потом наилучший. Полный перебор позволяет получить точное решение задачи, но не является неэффективным способом её решения, поскольку не подходит для большого количества городов.

— Методы динамического программирования показывают лучшее время по сравнению с полным перебором, но также могут быть неэффективны для большого количества городов.

— Эвристические методы, в основе которых лежат приближенные алгоритмы для нахождения хороших, но не обязательно оптимальных решений. Наиболее известными эвристическими алгоритмами являются жадный алгоритм, алгоритм ближайшего соседа, метод ветвей и границ, а также методы локального поиска. Недостаток данной группы алгоритмов

состоит в том, что они не всегда позволяют получить точное решение задачи, получая только одно из хороших допустимых решений.

Среди эволюционных алгоритмов задача поиска оптимального пути может решаться с помощью генетического алгоритма. Под генетическим алгоритмом понимается эволюционный алгоритм поиска, который использует идею последовательного подбора, комбинирования и вариации искомых параметров с использованием механизмов, которые напоминающих биологическуюэволюцию.

Процесс поиска оптимального пути начинается с множества случайных решений, которые впоследствии оцениваются, из них выбираются лучшие решения. После эти выбранные решения комбинируются и изменяются. Процесс повторяется до тех пор, пока не будет найдено удовлетворительное решение.

Иными словами, генетический алгоритм также позволяет найти только одно из допустимых решений задачи (не обязательно оптимальное), но он обладает определенным преимуществом среди всех других алгоритмов, которые упоминались выше.

Это преимущество состоит в том, что маршрут, который будет получаться в ходе работы генетического алгоритма, можно немного «настроить», т.е. добавить некоторые уточнения, которые будут учитываться в последующем при получении решения. Например, можно указать, какой из пунктов обязательно должен быть посещён вторым по счёту или предпоследним из всех. Поэтому в качестве основного алгоритма решения задачи был выбран именно генетический алгоритм.

2.2.2 Генетический алгоритм в решении логистических задач

Генетический алгоритм – это метод поиска решений для сложных задач, который опирается на один из естественных процессов: процесс эволюции.

В общем виде последовательность действий алгоритма может быть отражена следующим образом (рис.2.4).

Рисунок 2.4 – Последовательность действий генетического алгоритма

Рассмотрим шаги алгоритма более подробно.

Начальный шаг, который обязательно нужно пройти, – это инициализация или, иными словами, создание начальной популяции. На данном этапе формируется N случайных возможных маршрутов от начального до конечного пункта. Эти возможные решения называются особями, а их объединение – популяцией.

Далее следует этап оценки особей популяции. Для каждой особи вычисляется пригодность (обычно эта функция называется fit). Т.е. для каждого элемента популяции находится расстояние от начальной до конечной точки маршрута.

Далее алгоритм приступает к этапу селекции. Из всей полученной популяции отбираются лучшие особи. В поставленной задаче это будут особи с наименьшей длиной маршрута.

После этого пары лучших особей комбинируются, чтобы создать новые наборы инструкций. В алгоритме этот шаг называется этапом скрещивания. Полагается, что на этом этапе соединяются две хорошие особи, чтобы на основе их комбинации получить ещё более лучшую особь.

Далее идёт этап мутации или внесения случайных изменений. К случайно выбранным особям из популяции добавляются случайные изменения. Этот процесс называется мутация. Полагается, что этот этап позволит получить неожиданные новые характеристики особи и не позволит алгоритму уйти к выбору особи одного типа.

После этапы оценки, селекции, скрещивания и мутации повторяются много раз, каждый раз формируя новую популяцию особей. Согласно алгоритму, каждая следующая популяция будет показывать решение, все больше и больше приближенное к точному.

Генетический алгоритм имеет ряд очевидных преимуществ. К ним можно отнести:

— гибкость алгоритма и возможность его применения для широкого круга задач;

— скорость поиска подходящего решения за счёт логики работы алгоритма;

— способность избегать локальных минимумов благодаря процедурам мутации и скрещивания.

Однако, как и все алгоритмы поиска допустимого решения, генетический алгоритм имеет и ряд недостатков:

— алгоритм может задействовать значительные вычислительные ресурсы;

— генетический алгоритм не гарантирует оптимальное решение;

— эффективность алгоритма сильно зависит от параметров, например, он вероятности мутации или количества особей, которые будут скрещиваться.

Генетический алгоритм, как говорилось выше, имеет широкую сферу применения. Но, из-за своих особенностей, он широко применяется в решении логистических задач. Основные направления применения генетического алгоритма приведены в табл. 2.3.

Таблица 2.3 – Сферы применения генетического алгоритма для решения задач транспортных компаний

Решаемая задача	Особенности применения
Оптимизация маршрутов	может использоваться для нахождения оптимальных маршрутов доставки товаров с учётом различных факторов, таких как время в пути, стоимость перевозки, количество ресурсов
	может оптимизировать уже найденные решения, для минимизации затрат или времени доставки
Распределение складских запасов	может применяться для оптимизации распределения товаров между различными складами, учитывая спрос, расстояния и ёмкости складов
	формирование рекомендаций о том, какие товары и в каком количестве следует переместить для оптимизации общих расходов на хранение и доставку
Управление запасами и заказами	управление запасами и заказами, определение оптимальных уровней запасов для минимизации издержек и удовлетворения потребностей клиентов
	может автоматически определять, когда и сколько товара следует заказывать, исходя из анализа данных о спросе и поставках

2.2.3 Реализация выбранного алгоритма

Как было сказано ранее, настройки генетического алгоритма очень сильно определяют эффективность его работы. Поэтому значения базовых параметров настройки, таких как количество получаемых популяций, вероятность мутации и т.д. должны храниться где-то вне кода основной программы, чтобы, в случае необходимости, они могли быть настроены.

В разрабатываемом приложении все настройки хранятся в файле электронной таблицы «params.xls» и, при запуске приложения, они считываются во внутренние переменные.

```
import pandas as pd

# Открытие файла Excel
file_path = 'params.xls'
xls = pd.ExcelFile(file_path)

# Чтение данных из первого листа
df = pd.read_excel(xls, sheet_name=xls.sheet_names[0])
# Извлечение значений параметров
```

```
population_size = df.iloc[1, 0]
crossover_probability = df.iloc[1, 1]
mutation_probability = df.iloc[1, 2]
num_generations = df.iloc[1, 3]
```

В разрабатываемом приложении используется библиотека pandas. Pandas – это мощная и гибкая библиотека Python, которая предназначена для работы с данными. Она предоставляет удобные структуры данных и инструменты для анализа данных. Pandas поддерживает чтение и запись данных в различные форматы, включая CSV, Excel, JSON и другие.

После загрузки параметров алгоритма требуется посмотреть пользовательские настройки маршрута:

— узнать начальную и конечную точки;

— уточнить, требуется ли посетить какой-либо пункт в определённый момент.

```
import pandas as pd
# Открытие файла Excel
file_path = 'route.xls'
df = pd.read_excel(file_path)

# Извлечение начальной и конечной точки маршрута
start_point = df.iloc[0, 0]
end_point = df.iloc[0, 1]

# Извлечение городов маршрута и их номеров
cities = []
city_numbers = {}
# Начало с третьего столбца (индекс 2) и второй строки (индекс 1)
for col in range(2, df.shape[1]):
     city = df.iloc[0, col]
     if pd.notna(city):
          # если город не пустой
          cities.append(city)
          if len(df) > 1:
               # если есть вторая строка
               city_number = df.iloc[1, col]
               if pd.notna(city_number):
                    # если номер города не пустой
                    city_numbers[city] = city_number
```

Алгоритм получения настроек маршрута пользователем приведен на диаграмме 2.5.

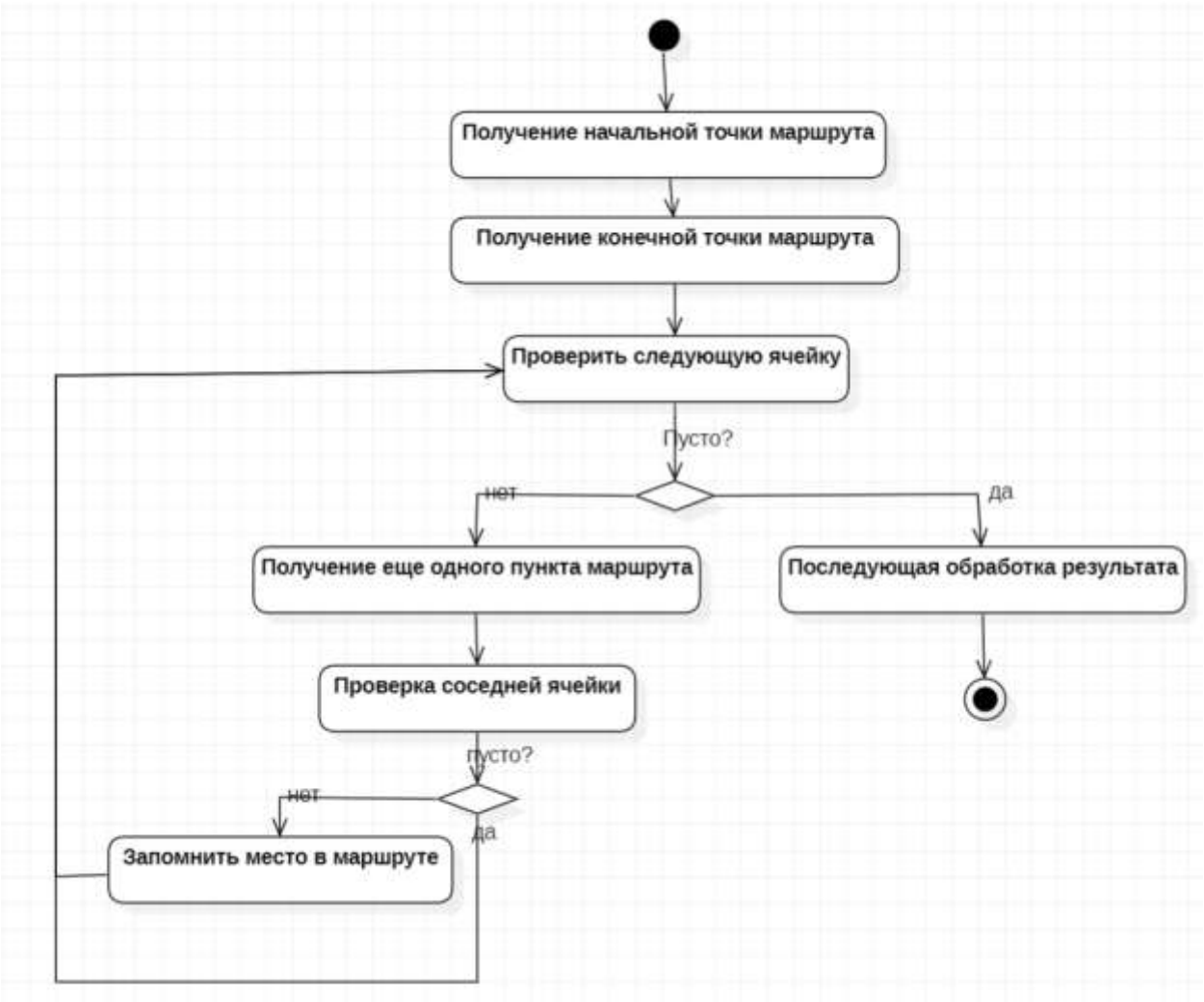


Рисунок 2.5 – Получение данных пользователя о маршруте

Под шагом «Последующая обработка результатов» понимается составление таблицы расстояний между пунктами маршрута.

В распоряжении системы есть таблица с расстояниями между всеми пунктами «distance.xls». В ней содержатся как те пункты, которые вошли в новый маршрут, так и те пункты, которые в него не входят. Поэтому работа с исходной таблицей будет содержать множество действий, которые будут выполняться без особого смысла (например, обход в таблице расстояний тех пунктов, которые не вошли в данный маршрут, но все равно проверяются ввиду алгоритма обхода). Это приведёт к лишним затратам времени, которые вовсе не являются обязательными.

Поэтому более рациональным решением будет получить на основе исходной таблицы расстояний новую таблицу, которая будет содержать в себе только расстояния между пунктами маршрута.

```
distance_file_path = 'distance.xls'
df_distance = pd.read_excel(distance_file_path, index_col=0)

# Создание таблицы расстояний между городами из списка
distance_matrix = pd.DataFrame(index=cities, columns=cities)
for city1 in cities:
     for city2 in cities:
          if city1 != city2:
               distance_matrix[city1,city2]=   df_distance[city1,
city2]

# Заполнение диагонали нулями, так как расстояние от города до
самого себя равно нулю
for city in cities:
     distance_matrix[city, city] = 0
```

После этих подготовительных действий таблица distance_matrix будет содержать расстояния между пунктами маршрута и может быть использована в работе генетического алгоритма.

Сам алгоритм обсудим согласно его шагам, приведённым в п.2.2.2 данной выпускной квалификационной работы.

Первым этапом там являлось создание начальной популяции. В разработанной системе за это будет отвечать функция:

```
def generate_population(population_size, len(city)):
    population = []
    for i in range(population_size):
osob = []
osob.append(start_point)
        j=1
while j!=(len(city)-1):
            tmp = random.randint(1, len(city)
            if  tmp!=start_point  and  tmp!=end_point  and  tmp  not
in osob:
osob.append(random.randint(1, len(city))
            j+=1
osob.append(end_point)
        population.append(osob)
    return population
```

В приведённой выше функции вся исходная популяция хранится в списке. Новая особь представляет собой также список, состоящий из номеров пунктов маршрута.

Также функция создания начальной популяции показывает, что начальная и конечная точки маршрута добавляются в формируемую особь

автоматически, а последующее распределение пунктов происходит случайным образом с помощью библиотеки random.

Библиотека random предоставляет функции для генерации псевдослучайных чисел и выполнения различных операций, связанных со случайным выбором, например выбор случайных элементов из последовательностей, перемешивание данных и создание случайных выборок и т.д.

Также следует отметить, что в алгоритме при создании особи используются не названия пунктов маршрута, а их номера в порядке появления в таблице расстояний distance_matrix.

Далее для каждой из полученных особей находится значение функции приспособленности, которая рассчитывает длину маршрута, закодированного в данной особи.

```
def fit(solution, distance_matrix):
     total_distance = 0
     num_cities = len(popualtion)
     # Проходим по всем парам городов в маршруте
     for i in range(num_cities - 1):
          from_city = popualtion[i]
          to_city = popualtion[i + 1]
          total_distance += distance_matrix[from_city, to_city]

     # Добавляем расстояние от последнего города до первого
(замыкаем маршрут)
    total_distance+=distance_matrix[popualtion[-1],popualtion[0]]
     return total_distance
```

Сначала происходит инициализация переменной total_distance, которая будет содержать в результате общую длину маршрута. Далее осуществляется проход по всем парам городов в маршруте и соответствующие расстояния добавляются из distance_matrix. После маршрут «замыкается»: добавляется расстояние от последнего города до первого, поскольку транспорт должен вернуться на исходную позицию.

Все расстояния, полученные в результате работы функции fit, сохраняются в отдельном массиве, чтобы пройти этап селекции.

На этапе селекции в исходной популяции отбираются особи с наилучшими результатами функции fit, т.е. с наименьшими значениями длин маршрутов.

```
import heapq
def find_n_smallest_elements(array, n):
     if n <= 0:
          return []
     return heapq.nsmallest(n, array)
```

Модуль heapq предоставляет функции для работы с кучами (heaps), которые являются эффективными структурами данных для поиска минимальных или максимальных элементов.

Функция heapq.nsmallest принимает два аргумента: количество наименьших элементов, которые нужно найти, и сам массив. Она возвращает список из n наименьших значений в массиве.

После этапа селекции наступает этап скрещивания особей. Вероятность скрещивания задаётся одним из параметров системы и определяет следующие характеристики алгоритма (рис. 2.6).

Рисунок 2.6 — Влияние вероятности скрещивания на результат работы генетического алгоритма

Существует несколько вариантов скрещивания особей в рамках концепции генетического алгоритма (рис. 2.7).

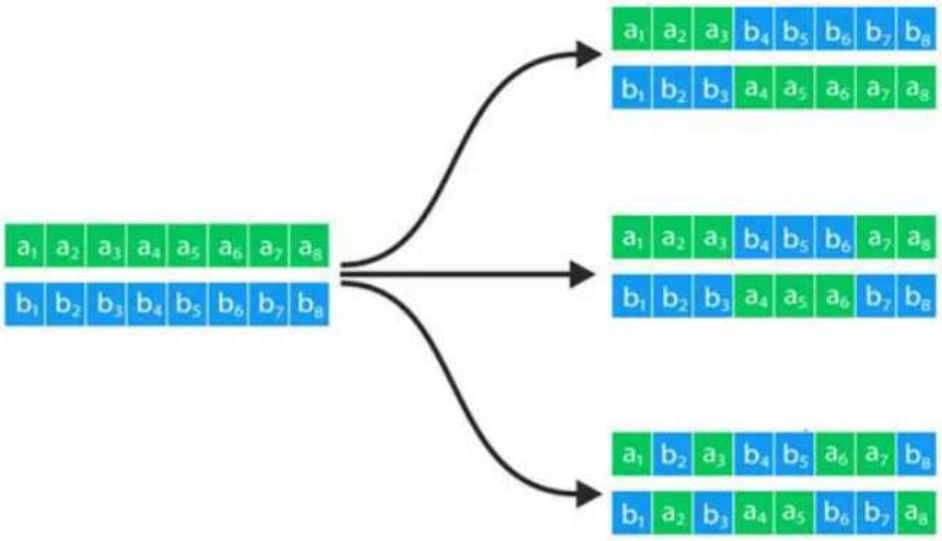


Рисунок 2.7 — Варианты скрещивания особей

Это одноточечное скрещивание (первая ситуация), двухточечное скрещивание (вторая ситуация) и многоточечное скрещивание (третий случай).

Для разрабатываемой системы выберем одноточечное скрещивание.

```
def crossover(parent1, parent2, crossover_probability):
    if random.random() < crossover_probability:
        crossover_point = random.randint(1, len(parent1)-1)
        child1 = parent1[:crossover_point] +
parent2[crossover_point:]
        child2 = parent2[:crossover_point] +
parent1[crossover_point:]
        return child1, child2
    else:
    return parent1, parent2
```

Этот вариант скрещивания предполагает, что в произвольном месте одной из особей-родителей выбирается случайным образом позиция, и две дочерних особи получаются из двух частей каждого из родителей, скопированных до выбранного ранее места разделения.

Мутация для особи в задаче выбора оптимального пути состоит в том, что в произвольном месте в особи выбирается позиция, и два соседних пункта маршрута меняются местами.

Предложенный алгоритм мутации требует обязательной проверки, что начальный и конечный пункты маршрута не изменились.

В разработанной системе за данный вариант мутации отвечает соответствующая функция.

```
def mutate(osob, mutation_probability):
    position = random.random(len(osob)-1)
    tmp = osob[:position-1]+osob[position+1]
    tmp+= osob[position] + osob[position+2:]
    return tmp
```

Полученная после мутации популяция снова отправляется на этап селекции, и далее цикл повторяется ещё несколько раз.

Какое число популяций, которые должны быть созданы для получения решения, близкого к оптимальному, – это один из нетривиальных вопросов данного алгоритма.

Выбор числа популяций в генетическом алгоритме зависит от нескольких факторов, среди которых как сложность задачи, так и требования к точности решения. Оптимальное количество поколений часто определяется эмпирически и может варьироваться в зависимости от конкретного случая.

Основные критерии, которые влияют на выбор числа популяций в генетическом алгоритме, приведены на рис. 2.8.

Рисунок 2.8 — Критерии определения числа популяций

Одним из основных улучшений генетического алгоритма, предложенным в рамках разрабатываемой системы, является возможность закрепить за какими-либо пунктами маршрута, помимо первого и последнего, их позиции.

Это улучшение очень востребовано для решения логистических задач, поскольку при осуществлении реальных доставок, такие ограничения постоянно присутствуют. Например, требуется сначала отвезти скоропортящийся товар или какой-то из пунктов маршрута после определённого времени не будет работать, поэтому эти пункты нужно посетить как можно раньше.

Алгоритм реализации этого улучшения в рамках одной популяции приведён на рис. 2.9.

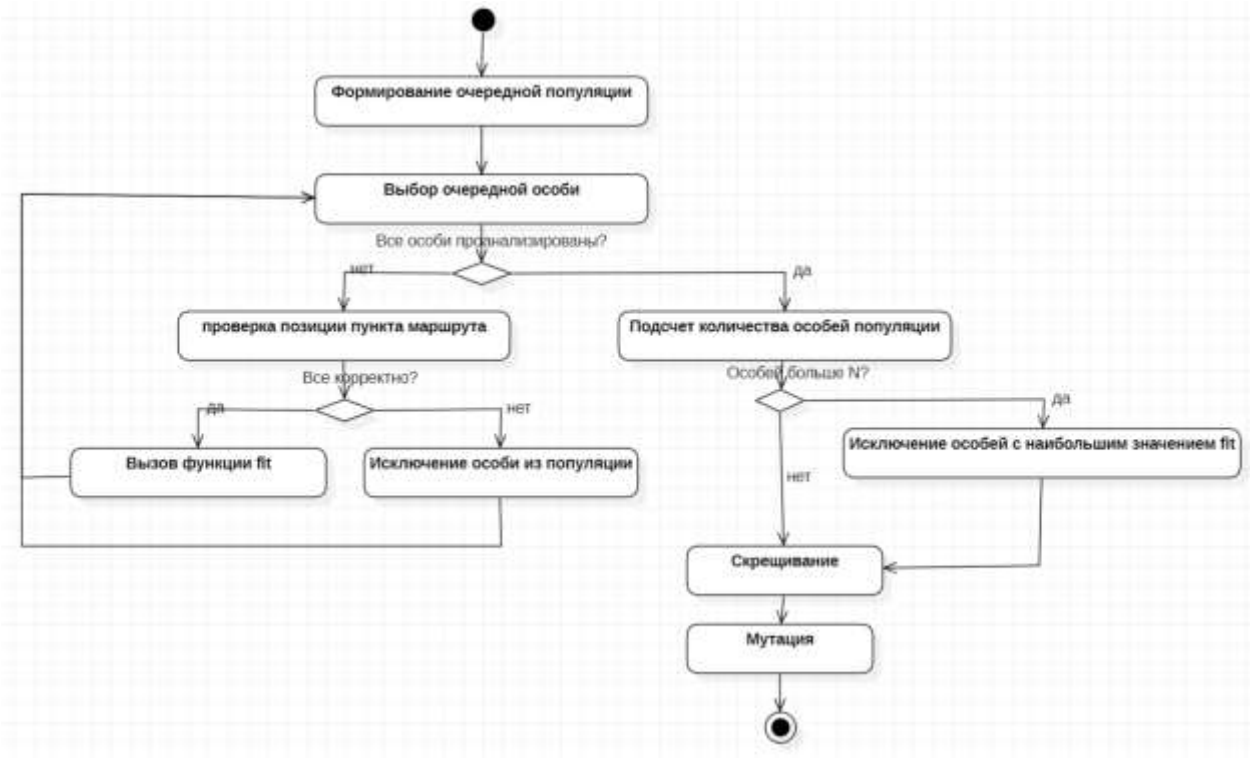


Рисунок 2.9 — Предлагаемое улучшение генетического алгоритма

То есть, в предложенном улучшении селекция осуществляется в два этапа:

— сначала каждая особь проверяется исходя из требования к позиции пункта в маршруте,

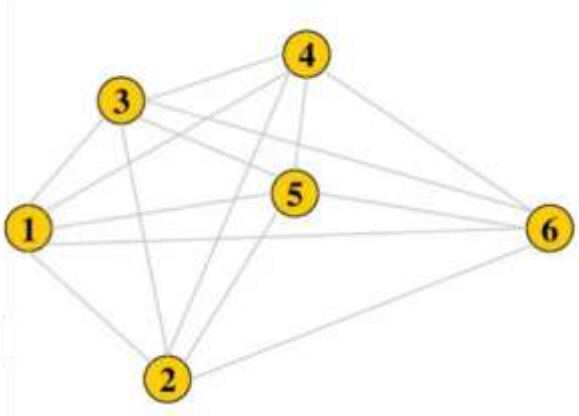
— Далее для отобранных особей находится значение функции fit и, если необходимо, селекция продолжается классическим способом.

2.3 Анализ эффективности работы системы

Для того, чтобы оценить эффективность работы системы и её способность за ограниченное число популяций найти оптимальное решение, проведём анализ её работы и основных показателей.

Для примера дадим системе маршрут следующего вида (рис. 2.10).

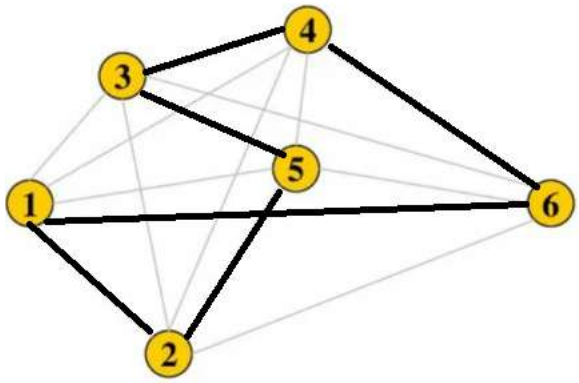
Для проверки попробуем с помощью реализованной системы найти расстояние от пункта 1 до пункта 6.



C_{ij}	1	2	3	4	5	6
1		6	4	8	7	14
2	6		7	11	7	10
3	4	7		4	3	10
4	8	11	4		5	11
5	7	7	3	5		7
6	14	10	10	11	7	

Рисунок 2.10 — Представление заданных дорог и пунктов назначения в виде графа и матрицы смежности

Предложенная задача имеет известное точное решение, представленное на рис. 2.11.



C_{ij}	1	2	3	4	5	6
1		6	4	8	7	14
2	6		7	11	7	10
3	4	7		4	3	10
4	8	11	4		5	11
5	7	7	3	5		7
6	14	10	10	11	7	

Рисунок 2.11 — Точное решение задачи

Проведём тестирование разработанной системы. В качестве количества элементов в популяции выберем 5 (поскольку задача не очень большая), количество популяций установим в 10.

Результаты работы системы приведены на рис. 2.12.

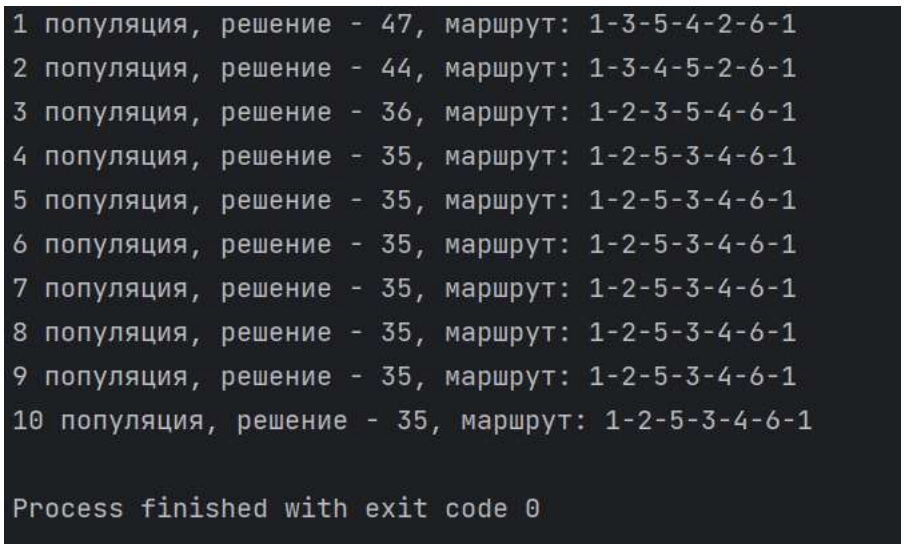


Рисунок 2.12 — Результаты работы системы

Как видно из представленного результата, уже на четвёртой популяции система нашла оптимальное решение и далее оно передавалось из популяции в популяцию.

Теперь рассмотрим представленное в разработанной системе улучшение алгоритма и скажем, что пункт с номером 4 обязательно должен быть посещён вторым по счёту.

Результаты тестирования системы на данных условиях приведены на рис. 2.13.

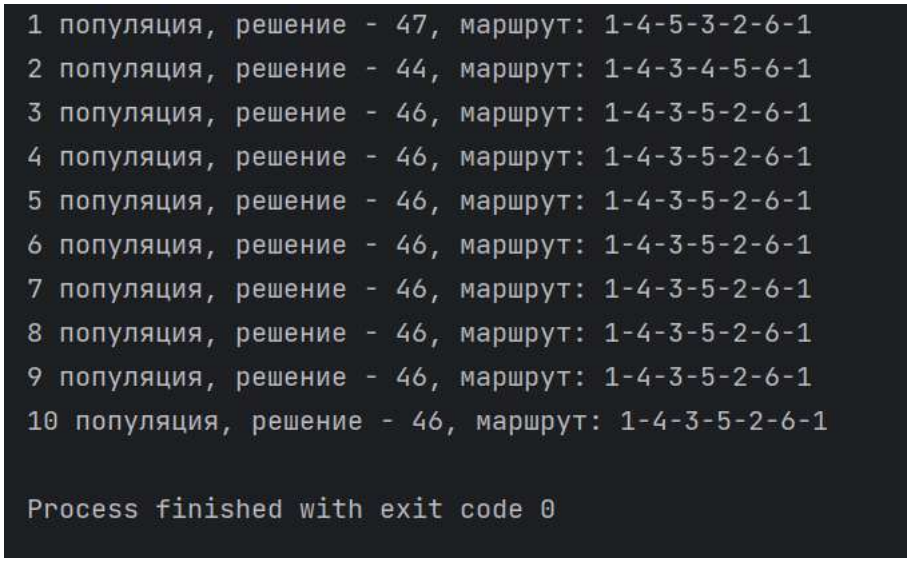


Рисунок 2.13 —Результаты работы системы с учётом зафиксированного пункта

Представленный результат, позволяет увидеть, что уже на третьей популяции система нашла оптимальное решение. Более быстрый поиск решения связан с тем, что в данном случае было наложено большее число

ограничений, следовательно, алгоритм, лежащий в основе разработанной системы был менее свободен в вариантах формирования особей.

2.3 Руководство пользователя

Как было сказано ранее, ООО «Ника» – это одна из компаний-долгожителей рынка. Любые нововведения, ввиду особенностей персонала и руководства компании, внедряются с некоторыми сложностям. Поэтому одной из задач разработанной системы было максимальное удобство использования для сотрудников компании, которые достаточно инертны к переменам.

Поэтому для удобства их работы было принято решение вносить все данные, используемые с системе в электронные таблицы.

Первая из таблиц, которая упоминалась в п. 2.2.3 данной выпускной квалификационной работы – это таблица «params.xls», содержащая в себе настройки системы (рис. 2.14):

	A	B	C
1	Размер популяции	11	
2	Вероятность скрещивания	1	
3	Вероятность мутации	0.008	
4	Число поколений	10	
5	Уровень селекции	50	
6			
7			

Рисунок 2.14 — Таблица настроек

Еще один файл, структура которого подробно была рассмотрена в п.2.2.3 – это файл «distance.xls», один из листов которого содержит полную матрицу расстояний (рис. 2.15).

A	B	C	D	E	F	G	H	I	J
	свх. Масловский	Репное	Новая Усмань (база)	Новожив отинное	Старожи вотинное	Архангельское	Рогачевка	Нечаевка	Рамонь (склад)
свх. Масловский	0	24	13	51	52	42	21	9.8	23
Репное	24	0	15	32	31	61	46	12	41
Новая Усмань (база)	13	15	0	45	44	57	27	16	54
Новоживотинное	51	32	45	0	1	91	72	38	17
Староживотинное	52	31	44	1	0	83	64	29	25
Архангельское	42	61	57	91	83	0	62	56	102
Рогачевка	21	46	27	72	64	62	0	24	81
Нечаевка	9.8	12	16	39	29	56	24	0	75
Рамонь (склад)	23	41	54	17	25	102	81	75	0
Рамонь (база)	23	39	52	11	19	100	79	73	2

Рисунок 2.15. — Таблица расстояний

На другом листе сотрудник ООО «Ника» может указать данные маршрута, который должен быть построен.

Начальная и конечная точки маршрута пользователем выбираются из выпадающего списка (рис. 2.16).

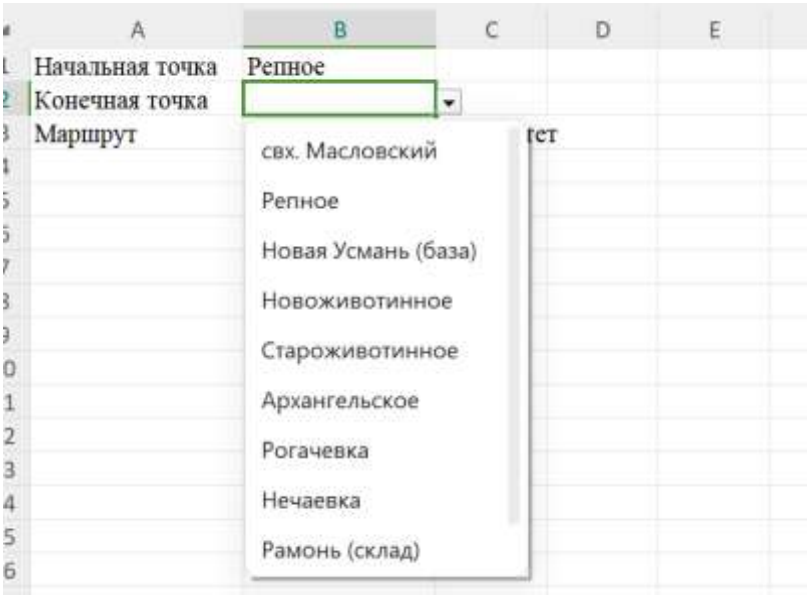


Рисунок 2.16 — Выбор начальной и конечной точек маршрута

Далее аналогичным образом необходимо выбрать все пункты, входящие в маршрут (рис. 2.17).

A	B	C	D
Начальная точка	Репное		
Конечная точка	Староживотинное		
Маршрут	Пункт	Приоритет	
	свх. Масловский		
	Новая Усмань (база)	2	
	Рогачевка		
	Нечаевка		

Рисунок 2.17 — Выбор пунктов в маршруте

Если рядом с каким-либо пунктом стоит приоритет, то это значит, что пункт в маршруте будет зафиксирован, как приведено в примере. Если приоритет не проставлен, но маршрут будет строиться только с учётом начального и конечного пунктов.

Если в списке пунктов маршрута добавится новый пункт, то он автоматически добавится и в выпадающий список (рис. 2.18).

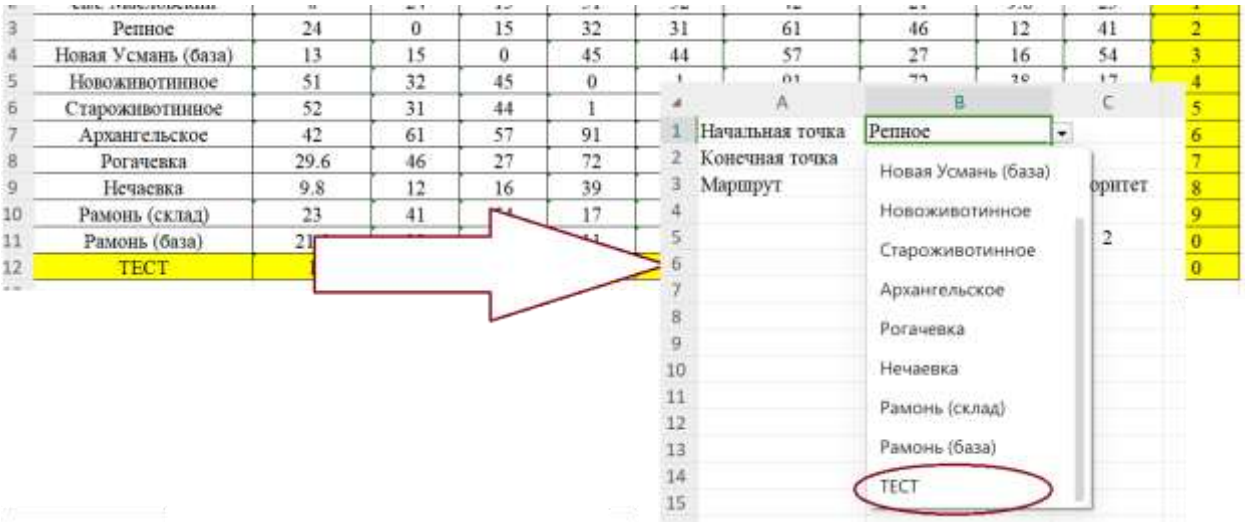


Рисунок 2.18 — Появление пункта в списке после добавления его в таблицу

Далее, сотруднику требуется запустить exe-файл с автоматизированной информационной системой.

Результаты планирования маршрута будут сохранены в ту же книгу, куда вводились требования к маршруту на новый лист (рис. 2.19).

A	B	C
Маршрут	144	км
Репное		
Новая Усмань (база)		
свх. Масловский		
Рогачевка		
Нечаевка		
Староживотинное		

Рисунок 2.19— Таблица, полученная в результате работы системы

Возможно, работа с системой с помощью электронных таблиц может показаться несколько необычной, но данный способ получения данных от пользователя является наиболее органичным, исходя из характеристик исследуемой компании и её персонала. Все сотрудники ООО «Ника» хорошо умеют пользоваться электронными таблицами, поэтому для них данный вариант является наиболее предпочтительным.

Автоматизированная информационная система не обязательно должна иметь интерфейс для пользователя. Наличие интерфейса зависит от целей и особенностей применения системы.

ЗАКЛЮЧЕНИЕ

Современный мир совершенно немыслим без логистики разного уровня.

Крупные компании строят большие логистические центры, чтобы доставлять свои товары в максимально возможное число конечных пунктов. Курьеры, занимающиеся доставкой небольших заказов постоянно встречаются то тут, то там.

Логистика и ее задачи настолько плотно вошли в жизнь современного человека, что стали уже рутинными и незаметными.

Однако, когда что-то идёт не по плану, и доставка срывается, как конечные пользователи, так и грузоотправители и иные посредники замечают это сбои и негативно реагируют на них. Поэтому эффективная доставка, грамотно составленные маршруты и учёт всех деталей – это один из приоритетов компании, которые занимаются грузоперевозками.

В рамках данной выпускной квалификационной работы были решены все задачи, которые ставились в начале её написания.

Во-первых, были изучены основные особенности автоматизированных информационных систем, которые применяются в деятельности транспортных компаний.

Во-вторых, после анализа деятельности ООО «Ника» были сформулированы основные критерии и параметры, влияющие на процесс планирования и составления маршрута, а также была проведена оценка необходимости создания соответствующей автоматизированной информационной системы.

В-третьих, был проведён подробный анализ и оценка существующих программных решений, выделены их особенности и нюансы работы.

Также был проведён анализ языков программирования, которые могут быть использованы для создания автоматизированной информационной системы, определены их особенности и оценены их недостатки.

В-четвёртых, были изучены основные алгоритмы выбора оптимального пути, изучен генетический алгоритма. Было описано и реализовано расширение генетического алгоритма, учитывающее дополнительную логику.

В-пятых, разработанная автоматизированная информационная система была реализована на основе выбранных технологий и инструментов программирования, а после была проведена оценка результатов работы системы.

Задача, поставленная в рамках данной выпускной квалификационной работы была решена в полном объёме. В ходе выполнения поставленных задач учитывались все пожелания руководства ООО «Ника».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеев В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений/ В.Е. Алексеев. – М.: Бином. Лаб. знаний, 2019. – 319 с

2. Ахо Альфред В. Структуры данных и алгоритмы: Вильямс / А. В. Ахо . — М.: Вильямс, 2011. – 382 с.

3. Банокин П.И. Методы и средства проектирования информационных системи технологий: учебное пособие / П.И. Банокин; Томский политехнический университет.– Томск: Изд-во Томского политехнического университета, 2019. – 92 с.

4. Бежанова М. М. Практическое программирование. Структуры данных и алгоритмы/ М. М. Бежанова. – М.: Логос, 2021. – 223с.

5. Бэрри П. Изучаем программирование на Python / П. Бэрри. – М.: Эксмо, 2019.–332 с.

6. Васильев А. Н. Python на примерах. Практический курс по программированию / А.Н. Васильев. – М.: Наука и техника, 2021. – 432 с.

7. Гуриков С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков – М.: Форум, 2018. – 991 с.

8. Евстигнеев, В.А. Применение теории графов в программировании / В.А. Евстигнеев. – М.: Наука, 2023 – 352 с.

9. Златопольский Д. М. Основы программирования на языке Python / Д.М. Златопольский. – М.: ДМК Пресс, 2021. – 277 с.

10. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев. – СПб.: БХВ-Петербург, 2023. – 1104 с.

11. Костюкова, Н.И. Графы и их применение. Комбинаторные алгоритмы для программистов / Н.И. Костюкова. – М.: Интуит, 2017 – 311с.

12. Кураков Л. П., Новые информационные технологии/ Л. П. Кураков – Чебоксары, 2019. – 284 с.

13. Леоненков А. В. Самоучитель UML: учебник/ СПб: БХВ – Петербург, 2022. – 235 с.

14. Липский В. Комбинаторика для программистов/ В. Липский. – М.: Мир, 1988. – 200 с.

15. Лутц М. Изучаем Python том I, II/М. Лутц. – СПБ.: «Вильямс», 2019г. – 852 с.

16. МакГрат М. Python. Программирование для начинающих/ М. МакГрат. – М.: Эксмо, 2023. – 727с.

17. Макконнелл Дж. Основы современных алгоритмов. 2-е изд., доп./ Дж. Макконелл – М.: Техносфера, 2024. – 368 с.

18. Мейер Б.Методы программирования: в 2-х томах/ Б. Мейер. – М.: Мир, 2022. – 356 с.

19. Новиков Ф.А. Дискретная математика для программистов/ Ф. А. Новиков: Учебник для вузов.3-е изд. – СПб. : Питер, 2018. – 384 с.

20. Окулов С.М. Программирование в алгоритмах/ С. М. Окулов. – 3-е изд. – М.: БИНОМ. Лаборатория знаний, 2019. – 383 с.

21. Пападимитриу Х.Комбинаторная оптимизация. Алгоритмы и сложность/ Х.Пападимитриу. – М.: Мир, 2020. – 512 с.

22. Райли Д. Абстракция и структуры данных: Вводный курс/ Д. Райли.М.: Мир, 2023. – 750 с.

23. Рейнгольд Э.Комбинаторные алгоритмы. Теория и практика/ Э. Рейнгольд. – М.: Мир, 2019. – 476 с.

24. Сачков В.Н. Введение в комбинаторные методы дискретнойматематики/ В.Н. Сачков.– М.: Наука, 2022.– 492 с.

25. Свами, М. Графы, сети и алгоритмы / М. Свами. – М.: Мир, 2021. – 455 с.

26. Холл М. Комбинаторика/ М. Холл. – М.: Мир, 1970. – 273 с.

27. Фёдоров Д. Ю. Основы программирования на примере языка Python / Д. Ю. Фёдоров. – СПб.: Юрайт, 2018. – 167 с.

28. Эрик М. Изучаем Python. Программирование игр, визуализация данных, веб-приложения / М. Эрик. – М.: Питер, 2023. – 551 с.

29. Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Vazirani. Algorithms – McGraw-Hill Companies, Incorporated, 2020. – 336 p.

30. Handbook of Theoretical Computer Science. Vol.A: Algorithms andComplexity Theories.– North Holland Publ. Comp., Amsterdam, 2020. – 436 p.

ПРИЛОЖЕНИЕ

```
import math
import pandas as pd

# Функция приспособленности
def fit(solution, distance_matrix):
     total_distance = 0
     num_cities = len(popualtion)
     # Проходим по всем парам городов в маршруте
     for i in range(num_cities - 1):
          from_city = popualtion[i]
          to_city = popualtion[i + 1]
          total_distance += distance_matrix[from_city, to_city]

     # Добавляем расстояние от последнего города до первого
(замыкаем маршрут)
    total_distance+=distance_matrix[popualtion[-1],popualtion[0]]
     return total_distance

# Создание начальной популяции
def generate_population(population_size, chromosome_length):
population = []
    for i in range(population_size):
osob = []
osob.append(start_point)
        j=1
while j!=(len(city)-1):
            tmp = random.randint(1, len(city)
            if tmp!=start_point and tmp!=end_point and tmp not
in osob:
osob.append(random.randint(1, len(city))
            j+=1
        osob.append(end_point)
        population.append(osob)
    return population


# Расчет значения функции приспособленности для каждого
кандидата в популяции
def evaluate_population(population):
    fitness_values = []
    for chromosome in population:
        x = decode_chromosome(chromosome)
        fitness_values.append(fitness_func(x))
    return fitness_values

# Декодирование хромосомы в значение x
def decode_chromosome(chromosome):
    n = len(chromosome)
    decimal_value = 0
    for i in range(n):
        decimal_value += chromosome[i] * (2**(n-i-1))
    x = decimal_value / (2**n-1) * 0.31
```

```
    return x

# Турнирная селекция
def tournament_selection(population, fitness_values,
tournament_size):
    tournament = random.sample(range(len(population)),
tournament_size)
    winner_index = tournament[0]
    for i in tournament[1:]:
        if fitness_values[i] > fitness_values[winner_index]:
            winner_index = i
    return population[winner_index]

# Кроссинговер
def crossover(parent1, parent2, crossover_probability):
    if random.random() < crossover_probability:
        crossover_point = random.randint(1, len(parent1)-1)
        child1 = parent1[:crossover_point] +
parent2[crossover_point:]
        child2 = parent2[:crossover_point] +
parent1[crossover_point:]
        return child1, child2
    else:
        return parent1, parent2

# Мутация
def mutate(chromosome, mutation_probability):
    position = random.random(len(osob)-1)
    tmp = osob[:position-1]+osob[position+1]
    tmp+= osob[position] + osob[position+2:]
    return tmp

# Генетический алгоритм
def genetic_algorithm(population_size, chromosome_length,
tournament_size, crossover_probability, mutation_probability,
num_generations):
    population = generate_population(population_size,
chromosome_length)
    for i in range(num_generations):
        fitness_values = evaluate_population(population)
        parents = [tournament_selection(population,
fitness_values, tournament_size) for i in range(population_size)]
        offspring = []
        for j in range(0, population_size-1, 2):
            parent1 = parents[j]
            parent2 = parents[j+1]
            child1, child2 = crossover(parent1, parent2,
crossover_probability)
            child1 = mutate(child1, mutation_probability)
            child2 = mutate(child2, mutation_probability)
            offspring.append(child1)
            offspring.append(child2)
        population = offspring
```

```
        best_fitness = max(fitness_values)
        best_chromosome =
population[fitness_values.index(best_fitness)-1]
        best_x = decode_chromosome(best_chromosome)
        print(f"Поколение {i+1}:")
        print(f"Лучшее решение: x = {best_x}, f(x) =
{best_fitness}")
        print("Хромосомы:")
        for chromosome in population:
            x = decode_chromosome(chromosome)
            fitness = fit(x)
            print(f"{chromosome} -> x = {x}, f(x) = {fit}")

# Открытие файла Excel
file_path = 'params.xls'
xls = pd.ExcelFile(file_path)

# Чтение данных из первого листа
df = pd.read_excel(xls, sheet_name=xls.sheet_names[0])
# Извлечение значений параметров
population_size = df.iloc[1, 0]
crossover_probability = df.iloc[1, 1]
mutation_probability = df.iloc[1, 2]
num_generations = df.iloc[1, 3]

file_path = 'route.xls'
df = pd.read_excel(file_path)

# Извлечение начальной и конечной точки маршрута
start_point = df.iloc[0, 0]
end_point = df.iloc[0, 1]

# Извлечение городов маршрута и их номеров
cities = []
city_numbers = {}
# Начало с третьего столбца (индекс 2) и второй строки (индекс 1)
for col in range(2, df.shape[1]):
     city = df.iloc[0, col]
     if pd.notna(city):
          # если город не пустой
          cities.append(city)
          if len(df) > 1:
               # если есть вторая строка
               city_number = df.iloc[1, col]
               if pd.notna(city_number):
                    # если номер города не пустой
                    city_numbers[city] = city_number

distance_file_path = 'distance.xls'
df_distance = pd.read_excel(distance_file_path, index_col=0)

# Создание таблицы расстояний между городами из списка
distance_matrix = pd.DataFrame(index=cities, columns=cities)
```

```
for city1 in cities:
     for city2 in cities:
          if city1 != city2:
               distance_matrix[city1,city2]=   df_distance[city1,
city2]

# Заполнение диагонали нулями, так как расстояние от города до
самого себя равно нулю
for city in cities:
     distance_matrix[city, city] = 0

# Запускаем генетический алгоритм
genetic_algorithm(population_size, chromosome_length,
crossover_probability, mutation_probability, num_generations)
```