Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

УДК 004.02

**Отчет по лабораторной работе**

**“Задача коммивояжера”**

Выполнил:

студент группы РИС-23-1б

Марданов Р. Е.

Проверил:

доцент кафедры ИТАС

Петренко А. А.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_gjdgxs)

[ВВЕДЕНИЕ 3](#_1fob9te)

[1. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА 4](#_3znysh7)

[2. АЛГОРИТМ РЕШЕНИЯ 9](#_2et92p0)

[3. ВИЗУАЛИЗАЦИИ 11](#_sttbc9oau3db)

[ЗАКЛЮЧЕНИЕ 12](#_tyjcwt)

[ПРИЛОЖЕНИЕ А 13](#_fjf07d33fzh)

[ПРИЛОЖЕНИЕ Б 14](#_obd5s7qt67k9)

# ВВЕДЕНИЕ

Задача коммивояжера (Traveling Salesman Problem, TSP) является одной из классических задач комбинаторной оптимизации. В данной задаче необходимо найти кратчайший путь, проходящий через все города из заданного списка ровно один раз и возвращающийся в исходный город. TSP имеет огромное практическое значение в логистике, транспорте, проектировании микросхем, а также в других областях, где требуется оптимизация пути или маршрута.

Решение задачи коммивояжёра также является сложной задачей, требующей применения различных оптимизационных и алгоритмических подходов. Существует множество методов для решения TSP, включая методы полного перебора, приближенные алгоритмы, генетические алгоритмы, муравьиные алгоритмы и другие.

Одним из первых, кто сформулировал ранний вариант этой задачи, был выдающийся ирландский математик, физик и механик Уильям Роуэн Гамильтон. Ученый изучал симметрии икосаэдра (двадцатигранника) и в 1857 году предложил игру , цель которой заключалась в прохождении всех вершин додекаэдра (двенадцатигранника) ровно по одному разу и только по его ребрам, с последующим возвратом в отправную точку. Иными словами нужно было найти так называемый гамильтонов цикл на графе.

Исследование задачи коммивояжёра позволяет не только находить оптимальные или приближенные решения для конкретных ситуаций, но и изучать общие принципы оптимизации пути и маршрутов. Решение TSP также имеет важное значение в разработке алгоритмов для других комбинаторных задач и служит основой для изучения сложных оптимизационных проблем в области искусственного интеллекта и логистики.

Данная лабораторная работа заключается в анализе методов решения задачи коммивояжёра, исследовании их эффективности, изучении возможных оптимизаций, а также она включает в себя поиск наиболее эффективной визуализации решения. Исследование эффективности алгоритмов коммивояжёра требует изучения их скорости работы, точности результата, способности к решению больших задач и т.д.

Таким образом, выполнение данной лабораторной работы способствует развитию не только теоретических знаний, но и практических навыков в области разработки и оптимизации алгоритмов.

Объект исследования: вычислительный процесс нахождения оптимального пути методом ветвей и границ.

Предмет исследования: программа-приложение, определяющая кратчайший путь с помощью метода ветвей и границ для решения задачи коммивояжера.

Цель: реализация программы на языке программирования С++ для решения задачи коммивояжёра, выполнение визуализации решения.

Задачи:

1. Провести анализ алгоритмов решения;
2. Разработать кода на языке программирования С++;
3. Разработать визуализацию решения задачи коммивояжёра.

# 1. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА

Перед тем как решать задачу коммивояжера, мы должны проанализировать все возможные методы решения, выявить их достоинства и недостатки, быстроту, доступность(простоту). Для этой задачи были разработаны следующие алгоритмы:

**1.Полный перебор:**

Полный перебор (или метод «грубой силы») — метод решения задачи путем перебора всех возможных вариантов. Сложность полного перебора зависит от количества всех возможных решений задачи. Если пространство решений очень велико, то полный перебор может не дать результатов в течение нескольких лет или даже столетий.

**Анализ:**

Плюсы:

Гарантированное нахождение оптимального решения: Полный перебор рассматривает все возможные варианты, поэтому гарантированно найдет наилучшее решение.

Простота реализации: Алгоритм полного перебора обычно прост в реализации, поскольку он прямолинеен и не требует сложных эвристик.

Минусы:

Экспоненциальная сложность: С увеличением числа вариантов решения время работы алгоритма возрастает экспоненциально.

Неэффективность для больших задач: Для задач с большим пространством решений полный перебор может быть непрактичным из-за его огромных вычислительных требований.

**2.Случайный перебор:**

Обычно выбор решения можно представить последовательностью выборов. Если делать эти выборы с помощью какого-либо случайного механизма, то решение находится очень быстро, так что можно находить решение многократно и запоминать «рекорд», т. е. наилучшее из встретившихся решений. Этот наивный подход существенно улучшается, когда удается учесть в случайном механизме перспективность тех или иных выборов, т. е. комбинировать случайный поиск с эвристическим методом и методом локального поиска. Такие методы применяются, например, при составлении расписаний для Аэрофлота.

**Анализ:**

Плюсы:

Быстрота: Случайный перебор может быстро находить приемлемые решения, особенно если пространство решений велико.

Простота: Алгоритм случайного перебора прост в реализации.

Возможность улучшения: Случайный перебор можно улучшить, добавив эвристики и методы локального поиска, что делает его более эффективным.

Минусы:

Не гарантирует оптимальное решение: Случайный перебор не гарантирует нахождения наилучшего решения, он может найти только «хорошее» решение.

Зависимость от случайности: Качество решения зависит от случайного выбора, что может привести к непостоянству результатов.

Трудность выбора подходящей эвристики: Для эффективного случайного перебора необходимы хорошо подобранные эвристики, которые могут быть сложными в разработке.[2]

3.**Метод ветвей и границ**:

Метод ветвей и границ предложен в 1963 году группой авторов Дж. Литлом, К. Мурти, Д. Суини, К. Кэролом. Широко используемый вариант поиска с возвращением, фактически является лишь специальным частным случаем метода поиска с ограничениями. Ограничения в данном случае основываются на предположении, что на множестве возможных и частичных решений задана некоторая функция цены и что нужно найти оптимальное решение, т.е. решение с наименьшей ценой. Для применения метода ветвей и границ функция цены должна обладать тем свойством, что цена любого частичного решения не превышает цены любого расширения этого частичного решения (Заметим, что в большинстве случаев функция цены неотрицательна и даже удовлетворяет более сильному требованию).

Столь большой успех применения данного метода объясняется тем, что авторы первыми обратили внимание на широту возможностей метода, отметили важность использования специфики задачи и сами воспользовались спецификой задачи коммивояжера.

В основе метода ветвей и границ лежит идея последовательного разбиения множества допустимых решений на подмножества. На каждом шаге метода элементы разбиения подвергаются проверке для выяснения, содержит данное подмножество оптимальное решение или нет. Проверка осуществляется посредством вычисления оценки снизу для целевой функции на данном подмножестве. Если оценка снизу не меньше рекорда — наилучшего из найденных решений, то подмножество может быть отброшено. Проверяемое подмножество может быть отброшено еще и в том случае, когда в нем удается найти наилучшее решение. Если значение целевой функции на найденном решении меньше рекорда, то происходит смена рекорда. По окончании работы алгоритма рекорд является результатом его работы.

Если удается отбросить все элементы разбиения, то рекорд — оптимальное решение задачи. В противном случае, из не отброшенных подмножеств выбирается наиболее перспективное (например, с наименьшим значением нижней оценки), и оно подвергается разбиению. Новые подмножества вновь подвергаются проверке и т.д.[3]

**Преимущества:**

Гарантированное нахождение оптимального решения (при правильной реализации): Метод гарантирует нахождение оптимального решения, если оно существует, при условии правильного определения нижней оценки.

Эффективность: Метод отсекает целые ветви пространства поиска, которые не могут содержать оптимальное решение, что существенно сокращает количество рассматриваемых вариантов.

**Недостатки:**

Зависимость от функции оценки: Эффективность метода сильно зависит от качества функции оценки. Неточная или слишком слабая оценка может привести к рассмотрению большого количества лишних вариантов.

Потенциально большой объем памяти: Метод может требовать значительного объема памяти для хранения информации о ветвях и оценках.

**Вывод:**

Таким образом, для задач с небольшим пространством поиска, где требуется гарантированно найти наилучшее решение, полный перебор может быть вполне приемлемым. Если же пространство поиска обширно и допустимо довольствоваться "хорошим" решением, то случайный перебор, особенно в сочетании с эффективными эвристиками, станет предпочтительным выбором. Для сложных задач оптимизации, требующих оптимального решения и допускающих хорошую функцию оценки, метод ветвей и границ представляет собой наиболее эффективный подход, несмотря на сложность его реализации.

# 2. ИСПОЛЬЗУЕМЫЙ АЛГОРИТМ РЕШЕНИЯ

**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ МЕТОДОМ ВЕТВЕЙ И ГРАНИЦ**

**Алгоритм решения:**

На каждом шаге алгоритма выбирается один город, который включается в маршрут. В результате работы алгоритма будет найден оптимальный маршрут, проходящий через все города ровно один раз и имеющий минимальную стоимость. Сложность алгоритма зависит от количества городов и структуры матрицы расстояний. Сложность задачи коммивояжера, решаемой с помощью метода ветвей и границ, оценивается как полное число порожденных вершин дерева решений, а не фактическое время выполнения алгоритма.

1. Подготовка данных.

Определить количество городов и расстояния между ними.

Построить матрицу расстояний.

1. Определение нижней границы.

Найти минимальную стоимость маршрута, проходящего через все города.

Этот маршрут будет служить нижней границей стоимости всех возможных маршрутов.

1. Разбиение на подмножества.

Выбрать город, который будет начальным.

Разделить множество маршрутов на два подмножества: маршруты, начинающиеся с выбранного города, и маршруты, не проходящие через него.

Для каждого подмножества определить верхнюю границу стоимости маршрута.

1. Решение задачи для подмножеств.

Решить задачу для каждого подмножества отдельно.

Если подмножество содержит только один маршрут, то он будет оптимальным.

Если подмножество содержит несколько маршрутов, то выбрать маршрут с наименьшей стоимостью.

1. Сравнение результатов.

Сравнить стоимость маршрута из подмножества с нижней границей.

Если стоимость меньше или равна нижней границе, то маршрут является оптимальным.

Если стоимость больше нижней границы, то маршрут не является оптимальным и можно исключить его из рассмотрения.

1. Выбор оптимального маршрута.

Из всех маршрутов, удовлетворяющих условию, выбрать маршрут с наименьшей стоимостью.

Этот маршрут будет оптимальным решением задачи коммивояжёра.

1. Завершение работы.

Вывести оптимальный маршрут и его стоимость.

**Вывод:**

Таким образом, чтобы правильно решить задачу коммивояжёра методом ветвей и границ, нужно следовать алгоритму. Написание алгоритма для решения задачи помогает структурировать последующую работу, что облегчает выполнение лабораторной работы.

**2.2 СРЕДА РАЗРАБОТКИ ДЛЯ НАПИСАНИЯ КОДА**

Для того, чтобы написать код следуя выбранному алгоритму, нужно выбрать платформу, на которой будет происходить реализация. Для своей лабораторной работы я буду использовать Qt Creator.

Интегрированная среда разработки Qt Creator является творческой стартовой площадкой, которую можно использовать для редактирования, отладки и сборки кода, а также для публикации приложения. В дополнение к стандартному редактору и отладчику, предоставляемых большинством интегрированных сред разработки, Qt Creator включает компиляторы, средства завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки программного обеспечения.

**Вывод:**

Таким образом, выбор Qt Creator в качестве платформы для своей лабораторной работы, упрощает процесс разработки программного обеспечения. Qt Creator позволит вам эффективно редактировать код, применяя графический интерфейс и удобный инструмент отладки.

# 3. ВИЗУАЛИЗАЦИЯ

Визуализация играет ключевую роль, для интуитивного понятнимания алгоритма. Это позволяет разработчикам анализировать эффективность алгоритмов, оптимизировать производительность. Для визуализации решения задачи коммивояжера был использован Qt Creator - это кроссплатформенная интегрированная среда разработки (IDE), созданная специально для разработки приложений на основе фреймворка Qt. Среда разработки Qt Creator содержит в себе ряд функциональных возможностей, таких как автодополнение кода, статический и динамический анализ кода, интеграцию с системой контроля версий, инструменты для написания юнит-тестов и многие другие.

Одной из ключевых особенностей Qt Creator является интеграция с фреймворком Qt, который облегчает создание кроссплатформенных приложений с использованием графического интерфейса, многопоточности, сетевого взаимодействия и других возможностей, Qt поддерживает разработку приложений для различных платформ, таких как Windows, macOS, Linux, Android и iOS.[4]

Визуализация решения задачи коммивояжёра показана в приложении В.

**Вывод:**

В целом, использование Qt Creator для визуализации решения задачи коммивояжера является наиболее эффективным способом, учитывая возможности этой среды разработки.

# ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы был разработан программный код, реализующий алгоритм решения задачи коммивояжера. Код успешно прошел тестирование и продемонстрировал корректную работу, находя оптимальный маршрут минимальной длины. Для наглядного представления решения была реализована визуализация с использованием среды разработки Qt Creator.

**Вывод:** Разработанное программное обеспечение с успехом решает поставленную задачу и может быть использовано для анализа и оптимизации логистических цепочек и транспортных сетей.

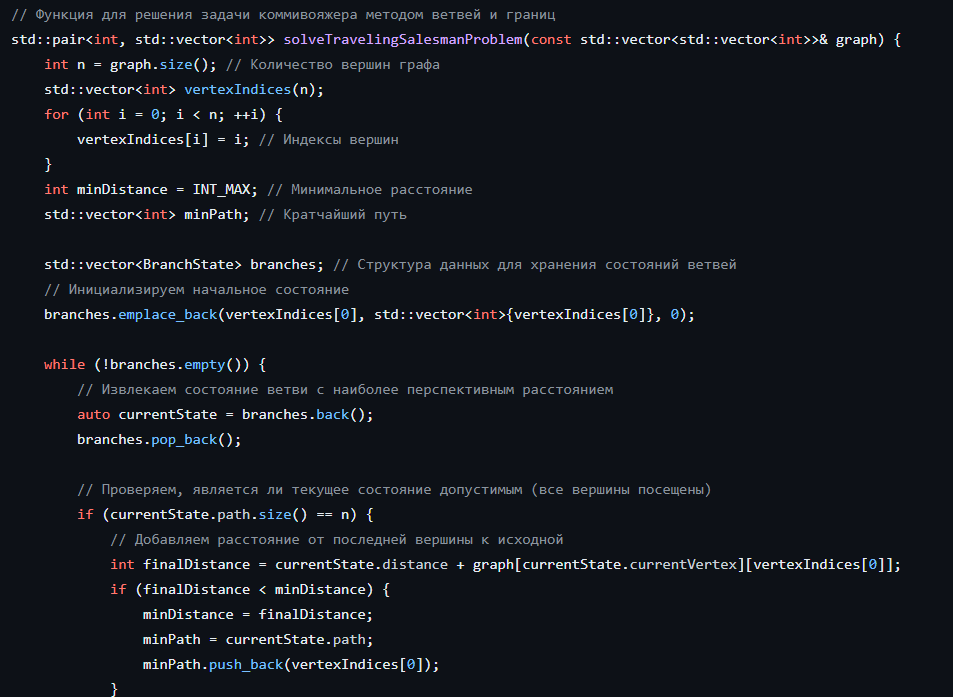
**СПИСОК ИСТОЧНИКОВ**

1. https://en.wikipedia.org/wiki/Travelling\_salesman\_problem
2. https://habr.com/ru/articles/560468/
3. https://visualstudio.microsoft.com/ru/#vs-section
4. https://www.qt.io/product/development-tools

# 

# ПРИЛОЖЕНИЕ А

**Код приложения**



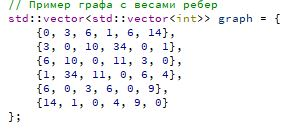


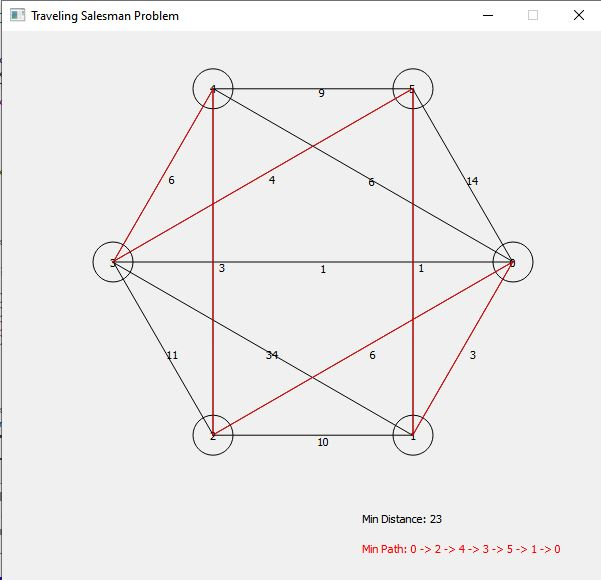
Полный код:

https://github.com/islvld/CW/tree/main/TSPWE

# ПРИЛОЖЕНИЕ Б

**Визуализация**

****

****