



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт (Филиал) № 8 «Компьютерные науки и прикладная математика» Кафедра 806
Группа М8О-410Б-19 Направление подготовки 02.03.02 Фундаментальная
информатика и информационные технологии

Профиль Информатика и компьютерные науки

Квалификация бакалавр

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

На тему: Анализ структурных графов задачи маршрутизации транспортных
средств с различной вместимостью для применения метода попарного
сходства

Автор ВКРБ Верховский Никита Сергеевич ()
(фамилия, имя, отчество полностью)

Руководитель Лемтюжникова Дарья Владимировна ()
(фамилия, имя, отчество полностью)

Консультант ()
(фамилия, имя, отчество полностью)

Консультант ()
(фамилия, имя, отчество полностью)

Рецензент ()
(фамилия, имя, отчество полностью)

К защите допустить

Заведующий кафедрой 806 Крылов Сергей Сергеевич ()
(№ каф) (фамилия, имя, отчество полностью)

_____ 2023г.

Москва 2023

РЕФЕРАТ

Выпускная квалификационная работа бакалавра состоит из 44 страниц, 9 рисунков, 1 таблица, 13 использованных источников, 1 приложения.

CVRP, МЕТОД ПОПАРНЫХ СРАВНЕНИЙ, МЕТРИЧЕСКИЙ ПОДХОД, ГРАФ, NP-ТРУДНАЯ ЗАДАЧА

Цель работы – Реализация комплексного анализа структурного графа в контексте задачи Capacitated Vehicle Routing Problem (CVRP) с применением методологии попарных сравнений для выявления структурных закономерностей и динамики изменений.

Для достижения поставленной цели был проведен анализ актуальной литературы, выявление частных случаев в задаче CVRP, построение алгоритмов оценки изменения целевой функции для данных случаев, численное определение функции попарных сравнений с критерием относительной погрешности.

Основными результатами работы, полученными в процессе разработки, являются алгоритмы нахождения частных случаев изменения целевой функции задачи CVRP, численное определение функции попарных сравнений с критерием относительной погрешности.

Данные результаты разработки предназначены для дальнейшего теоретического исследования метода попарных сравнений в контексте анализа маршрутизации транспортных систем. Кроме того, они предоставляют ценную информацию для транспортных компаний, стремящихся оптимизировать свои маршруты и минимизировать свои затраты.

Применение результатов данной работы позволяет улучшить понимание и прогнозирование структурных изменений в задаче CVRP. Это в свою очередь помогает в определении наиболее эффективных маршрутов для транспортных средств, учитывая их грузоподъемность. В долгосрочной перспективе, это может привести к значительной экономии ресурсов и повышению эффективности транспортных систем. Эти результаты также могут быть использованы для развития новых алгоритмов и методов решения проблемы CVRP.

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	4
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	6
ВВЕДЕНИЕ.....	7
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	9
1.1 Задача маршрутизации транспортных средств с ограниченной вместимостью.....	9
1.2 Подходы к решению задач CVRP.....	10
2 МЕТОДОЛОГИЯ И ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ЗАДАЧИ CVRP С ИСПОЛЬЗОВАНИЕМ МЕТОДА ПОПАРНЫХ СРАВНЕНИЙ.....	15
2.1 Теоретическое основание метода попарного сходства и формулировки задач CVRP.....	15
2.2 Стек используемых технологий.....	26
2.3 Процессная структура исследования.....	28
3 ПРАКТИЧЕСКАЯ ЧАСТЬ.....	31
3.1 Результаты работы.....	31
ЗАКЛЮЧЕНИЕ.....	42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	43
ПРИЛОЖЕНИЕ А Исходный код программы.....	44

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящей выпускной квалификационной работе бакалавра применяются следующие термины с соответствующими определениями:

Алгоритм — это набор инструкций или правил, которые следуют для решения задачи или достижения определенного результата. В контексте CVRP, алгоритмы могут использоваться для определения оптимальных маршрутов для транспортных средств

Генетический алгоритм — это поисковый алгоритм, основанный на механизмах естественного отбора и генетики. Они часто используются для решения оптимизационных и поисковых задач, таких как CVRP

Дерево — в контексте теории графов, дерево является неориентированным графом, в котором любые две вершины соединены ровно одной простой цепью. В контексте CVRP, деревья могут использоваться для представления маршрутов или структуры сети

Комбинаторика — это область математики, которая изучает способы счета или перечисления и организации объектов. Комбинаторика часто играет важную роль в задачах оптимизации, таких как CVRP, где требуется исследование различных возможных комбинаций или маршрутов

Линейное программирование — это метод оптимизации, где целевая функция и ограничения являются линейными. Он может быть использован для нахождения оптимального решения в задачах, таких как CVRP, при определенных условиях

Математическое программирование — это техника оптимизации, которая включает в себя нахождение наилучшего возможного решения из множества альтернатив, удовлетворяющих данному набору ограничений. Это является общим термином, который включает в себя различные методы, такие как линейное программирование, нелинейное программирование и т.д

Общий случай — это термин, который обычно используется для описания ситуации или задачи без каких-либо ограничений или специфических условий. В контексте CVRP, общий случай может включать ситуации, когда нет никаких ограничений на количество транспортных средств, их вместимость или количество клиентов, которых они могут посетить

Эвристический алгоритм - это приближенный алгоритм, используемый для решения сложных задач, таких как CVRP, где точное решение может быть сложным или невозможным для нахождения в разумные сроки. Эвристики предоставляют практические решения, которые достаточно хороши для решения задачи, хотя они могут не гарантировать оптимальное решение

CVRP (Capacitated Vehicle Routing Problem) — это вариация задачи коммивояжера, где каждому транспортному средству назначена определенная вместимость, и он может посетить один или несколько клиентов в одном поездке до возвращения на базу. Задача состоит в определении оптимального маршрута для всех транспортных средств, чтобы минимизировать общее время или расстояние путешествия, соблюдая при этом ограничения по вместимости

NP-полная задача — класс задач принятия решений, ответ на которые можно проверить на правильность за полиномиальное время, но для которых, как предполагается, нет алгоритма, который может решить все случаи этой задачи за полиномиальное время

Solver (решатель) — это инструмент, который используется для определения оптимального решения для различных типов оптимизационных задач, таких как линейное программирование, целочисленное программирование, квадратичное программирование и т. д. В контексте CVRP, Solver может использоваться для определения оптимальных маршрутов для транспортных средств

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей выпускной квалификационной работе бакалавра применяют следующие сокращения и обозначения:

CVRP — Задача маршрутизации транспортных средств с ограниченной вместимостью

PTAS — Полиномиальная аппроксимационная схема

DFS — Алгоритм поиска в глубину

TI — Двухиндексный

ВВЕДЕНИЕ

Актуальность темы данной работы связана с тем, что развитие логистики и планирования маршрутов становится все более важным в связи с ускорением темпов глобализации и увеличением сложности транспортных систем. Особое внимание уделяется таким задачам, как Capacitated Vehicle Routing Problem (CVRP). Важность исследования CVRP обусловлена необходимостью эффективного планирования маршрутов для минимизации общих затрат.

Цель работы – исследовать метод анализа изменения целевой функции задачи CVRP при внесении изменений в структуру графа, основанный на методе попарных сравнений.

Для достижения поставленной цели в работе были решены следующие задачи:

- 1) разработка и анализ метода для оценки изменения целевой функции задачи CVRP при внесении изменений в структуру графа;
- 2) применение предложенного метода на задаче CVRP на дереве и анализ его эффективности при переходе к общему случаю задачи CVRP;
- 3) проведение численных экспериментов для проверки эффективности предложенного метода.

Работа основывается на применении теории графов, методе попарного сходства для анализа изменения структуры графа при добавлении ребер. Также используется компьютерное моделирование и численные методы для проведения экспериментов и анализа полученных данных.

В ходе исследовательского процесса, основанного на строгости и методологической корректности, были достигнуты следующие результаты:

- 1) Исследована актуальность применения метода попарного сходства для анализа задачи CVRP.
- 2) Выявлена и формализована закономерность, которая устанавливает корреляцию между внедрением дополнительных ребер в граф задачи CVRP и корректировкой его структуры, в результате которой наблюдаются значимые изменения в целевой функции.
- 3) Проведены обширные численные эксперименты, результаты которых служат убедительным подтверждением теоретически сформулированных предположений и выводов. С применением точных вычислительных методов и современных компьютерных технологий было продемонстрировать полное соответствие эмпирических данных полученным теоретическим прогнозам.

- 4) Открытия и выводы, полученные в ходе данного исследования, открывают новые горизонты для совершенствования алгоритмов решения задачи CVRP. Это, в свою очередь, может стать причиной радикального улучшения эффективности операций в области логистики и управления цепями поставок, ведущему к снижению затрат и повышению уровня обслуживания клиентов.

Результаты работы предназначены для применения в области логистики и планирования маршрутов, в частности, для оптимизации маршрутов доставки. Результаты могут быть также полезны в области теоретической информатики для дальнейшего исследования задачи CVRP и разработки новых методов ее решения.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Задача маршрутизации транспортных средств с ограниченной вместимостью

Задача маршрутизации транспортных средств с ограниченными объемами (CVRP) является одной из классических задач комбинаторной оптимизации. В этой задаче, имеется набор клиентов с определенными запросами на доставку товаров, и набор транспортных средств (обычно грузовых автомобилей), которые должны быть использованы для доставки этих товаров. Каждый клиент имеет известный вес груза и расстояние до склада, и каждое транспортное средство имеет ограниченную грузоподъемность и объем. Цель состоит в том, чтобы определить оптимальный маршрут для каждого транспортного средства, учитывая ограничения грузоподъемности и объема.

Основные ограничения в задаче CVRP включают следующее:

- 1) каждый клиент должен быть доставлен ровно один раз;
- 2) каждое транспортное средство может использоваться только один раз;
- 3) объем и грузоподъемность каждого транспортного средства ограничены;
- 4) расстояние между каждой парой клиентов известно.

Решение этой задачи имеет большое значение для логистических компаний и других организаций, занимающихся доставкой грузов. Одной из основных причин актуальности решения задачи CVRP является экономический эффект, который может быть получен от оптимизации доставки грузов. Оптимизация маршрутов и распределение грузов между транспортными средствами может привести к сокращению затрат на топливо и транспортировку грузов, что существенно снизит общую стоимость доставки. Более того, оптимизация маршрутов и распределение грузов может улучшить качество обслуживания клиентов, ускорив доставку грузов и сократив время доставки.

Кроме того, существует множество дополнительных факторов, которые делают задачу CVRP актуальной и важной для исследований. Например, растущая глобализация рынков, появление новых технологий, изменения в потребительском спросе и прочие факторы, которые влияют на требования к доставке товаров и услуг. Также стоит отметить, что в последние годы произошел значительный рост объемов доставки товаров в связи с развитием электронной коммерции. Это привело к увеличению числа грузовиков и других транспортных средств, которые необходимы для доставки товаров. В таких условиях решение задачи CVRP становится еще более актуальным и требует разработки новых методов и алгоритмов для оптимизации доставки.

Задача CVRP является NP-полной, то есть время, необходимое для нахождения оптимального решения, быстро возрастает с увеличением размерности задачи. Из-за этого задача CVRP становится более сложной при увеличении числа клиентов и транспортных средств. Из-за NP-трудности задачи CVRP, необходимо использовать эвристические методы для нахождения приближенного решения. Это означает, что найденное решение может не быть оптимальным, но будет близким к оптимальному и выполнимым в практических условиях. В связи с этим, исследования в области нахождения эффективных алгоритмов для решения задачи CVRP продолжаются и имеют большое значение для логистических компаний и других организаций, занимающихся доставкой грузов.

Однако на сегодняшний день не существует аппроксимационного алгоритма для задачи CVRP в общем виде, который гарантировал бы точную нижнюю границу погрешности. Но мы можем рассмотреть задачу CVRP на специальном классе графов "деревья", являющейся также NP трудной, при этом для неё существует множество приближенных алгоритмов с точной нижней границей погрешности.

Учитывая ограничения, связанные с отсутствием точных аппроксимационных алгоритмов для общей задачи CVRP, возникает необходимость исследовать возможность приведения этой задачи к специальному классу графов, такому как "деревья". Это позволит применять множество аппроксимационных алгоритмов с точной нижней границей погрешности для получения более точных результатов решения задачи.

1.2 Подходы к решению задач CVRP

Существует ряд алгоритмов, предлагающих вычислить все возможные решения для данной задачи, пока не будет найден наилучший вариант. Однако, поскольку данная задача относится к классу NP-трудных, точные алгоритмы применяются только для задач с малым количеством входных данных и не могут быть использованы для решения реальных задач.

Рассмотрим метод ветвей и границ [1], который имеет следующие шаги для определения оптимального целочисленного решения:

- 1) найти оптимальное решение модели линейного программирования с релаксацией целых чисел;
- 2) на узле 1 пусть релаксированное решение является верхней границей, а округленное целочисленное решение – нижней границей;

- 3) выбрать переменную с наибольшей дробной частью для ветвления. Создать два ограничения для этой переменной, отражающие значения секционированного целого. Результатом будет новое ограничение и новое ограничение;
- 4) создать два новых узла: один для ограничения и один для ограничения;
- 5) решить свободную модель линейного программирования с новым ограничением, добавленным в каждом из этих узлов;
- 6) расслабленным решением является верхняя граница в каждом узле, а существующее максимальное целочисленное решение (на любом узле) является нижней границей;
- 7) если процесс создает допустимое целочисленное решение с наибольшим значением верхней границы любого конечного узла, достигается оптимальное целочисленное решение. Если возможное целочисленное решение не возникает, переходите от узла с наибольшей верхней границей;
- 8) вернуться к шагу 3. Для модели минимизации расслабленные решения округляются, а верхняя и нижняя границы меняются на противоположные.

Вторым методом рассмотрим метод ветвей с отсечением [2]. Метод основан на следующей идеи. Если нижняя граница значений функции на подобласти дерева поиска больше, чем верхняя граница на какой-либо ранее просмотренной подобласти, то может быть исключена из дальнейшего рассмотрения (правило отсева). Как правило, минимальную из полученных верхних оценок записывают в глобальную переменную. Любой узел дерева поиска, нижняя граница которого больше значения, может быть исключен из дальнейшего рассмотрения. Если нижняя граница для узла дерева совпадает с верхней границей, то это значение является минимумом функции и достигается на соответствующей подобласти.

Эвристические алгоритмы

Эвристические алгоритмы[3] – это алгоритмы, которые будучи основанными на некоем правиле (эвристике), не всегда следующей из строгих математических принципов, в подавляющем большинстве случаев дают решение, близкое к точному. Эвристические алгоритмы делятся на:

- 1) двухфазные алгоритмы;
- 2) конструктивные алгоритмы;
- 3) улучшающие алгоритмы.

Конструктивные алгоритмы – это алгоритмы, которые шаг за шагом выстраивают решение, учитывая общую стоимость, которая получается в ходе решения.

Метаэвристические алгоритмы

Метаэвристические алгоритмы [4], т.е. методы оптимизации, разработанные в соответствии с стратегиями, изложенными в метаэвристической структуре, - как следует из названия, всегда эвристичны по своей природе. Этот факт отличает их от точных методов, которые приходят с доказательством того, что оптимальное решение будет найдено в конечном (хотя и часто непомерно большом) количестве времени. Поэтому метаэвристика разработана специально для того, чтобы найти решение, которое «достаточно хорошо» в вычислительном времени, которое «достаточно мало». В результате они не подвергаются комбинаторному взрыву – явлению, в котором вычислительное время, необходимое для нахождения оптимального решения NP-трудных задач, возрастает как экспоненциальная функция размера задачи.

Список из самых часто используемых метаэвристических алгоритмов:

- 1) муравьиный алгоритм (Ant Algorithm)[5];
- 2) детерминированный отжиг (Deterministic Annealing)[6];
- 3) генетические алгоритмы (Genetic Algorithms)[7];
- 4) поиск с запретами (Tabu Search)[8].

Однако, следует подчеркнуть, несмотря на эффективность данных алгоритмов, они не могут гарантировать точную нижнюю границу погрешности целевой функции. Поэтому рассмотрим аппроксимационные алгоритмы для деревьев, для которых данная граница существует.

Ниже представлены приближенные алгоритмы CVRP на дереве с точной нижней границей погрешности.

Алгоритм Хамагучи и Като для CVRP на дереве

В своей статье[9] Хамагучи и Като предложили приближенный алгоритм для решения задачи маршрутизации транспортных средств с ограничениями по пропускной способности на дереве. Алгоритм опирается на концепцию D-минимальных вершин и итеративно удовлетворяет спрос вершин поддерева.

Вершина v называется D-минимальной, если она D-допустима ($D(T_v) \geq 1$, где T_v - поддерево с корнем в вершине v , и $D(T_v)$ - суммарный спрос вершин в поддереве T_v), но ни одна из ее дочерних вершин не является D-допустимой.

Алгоритм работает следующим образом:

- 1) Определяется маршрут одного или двух транспортных средств, который частично удовлетворяет спрос вершин в поддереве T_v . Для этого применяется одна из двух стратегий в зависимости от ситуации:

- Стратегия 1: Два транспортных средства отправляются к выбранной вершине, каждое из которых обслуживает свою часть спроса.
 - Стратегия 2: Одно транспортное средство отправляется к выбранной вершине и обслуживает спрос до своей максимальной пропускной способности. Эта стратегия может включать разделение спроса одной вершины на несколько частей.
- 2) После выполнения одной из стратегий обновляется оставшийся спрос вершин, которые были посещены в текущем маршруте. Если оставшийся спрос поддерева T_v равен нулю, поддерево T_v и ребро $(parent(v), v)$ удаляются.
 - 3) Алгоритм повторяется с шага 1 до тех пор, пока не будет удовлетворен спрос всех вершин дерева.

Если в какой-то момент не остается D -допустимых вершин, алгоритм завершает работу, поскольку оставшийся спрос может быть удовлетворен, посещая вершины в порядке обхода в глубину.

Таким образом, алгоритм Хамагучи и Като предлагает эффективный приближенный метод для решения задачи маршрутизации транспортных средств с ограничениями по пропускной способности на дереве. Алгоритм позволяет достичь приближения 1.5, в условиях, когда разрешено разделение спроса. Когда разделение спроса не разрешено, использование только стратегии 1 позволяет достичь приближения 2. Авторы также приводят пример, в котором отношение приближения их алгоритма составляет 1.25.

Алгоритм Асано, Като и Кавашимы

В своей статье[10] Асано, Като и Кавашима предложили улучшенный приближенный алгоритм для задачи маршрутизации транспортных средств с ограничениями по пропускной способности на дереве, достигая уровня приближения 1.35078. Основная идея улучшения заключается в использовании операций реформирования, сохраняющих нижнюю границу стоимости и объединении двух или более различных стратегий.

В отличие от предыдущего алгоритма, предложенного Хамагучи и Като, алгоритм Асано, Като и Кавашимы сконцентрирован на поддеревьях с общим спросом, превышающим 2. Алгоритм состоит из нескольких раундов, и на каждом этапе обращает внимание на определенный q -узел u , подготавливая несколько стратегий для (частичного) удовлетворения спроса в поддереве T_u с помощью двух или трех транспортных средств. Лучшая стратегия выбирается на основе наименьшего отношения стоимости маршрутов к уменьшению нижней границы. Когда больше нет ни одного q -узла, алгоритм переходит к

базовому случаю, и аналогично предыдущим случаям применяется стратегия с лучшим отношением.

Таким образом, алгоритм Асано, Като и Кавашимы представляет собой значительное улучшение предыдущего алгоритма Хамагучи и Като и демонстрирует уровень приближения 1.35078 для задачи маршрутизации транспортных средств с ограничениями по пропускной способности на дереве.

Полиномиальная схема приближения (PTAS) для CVRP на деревьях.

Данный алгоритм предполагает построение полиномиальной схемы приближения (PTAS) для задачи о маршрутизации транспортных средств с ограниченной грузоподъемностью (CVRP) на деревьях[11]. PTAS предоставляет решения, которые гарантированно близки к оптимальному решению, в то время как вычислительное время остается полиномиальным от размера задачи.

Алгоритм подходит как для версии CVRP с единичным требованием, так и для разбиваемой версии CVRP на деревьях. В версии с единичным требованием каждый терминал имеет единичное требование, которое удовлетворяется одним туром, в то время как в разбиваемой версии каждый терминал имеет положительное целочисленное требование, которое может быть удовлетворено несколькими турами.

PTAS достигает приближения $(1 + \epsilon)$, где ϵ можно сделать произвольно малым, без ослабления ограничений задачи. Это представляет собой значительное улучшение по сравнению с предыдущими алгоритмами приближения, которые достигали постоянного фактора приближения. Кроме того, алгоритм работает для всего диапазона грузоподъемности туров, что является значительным прогрессом по сравнению с предыдущими PTAS для CVRP, ограниченными малыми грузоподъемностями.

2 МЕТОДОЛОГИЯ И ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ЗАДАЧИ CVRP С ИСПОЛЬЗОВАНИЕМ МЕТОДА ПОПАРНЫХ СРАВНЕНИЙ

2.1 Теоретическое основание метода попарного сходства и формулировки задач CVRP

Поскольку метод попарного сходства представляет собой развитие метрического подхода, необходимо вначале охарактеризовать основы метрического подхода.

Рассмотрим концепцию метрического подхода, начав с универсального определения для задач, на которых строится его основная идея.

Определение 5.1. Мастер-задачами по точности назовем такие мастер-задачи $P'(\alpha, A)$, в которых выявлена определенная аналитическая закономерность α , и точное решение которых возможно получить с использованием полиномиального или псевдополиномиального алгоритма A . Так как закономерность может быть не единственной, следует определить множество мастер-задач.

Определение 5.2. Множеством мастер-задач P обозначим совокупность всех мастер-задач $P'(\alpha, A)$, известных для данной модели задачи. Учитывая, что речь идет о некоторой закономерности в данных α , для ясности сформулируем понятие вырожденной мастер-задачи.

Определение 5.3. Вырожденной мастер-задачей по точности назовем мастер-задачу по точности $P(\alpha, A)$, в которой аналитическая закономерность α соответствует нулю соответствующего задачного пространства, то есть все параметры задачи равны нулю.

Метрический подход сформулирован следующим образом:

Определение 5.4. Метрическим подходом назовем отображение $M: P' \rightarrow P$, для которого точное решение задачи P' является приближенным решением для задачи P с точностью до значения метрики $\rho(P', P)$. В настоящее время концепция метрического подхода состоит в следующем. Допустим, даны две задачи: мастер-задача по точности P' и рабочая задача P . С использованием метрики для рабочей задачи подбирается такая мастер-задача по точности P' , что значение метрики для $\rho(P, P')$ минимально. Для задачи P' определена закономерность α , и получено точное решение с помощью (псевдо-)полиномиального алгоритма A .

Построим модель для данной рабочей задачи P :

- 1) определим значимое описание для рабочей задачи P ;

- 2) построим в пространстве задачи некоторую дополнительную задачу P_1 для построения корректной метрики $\rho(P, P_1)$ (опускается, если метрика для данной задачи была построена ранее);
- 3) построим метрику $\rho(P, P_1)$ для данной задачи согласно алгоритму построения метрики;
- 4) определим множество мастер-задач P , соответствующих данной модели;
- 5) скорректируем метрику $\rho(P, P_1)$ $\rho(P, P')$ от рабочей задачи P , до мастер-задач из множества $P' \in P$ исходя из значимости параметров задачи P' ;
- 6) вычислим значение метрики $\rho(P, P')$ от рабочей задачи P , до каждой из мастер-задач из множества $P' \in P$;
- 7) вычислим минимальное значение метрики $\rho(P, P')$ от рабочей задачи P , до мастер-задач из множества $P' \in P$;
- 8) определим алгоритм AP' , соответствующий ближайшей мастер задаче P' по метрике $\rho(P, P')$;
- 9) вычислим решение XP' мастер-задачи P' с помощью алгоритма AP' , а также значение соответствующей целевой функции CP' ;
- 10) подставим полученные значения в неравенства и определим пределы значения целевой функции исходной задачи CP ;
- 11) уточним полученное решение с помощью процедуры постоптимального анализа, если это подразумевается согласно использованию алгоритма построения метрики $\rho(P, P')$ (опускается, если процедуры постоптимального анализа не предусмотрено).

Введем формальные определения, необходимые для построения метода попарных сравнений.

Определение 1. Специальным случаем $P \alpha$ будем называть подкласс исходной NP трудной задачи P , где для всех примеров задач $A \in P \alpha$ выполняется закономерность α .

Определение 2. Простым специальным случаем $P_{\alpha light}$ будем называть такой специальный случай, для которого существует A такой, что для всех примеров задач $A \in P_{\alpha light}$ с помощью алгоритма A может быть получено точное решение.

Определение 3. Почти простым специальным случаем $P_{s light}$ будем называть такой специальный случай, для которого существует полиномиальный (или

псевдополиномиальный) алгоритм, находящий приближенное решение с гарантированной погрешностью для всех примеров задач $A \in P_{slight}$.

Определение 4. Сложным специальным случаем $P_{\alpha hard}$ будем называть такой специальный случай, для которого не существует полиномиального (или псевдополиномиального) алгоритма, находящего точное решение для всех примеров задач $A \in P_{\alpha hard}$. Далее под специальными случаями, если не указано иное, будем понимать простые, почти простые и численно простые специальные случаи. Определим функцию попарных сравнений.

Определение 5. Функцией попарных сравнений будем называть такую функцию, которая зависит от параметров двух примеров задач и позволяет оценить, насколько данные примеры задач отличаются друг от друга.

Далее в качестве критерия сравнения τ будет использоваться абсолютная погрешность целевой функции примера A . Для этого для примера B должно быть получено точное решение. На место B будем подставлять примеры специальных случаев, для которых можно получить точное решение с помощью полиномиального или псевдополиномиального алгоритма. Шаги метода попарных сравнений описаны ниже. Заметим, что критериями сравнения примеров τ в методе попарных сравнений могут использовать различные свойства, такие как относительная погрешность, скорость сходимости метода, оценка сходимости нескольких методов, количество операций и т.д. Перейдем к основным определениям с учетом того, что рассматриваемая задача является задачей линейного программирования с одной целевой функцией. В методе попарных сравнений решение X_B примера B , полученного с помощью полиномиального (или псевдополиномиального) алгоритма, применяется к примеру A , возможно не допускающему такого решения. В зависимости от того, как будет применяться полученное решение X_B , существуют три стратегии применения полученного решения:

- 1) Оценка примера A : использование полученного решения X_B для оценки решения X_A , без учета того, насколько полученное решение допустимо. Для реализации этой стратегии не нужно проверять полученное решение на допустимость.
- 2) Релаксация примера A : использование полученного решения X_B для оценки решения X_A , с учетом того, какие именно ограничения примера A нарушаются, если в качестве его решения используется X_B . Для реализации данной стратегии необходимо проверить полученное решение на допустимость. То есть если пример B является релаксацией примера A , то необходима процедура проверки

удовлетворения ограничений для того, чтобы определить, каким именно ограничениям примера A не удовлетворяет полученное решение XB .

- 3) Допустимое решение XB для примера A : использование полученного решения XB в качестве приближенного решения примера A . Для этого необходимо проверить решение XB на допустимость и если какие-то ограничения были нарушены — найти ближайшее к полученному новое решение X^*A , для которого все ограничения будут удовлетворены. Таким образом, чтобы построить вектор решений XA для задачи A , то необходимо построить функцию, преобразующую XB в XA .

Таким образом метод попарных сравнений состоит из следующих шагов.

- 1) Найти один или несколько специальных случаев задачи. Для многих задач дискретной оптимизации определены специальные случаи. Далее представлены рекомендации, по нахождению новых специальных случаев.
- 2) Построить функцию попарных сравнений, которая позволяет оценивать расстояние от исходной задачи до каждого из возможных специальных случаев. Далее представлен алгоритм для нахождения функции попарных сравнений.
- 3) Найти специальный случай, до которого расстояние от исходного примера задачи будет минимальным согласно значению функции попарных сравнений.
- 4) Найти решение наиболее близкого примера из множества примеров, которые являются специальным случаем.
- 5) Оценить гарантированную погрешность полученного решения с помощью соответствующей функции попарных сравнений.
- 6) Выбрать стратегию применения полученного решения к исходному примеру задачи:
 - Для стратегии «Оценка примера A » использовать полученное решение без корректировки.
 - Для стратегии «Релаксация примера A » использовать полученное решение с учетом информации о том, нарушены ли ограничения исходного примера, и если да, то какие именно.
 - Для стратегии «Допустимое решение примера A » проверить полученное решение на допустимость, и если какие-то ограничения нарушены, то использовать функцию, преобразующую XB в XA .

Нахождение специальных случаев является нетривиальной задачей, поскольку на практике исследователи выделяют такие подзадачи в отдельный класс в процессе работы с

некоторой задачей дискретной оптимизации, опираясь на специальные свойства данного класса. Например, очевидно, что для классов задач, где граф ограничений является деревом, существуют быстрые точные алгоритмы, поскольку они базируются на классических алгоритмах обхода деревьев. Тем не менее, расширение множества специальных случаев позволяет более точно определить абсолютную погрешность исходного примера. Далее приведем несколько рекомендаций, которые позволяют находить специальные случаи. Наиболее очевидной рекомендацией видится расширение существующих специальных случаев за счет незначительного изменения исходной закономерности, приводится пример такого расширения за счет добавления дополнительных ребер в деревья с образованием циклов.

Вторая рекомендация основывается на следующем утверждении. Если для задачи P существует некоторый приближенный алгоритм A^{\sim} , то среди примеров задачи P можно выделить подмножество примеров P^* , для которого с помощью этого алгоритма будет получено точное решение. Для реализации этой рекомендации необходимо построить некоторое количество случайно выбранных примеров задачи P с различными параметрами и решить их с помощью точного алгоритма A и с помощью приближенного алгоритма A^{\sim} . Далее следует выделить подмножество примеров $P^* \subset P$, для которого приближенным алгоритмом A^{\sim} получены точные решения (или решения с гарантированной погрешностью). Заметим, что подмножество примеров P^* по определению является численно простым специальным случаем. Затем стоит определить, существует ли подмножество примеров $P^{**} \subseteq P^*$, для которого можно определить закономерность α . Если закономерность α удастся найти, то полученное множество примеров P^{**} по определению является простым или почти простым специальным случаем.

Третья рекомендация базируется на комбинаторном алгоритме A , для которого исследуется скорость сходимости. Если для задачи P можно выделить подмножество примеров P^* , для которого комбинаторный алгоритм будет сходиться достаточно быстро, то можно построить эвристический алгоритм A^{\sim} на основе алгоритма A , с помощью которого для этого подмножества примеров будет получено точное решение или приближенное решение с гарантированной погрешностью. Заметим, что понятие быстрой сходимости алгоритма в данном контексте используется неформально, поскольку формализация зависит от конкретной задачи. Для реализации этой рекомендации также необходимо построить некоторое количество случайно выбранных примеров задачи P с различными параметрами, а затем решить их с помощью точного алгоритма A . Далее следует выделить подмножество примеров $P^* \subset P$, для которого точным алгоритмом A точные решения получены достаточно быстро. Затем необходимо построить

эвристический алгоритм на основе шагов точного алгоритма, которые осуществляются в ходе решения подмножества примеров P^* . В результате мы получим, что P^* является численно простым специальным случаем. Далее по аналогии со второй рекомендацией осуществим поиск закономерности α для подмножества примеров $P^{**} \subseteq P^*$. И если такую закономерность удастся обнаружить, то P^{**} является простым специальным случаем.

Далее перейдем к построению функции попарных сравнений:

- 1) найти специальный случай P^* ;
- 2) зафиксировать все параметры задачи P^* , кроме одного параметра q ;
- 3) решить полученное множество примеров задачи P^* для различных q ;
- 4) определить закономерность βq между изменением значения параметра q и изменением точного значения целевой функции;
- 5) построить оценку закономерности βq в виде функции попарных сравнений
- 6) Повторить шаги (2)–(5) для других параметров задачи P ;
- 7) построить сумму функций попарных сравнений ρ для всех возможных изменений параметров задачи P .

Заметим, что в силу трудоемкости задачи функция попарных сравнений не всегда может быть найдена. В таком случае предлагается оценить функцию попарных сравнений численно.

Для применения метода попарного сходства требуется точное решение задачи CVRP (Capacitated Vehicle Routing Problem). Достижение этого результата возможно через использование разнообразных математических формулировок. Вместе с тем, прежде чем приступить к их рассмотрению, необходимо подробнее описать задачу CVRP.

Пусть $G = (V, E)$ - связный граф, где $V = \{v_1, \dots, v_n\}$ – множество вершин, и v обозначает вершину, в которой расположен депо; $E = \{(v_i, v_j)\}$ – множество дуг. Пусть l_{ij} – длина дуги (v_i, v_j) , а $q_i \geq 0$ – целочисленный вес, связанный с вершиной v_j ($q_1 = 0$). В депо находится флот однородных автомобилей с грузоподъемностью Q ($Q \geq q_i, i = 1, \dots, n$). Классическая задача CVRP состоит в определении маршрутов сбора грузов автомобилями, начинающихся и заканчивающихся в депо, таким образом, чтобы:

- 1) вес, связанный с любой данной вершиной, собирался ровно одним автомобилем;
- 2) сумма всех собранных весов одним автомобилем не превышала Q ;
- 3) была минимизирована линейная комбинация числа автомобилей и общего расстояния, пройденного этими автомобилями [12].

Формулировки в задаче маршрутизации транспортных средств с вместимостью (CVRP) представляют собой различные математические модели и ограничения, разработанные для представления и решения этой задачи. Цель состоит в том, чтобы спланировать оптимальные маршруты для транспортных средств, чтобы обслужить всех клиентов, минимизируя общие затраты на перемещение транспортных средств. Формулировки задачи CVRP разрабатываются с целью определения структуры задачи, применения ограничений, характеристик маршрутов и спроса клиентов, а также для разработки алгоритмов и методов решения. Различные формулировки могут использоваться в зависимости от специфических требований задачи, сложности и размера проблемы. Существуют различные подходы к решению задачи CVRP, такие как ветвление и границы (branch-and-bound), ветвление и отсечение (branch-and-cut), метаэвристики и т. д. Формулировки задачи CVRP позволяют преобразовать реальную задачу в математическую модель, которую можно решить с помощью оптимизационных алгоритмов или солверов. Они определяют переменные решения и ограничения, которые соответствуют реальным условиям задачи. Каждая формулировка имеет свои особенности, преимущества и недостатки, и выбор подходящей формулировки зависит от специфики задачи и требований к точности и производительности решения.

Two-index vehicle flow formulation

В качестве общей формулировки была выбрана Two-index vehicle flow formulation[11]. В данной формулировке, определена бинарная переменная решения x_{ij} , которая принимает значение 1 тогда и только тогда, когда существует маршрут, который идет от клиента i к j напрямую, для $i, j \in N$. Кроме того, y_j - это непрерывная переменная решения, соответствующая накопленному спросу на маршруте, который посещает узел $j \in N$ до этого визита. С учетом этих параметров и переменных решения, Two-index vehicle flow formulation задачи CVRP задается следующим образом:

$$\text{Minimize} \quad \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij} \quad (1)$$

$$\text{Subject to} \quad \sum_{j=1}^{n+1} x_{ij} = 1 \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{i=0}^n x_{ih} - \sum_{j=1}^{n+1} x_{hj} = 0, \quad h = 1, \dots, n, \quad (3)$$

$$\sum_{j=1} x_{0j} \leq K, \quad i, j = 0, \dots, n + 1, \quad (4)$$

$$y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}), \quad i, j = 0, \dots, n + 1, \quad (5)$$

$$d_i \leq y_i \leq Q, \quad i = 0, \dots, n + 1, \quad (6)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, n + 1 \quad (7)$$

Ограничения (2) гарантируют, что все клиенты посещаются ровно один раз. Ограничения (3) обеспечивают корректное движение автомобилей по дугам, заявляя, что если автомобиль прибывает к узлу $h \in N$, то он должен отбыть из этого узла. Ограничение (4) ограничивает максимальное количество маршрутов до K , где K - это количество автомобилей. Ограничения (5) и (6) вместе обеспечивают, что превышение вместимости автомобиля не допускается. Целевая функция определяется (1) и требует минимизации общей стоимости путешествий по маршрутам.

Преимущества двухиндексной потоковой формулировки CVRP включают следующее:

- 1) Полиномиальное количество ограничений: В отличие от других моделей, которые могут требовать экспоненциального числа ограничений, эта модель обладает полиномиальным числом ограничений относительно числа клиентов. Это делает модель более управляемой и менее затратной с точки зрения вычислений.
- 2) Предотвращение подтуринга: Ограничения модели (2.5) помогают избежать подтуринга в решении, то есть циклических маршрутов, которые не проходят через депо. Это важно для обеспечения эффективности маршрутов.

Несмотря на то, что задача CVRP на деревьях является NP-трудной [13], важно рассматривать математические формулировки для этого класса задач. Причина этого заключается в том, что эти формулировки учитывают специфику древовидной структуры и позволяют находить точное решение за меньший период времени по сравнению с общими формулировками задачи.

В контексте данного исследования предлагается также рассмотреть две формулировки, которые широко используются для решения поставленной задачи.

Treeroute Formulation

В данной формулировке[12] используется тот факт, что маршрут автомобиля представляет собой дерево. Ребра в дереве заменяются на дуги, указывающие "вниз", что устраняет неоднозначность в отношении родителя и ребенка у узлов на ребре. При наличии дуги (i,j) i является родителем, а j - ребенком. Строительство маршрута для каждого автомобиля осуществляется "снизу вверх" от каждого обслуживаемого узла к депо. Это приводит к созданию покрывающего поддерева узлов, которое в дальнейшем называется деревом маршрутов соответствующего автомобиля.

Входные параметры:

L_{ij} – расстояние от узла i до узла j ;

D_i – спрос на узле i ;

P_i – родительский узел i ;

V – набор транспортных средств;

N – набор узлов, отличных от депо;

C – вместимость каждого транспортного средства;

$c(i)$ – набор дочерних узлов i ;

$T(i)$ – поддерево с корнем в узле i ;

A – набор всех дуг (направленных вниз) в дереве;

v_{min}^i – нижняя граница количества требуемых транспортных средств для обслуживания всех узлов в $T(i)$.

Теперь зададим формулировку следующим образом:

$$\text{Minimize} \quad 2 \sum_{i=1}^n L_{ij} \sum_{v \in V} x_{ijv} \quad (8)$$

$$\text{Subject to} \quad x_{P_i v} \geq C \quad \forall i \in N \quad (9)$$

$$x_{P_i v} \geq C \quad \forall i, j, k \in N : i < j < k \quad (10)$$

$$\sum_{i \in N} y_{iv} D_i \leq C \quad \forall v \in V \quad (11)$$

$$\sum_{i \in V} y_{iv} = 1 \quad \forall i \in N \quad (12)$$

$$z_i + \sum_{j=i+1}^n x_{ij} = 1 \quad \forall i \in N \quad (13)$$

$$y_{iv}, x_{ijv} \in \{0, 1\} \quad \forall (i, j) \in A, v \in V \quad \forall i, j \in N: i < j \quad (14)$$

Целью является минимизация взвешенной суммы по всем дугам числа автомобилей, использующих каждую дугу. Ограничение (11) гарантирует, что если дуга (i, j) входит в дерево маршрутов автомобиля, то уникальная предшествующая дуга (P_i, i) также должна быть в дереве маршрутов. Ограничение (12) обеспечивает, что для обслуживания определенного узла автомобиль должен пройти по уникальной дуге, ведущей к этому узлу. Ограничение (13) является ограничением на вместимость автомобиля. Ограничение (14) требует, чтобы спрос на каждом узле был полностью удовлетворен ровно одним автомобилем.

Следует отметить, что переменные x могут быть ослаблены без влияния на целостность решения; однако исходные вычисления показывают, что проблема решается быстрее, если все переменные определены как бинарные. Также стоит отметить, что число целочисленных переменных в формулировке "дерево маршрутов" составляет $O(|N||V|)$, а количество ограничений - $O(|N||V|)$.

Depth First Ordered Formulati

Данная формулировка основана на использовании следующей леммы.

Лемма: Если предположить, что узлы в дереве проиндексированы в порядке обхода в глубину, то, определив набор узлов на маршруте автомобиля, можно получить маршрут с минимальной стоимостью, посещая узлы в порядке возрастания их индекса.

Далее, предполагается, что обход выполняется в порядке обхода в глубину. Введено понятие покрывающего поддерева S . В таком поддереве $T(i)$ представляет собой поддерево S с корнем в узле i . Для любого i из S узел с наименьшим индексом в $T(i)$ - это i (основываясь на порядке обхода узлов в глубину).

В каждом i из S , автомобиль входит в поддерево $T(i)$ и обслуживает все узлы в $T(i)$. Основное наблюдение состоит в том, что как только автомобиль покидает $T(i)$, он больше не возвращается, так как узлы обслуживаются в порядке возрастания их индекса. Следовательно, для всех i в S , ребро от i к его родителю проходит не более двух раз - один раз в поддерево и один раз из него. Это позволяет формировать допустимый маршрут, длина которого в два раза превышает вес покрывающего поддеревя.

Поскольку длина полученного маршрута совпадает с нижней границей, можно утверждать, что данный маршрут является оптимальным. Маршрут для каждого автомобиля включает путь от депо до первого (с наименьшим индексом) узла на маршруте автомобиля, набор дуг, соединяющих узлы на маршруте автомобиля в порядке возрастания индекса узла, и путь от последнего (с наименьшим индексом) узла на маршруте обратно в депо.

Входные параметры:

D_i – спрос на узле i ;

L_{ij} – кратчайшее расстояние между узлами i и j ;

S_i – кратчайшее расстояние от узла i до депо;

N – набор всех узлов, за исключением депо;

C – вместимость каждого автомобиля;

V_{min_k} – нижняя граница количества автомобилей, необходимых для обслуживания узлов.

Теперь сформулируем формулировку следующим образом, предполагая, что узлы индексируются по глубине первого порядка:

$$\text{Minimize } \sum_{i=1}^n S_i(w_i + z_i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^n L_{ij} x_{ij} \quad (15)$$

$$\text{Subject to } D_i + \sum_{j=i+1}^n D_j y_{ij} \leq C \quad \forall i \in N \quad (16)$$

$$y_{ij} + y_{jk} - y_{ik} \leq 1 \quad \forall i, j, k \in N : i < j < k \quad (17)$$

$$x_{ij} - y_{ij} \leq 0 \quad \forall i, j \in N \quad (18)$$

$$w_j + \sum_{i=1}^{j-1} x_{ij} = 1 \quad \forall j \in N \quad (19)$$

$$z_i + \sum_{j=i+1}^n x_{ij} = 1 \quad \forall i \in N \quad (20)$$

$$w_i, x_{ij}, y_{ij}, z_i \in \{0, 1\} \quad \forall i, j \in N: i < j \quad (21)$$

Целью данной формулировки является минимизация дистанции всех пройденных дуг (включая выезд из депо, въезд в депо и все промежуточные дуги). Ограничение (16) представляет собой ограничение на вместимость автомобиля. Несмотря на отсутствие явного понятия автомобиля в этой формулировке, ограничения вместимости учитываются путем суммирования спроса по узлам, которые находятся в одном и том же автомобиле. Ограничение (17) создает клику среди узлов в том же автомобиле - если узлы i и j находятся в том же автомобиле, и узлы j и k находятся в том же автомобиле, то узлы i и k также должны находиться в том же автомобиле. Ограничение (18) обеспечивает, что узел i не может предшествовать узлу j , если они не находятся в том же автомобиле. Ограничения (19) и (20) обеспечивают обслуживание спроса на узле ровно одним автомобилем. Ограничение (21) требует, чтобы все переменные были бинарными. Важно отметить, что переменные x , w и z могут быть расслаблены (то есть $0 \leq x, w, z \leq 1$) без влияния на целостность решения. Однако было замечено, что в практических условиях проблема решается быстрее, если все переменные определены как бинарные. Количество целочисленных переменных в формулировке обхода в глубину составляет $O(|N|^2)$, а количество ограничений - $O(|N|^3)$. Изображение нумерации вершин представлено на рисунке 1

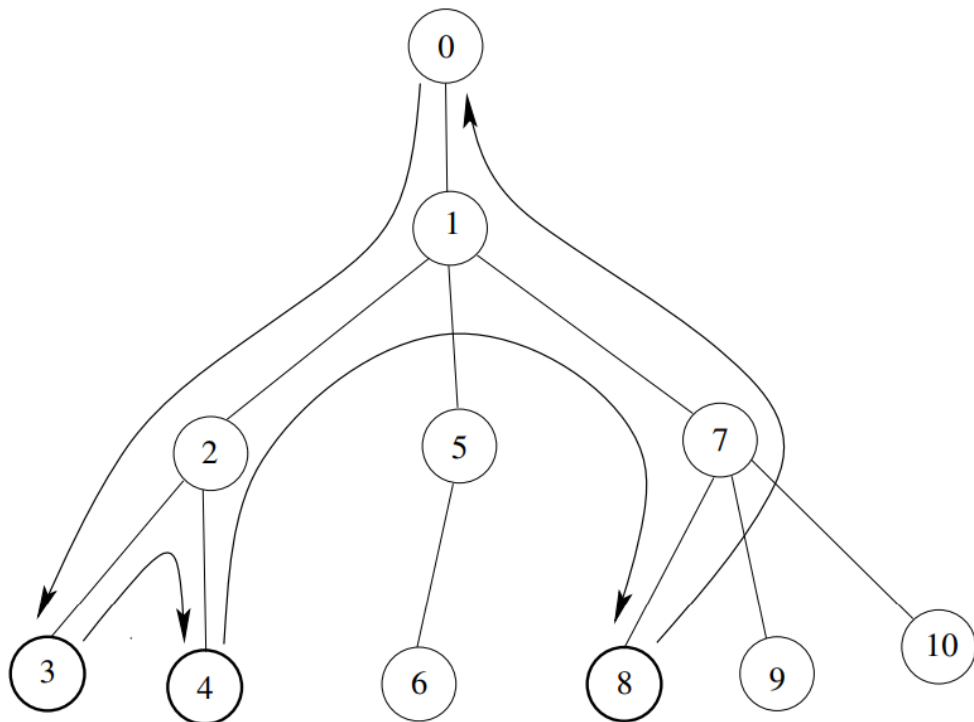


Рисунок 1 – Порядок обхода дерева

2.2 Стек используемых технологий

Задача CVRP является задачей оптимизации, и для ее решения необходимы специализированные инструменты. В этом исследовании были выбраны Gurobi Optimizer и Google OR-Tools как основные инструменты для решения CVRP. Подробнее об этих инструментах и обосновании их выбора можно прочитать ниже.

COPT Optimizer - это высокопроизводительный пакет оптимизации, который предлагает решения для широкого спектра проблем оптимизации. Среди его ключевых преимуществ следует отметить:

- 1) Производительность: COPT известен своей производительностью и способностью решать сложные задачи оптимизации. Он оптимизирован для высокоскоростных вычислений и может обрабатывать большие объемы данных, что идеально подходит для задач, таких как CVRP.
- 2) Точность: COPT имеет высокую точность, что критически важно при решении задач оптимизации, где малейшие различия могут значительно повлиять на результат.
- 3) Гибкость: COPT поддерживает различные методы оптимизации, включая линейное и квадратичное программирование, что позволяет гибко подходить к решению задач.
- 4) Поддержка: COPT имеет активную и крупную общность разработчиков, что облегчает решение возникающих вопросов и проблем.

С другой стороны, Google OR-Tools был выбран в качестве инструмента для проверки формулировок CVRP, написанных для COPT Optimize. OR-Tools - это библиотека открытого исходного кода, которая предлагает решения для широкого диапазона задач оптимизации. Ее ключевые преимущества включают:

- 1) Открытость: Как инструмент с открытым исходным кодом, OR-Tools может быть легко настроен и адаптирован под конкретные нужды исследования.
- 2) Поддержка различных типов задач: OR-Tools поддерживает различные типы задач оптимизации, включая CVRP, что позволяет его использовать для проверки и сравнения результатов.
- 3) Независимая проверка: Использование OR-Tools для проверки формулировок CVRP, написанных для COPT, позволило провести независимую проверку результатов, что увеличило достоверность исследования.

В целом, выбор COPT Optimizer и Google OR-Tools в качестве основного технологического стека был обусловлен их высокой производительностью, точностью, гибкостью, а также поддержкой большого сообщества разработчиков. Это надежные и эффективные инструменты, проверенные временем и рекомендованные экспертами в области оптимизации.

Для работы с технологическим стеком, включающим в себя COPT Optimizer и Google OR-Tools, был выбран Python, как универсальный высокоуровневый язык программирования. Выбор Python обусловлен рядом ключевых преимуществ, которые делают его особенно подходящим для решения задач, связанных с оптимизацией и анализом данных:

- 1) Первое и, возможно, наиболее важное преимущество Python заключается в его глубокой интеграции и совместимости с Gurobi и OR-Tools. Оба инструмента предлагают обширные библиотеки для Python, что позволяет полноценно и эффективно использовать все возможности этих инструментов прямо из Python. Это значительно облегчает процесс разработки и тестирования решений для задач CVRP, делая Python оптимальным выбором для этой работы.
- 2) Второе преимущество Python заключается в его простоте и читаемости. Python предназначен для быстрой и эффективной разработки, его синтаксис прост и понятен, что способствует быстрому написанию и отладке кода. Это крайне важно при работе с сложными задачами оптимизации, такими как CVRP, где более сложные или низкоуровневые языки могли бы усложнить процесс разработки и отладки.
- 3) Третье преимущество Python - это его богатая экосистема научных библиотек, таких как NumPy, SciPy, Matplotlib и многих других. Эти библиотеки облегчают выполнение сложных математических вычислений и визуализацию данных, что является важной частью анализа результатов в задачах оптимизации. В частности, Matplotlib используется для построения графиков и диаграмм, которые необходимы для визуализации и анализа результатов CVRP.
- 4) Четвертое преимущество Python заключается в его популярности в научном сообществе. Python широко используется в области научных исследований, что означает доступность большого количества ресурсов, обучающих материалов и готовых решений, которые могут быть использованы в ходе исследования.

Наконец, гибкость Python как мультипарадигмального языка программирования позволяет адаптировать стиль программирования под конкретные задачи. Python поддерживает процедурное, объектно-ориентированное и функциональное

программирование, что позволяет ему легко адаптироваться под различные задачи и стили программирования.

Таким образом, Python, благодаря своей совместимости с Gurobi и OR-Tools, простоте использования, богатой научной экосистеме, популярности в научном сообществе и гибкости, стал идеальным выбором для этого исследования. Это подтверждает его роль как одного из наиболее востребованных языков программирования для научных исследований и анализа данных.

2.3 Процессная структура исследования

Генерация случайного дерева и n-парного дерева

Первый шаг в исследовании – это генерация деревьев, которые будут использоваться в задачах CVRP.

Использовались два подхода: генерация случайного дерева и генерация n-парного дерева.

Генерация случайного дерева выполняется с помощью самостоятельно разработанного алгоритма, который учитывает такие параметры, как вместимость транспортного средства, интервал вместимости для клиентов, количество потомков у дерева. Генерация n-парного дерева выполняется с использованием библиотеки NetworkX в Python. NetworkX предоставляет инструменты для генерации различных типов графов, включая n-парные деревья.

Реализация формулировок CVRP для деревьев и общей формулировки

После генерации дерева реализуются формулировки CVRP на языке Python с использованием оптимизационного пакета COPT Optimize, специально адаптированные для деревьев, и общая формулировка. Также для проверки корректности решения используется библиотека OR-Tools, которой на вход подаётся матрица расстояний. Это позволяет сравнить результаты, полученные с использованием различных подходов.

Полученные решения, визуализировались в виде графов с помощью инструментов Python для графического представления данных. Это позволяет наглядно проанализировать структуру оптимальных маршрутов и их соответствие условиям задачи.

Для сравнения скорости и точности различных формулировок на одинаковых входных данных была создана таблица результатов целевой функции и времени выполнения. После определения оптимального решения, к дереву добавлялось случайное ребро с минимальным весом, что изменяло класс задачи и приводило к появлению цикла. Дальнейший анализ позволил оценить влияние добавления нового ребра на

оптимальность решения. Данная процедура выполнялась для множества случайных вершин и раскрыла определенные закономерности и зависимости между местом добавления ребра и изменением целевой функции.

Анализ изменений целевой функции при добавлении ребра в дерево и удалении ребра из образовавшегося цикла.

Данная часть исследования основана на применении метода попарных сравнений и охватывает две важные процедуры: добавление ребра в дерево для изменения класса задачи, а затем удаление ребра из образовавшегося цикла. В качестве простого специального случая, необходимого для применения метода попарного сравнения выбрана задача CVRP, граф которой является деревом. Вначале проводилась операция добавления случайного ребра в исходное дерево. Это приводило к формированию цикла и смене класса задачи, что позволяло оценить разность значений результатов между измененным деревом и специальным случаем. Следовательно, анализ изменений целевой функции и общих свойств задачи CVRP после такой операции имеет важное значение.

После того как новое ребро было добавлено и образовался цикл, одно из ребер этого цикла выбиралось случайным образом и удалялось. Эта операция позволяла возвращаться к структуре дерева и анализировать изменения в целевой функции, связанные с данным переходом. Удаление ребра из цикла позволило выявить нижнюю границу изменения целевого значения при переходе от циклической структуры к древовидной. Такой подход помогал оценить, как структурные изменения влияют на оптимальность решения задачи и на целевую функцию.

Важным моментом в ходе эксперимента было проведение достаточного количества итераций для обеспечения репрезентативности получаемых данных.

3 ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1 Результаты работы

Для генерации случайного дерева был реализован алгоритм, который генерирует случайное дерево, начиная с одного узла и добавляя дочерние узлы к существующим узлам, пока не будет достигнуто желаемое количество узлов. Количество дочерних узлов у каждого узла определяется случайным образом.

Алгоритм генерации случайного дерева

На вход подаются следующие параметры:

- 1) $n \in \mathbb{N}$ (количество узлов в графе);
- 2) $c_{min} \in \mathbb{N}$ (минимальное количество дочерних узлов у каждого узла);
- 3) $c_{max} \in \mathbb{N}$ (максимальное количество дочерних узлов у каждого узла);
- 4) $s \in \mathbb{N}$ (зерно для инициализации генератора случайных чисел).

Обозначим R как генератор случайных чисел, инициализированный зерном s . Далее, пусть G обозначает граф, который изначально содержит один узел с индексом 0. Обозначим t как текущий узел, который в начале равен 0. Затем, обозначим c как счетчик узлов, который в начале равен 1. Пусть E - это список узлов, которые нужно расширить. Изначально он содержит один элемент - узел 0.

Алгоритм работает следующим образом:

- 1) Пока E не пуст и c меньше n , продолжаем следующие шаги.
- 2) Выбираем и удаляем первый узел из списка E . Обозначим этот узел как v .
- 3) Генерируем случайное число r в диапазоне от c_{min} до c_{max} (включительно).
- 4) Для каждого числа от 0 до r (не включая r) выполняем следующие действия:
 - Если c достиг или превысил n , прерываем цикл.
 - Добавляем ребро между узлами v и c в граф G .
 - Добавляем c в список E .
 - Увеличиваем c на 1.
- 5) Возвращаем G в качестве результата алгоритма.

Далее задача CVRP была успешно реализована в общей формулировке, а также в двух специальных формулировках, предназначенных для деревьев. Данная работа была выполнена на языке Python с использованием оптимизационного пакета COPT Optimize. Для проверки корректности полученных результатов использовался OR-Tools. Сравнение эффективности работы данных формулировок и OR-Tools представлены на таблице 1.

Таблица 1 – Сравнение эффективности решения задачи CVRP для различных формулировок

n	Вместимость	DFS Formulation	Время(сек.)	Treeroute Formulation	Время(сек.)	OR-Tools	Время(сек.)	TI Formulation	Время(сек.)
20	[1,100]	3232	0.03	3232	0.03	3232	600	3232	2.0
20	[10,10]	2498	0.03	2498	0.03	2498	600	2498	27.0
20	[10,90]	3348	0.03	3348	0.12	3348	600	3348	2.0
40	[10,10]	9376	0.17	9396	500	9376	600	9396	600
40	[10,90]	8700	0.23	8700	21.07	8700	600	8700	600
40	[20,80]	9376	0.15	9376	0.24	9376	600	9376	600
40	[1,50]	6998	0.33	6998	600	6998	600	6998	600
40	[1,30]	6528	0.20	6402	2.96	6402	600	6402	600
40	[1,10]	6284	0.03	5952	0.03	5952	600	5952	600
40	[10,10]	6284	0.09	6078	0.26	6078	600	6078	600
40	[10,10]	8010	0.70	8010	600	8010	600	8010	600
60	[10:10]	12014	0.97	12013	600	12014	600	12013	600
60	[30,70]	15808	0.33	15808	600	15808	600	15808	600
60	[1,31]	10680	220.32	10494	448.24	10494	600	10494	600
60	[1,50]	11898	0.64	12090	600	11898	600	12090	600
60	[1,100]	15354	0.43	15372	600	15354	600	15372	600
60	[20,21]	11226	0.45	11130	600	11130	600	11130	600
90	[1,10]	17062	1.03	14952	1.83	14952	600	14952	600
100	[1,10]	22484	20.7	26718	22.08	22482	600	26718	600
100	[1,30]	18530	2.56	17908	600	17878	600	17908	600
120	[1,10]	21238	600	19210	600.01	19210	600	19210	600
120	[1,10]	25020	600	29220	600	25032	600	25032	600
20	[20,80]	4702	0.2	4702	0.41	4702	600	4702	7.64
20	[1,50]	2864	0.25	2864	0.48	2864	600	2864	600
20	[1,100]	3720	0.3	3720	0.10	3720	600	3720	12.43

В рамках проведенного исследования для всех анализируемых датасетов было установлено значение вместимости транспортного средства Q, равное 100. Этот параметр

был выбран в качестве стандарта для всех рассматриваемых случаев, чтобы обеспечить однородность условий при сравнении результатов и анализе эффективности предложенных решений.

Исходя из полученных численных результатов, можно сделать вывод, что задача CVRP, несмотря на свою NP-трудность в контексте класса графов, типизированных как "деревья", решается значительно быстрее по сравнению с общей постановкой этой задачи. Данный факт подтверждает эффективность предложенного подхода и значимость его внедрения для увеличения скорости решения CVRP в определённых условиях.

Следующим шагом в исследовании было изменение дерева путём добавления случайного ребра с минимальным весом.

В данном исследовании были рассмотрены два ключевых экспериментальных случая для выявления закономерностей между добавлением ребра и изменением целевой функции:

- 1) В первом случае, генерировалось дерево с изменяющимся числом связей в пределах от 1 до 7. Вес каждого ребра, обозначенный как x , выбирался случайным образом из диапазона $\{4, 120\}$. Вес вершин также выбирался случайно из заданного диапазона. Количество вершин в этом случае оставалось фиксированным и составляло $n = 20$.
- 2) Во втором случае, генерировались n -арные деревья с количеством связей, изменяющимся в пределах от 1 до 10. Вес ребер x определялся случайным образом в пределах того же диапазона $\{4, 120\}$. Количество вершин также оставалось фиксированным ($n = 20$).

Для каждого сгенерированного дерева было проведено 10 отдельных экспериментов, в каждом из которых к дереву добавлялось одно ребро. В ходе анализа данных экспериментов было определено значение относительной погрешности: минимальное, изображённое на рисунке 2, среднее значение относительной погрешности функции для 10 экспериментов, изображённое на рисунке 3, максимальное значение, изображённое на рисунке 4. Для визуализации полученных результатов был построен график зависимости относительной погрешности от конкретного экспериментального случая.

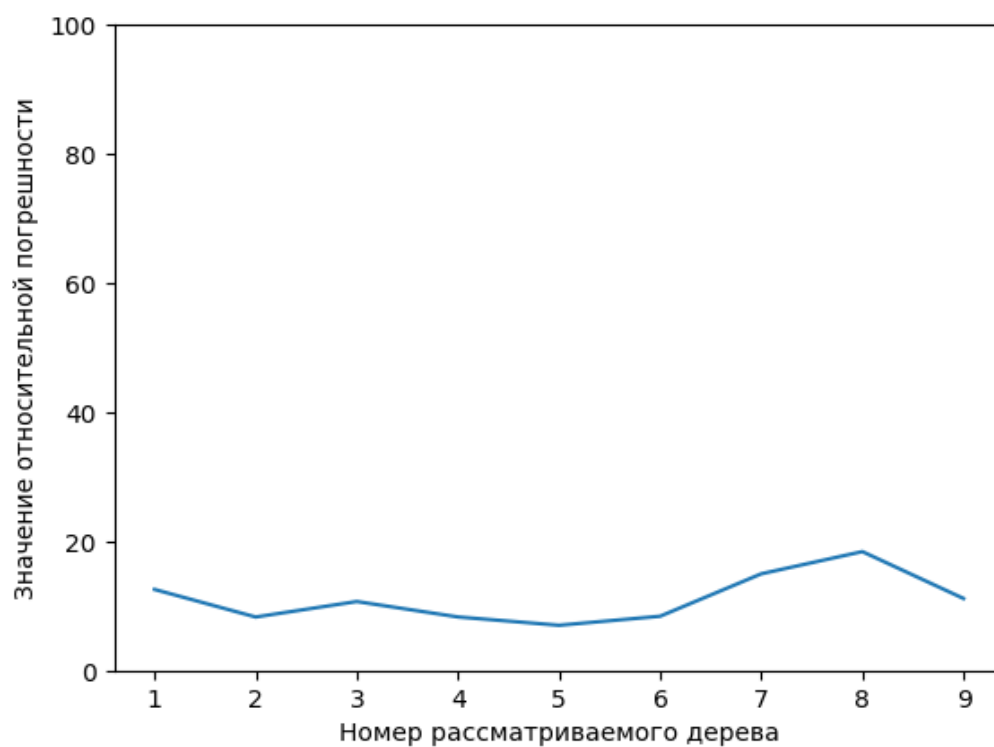


Рисунок 2 — Худшая относительная погрешность принадлежит диапазону значений $6.99 \leq \delta \leq 18.40$

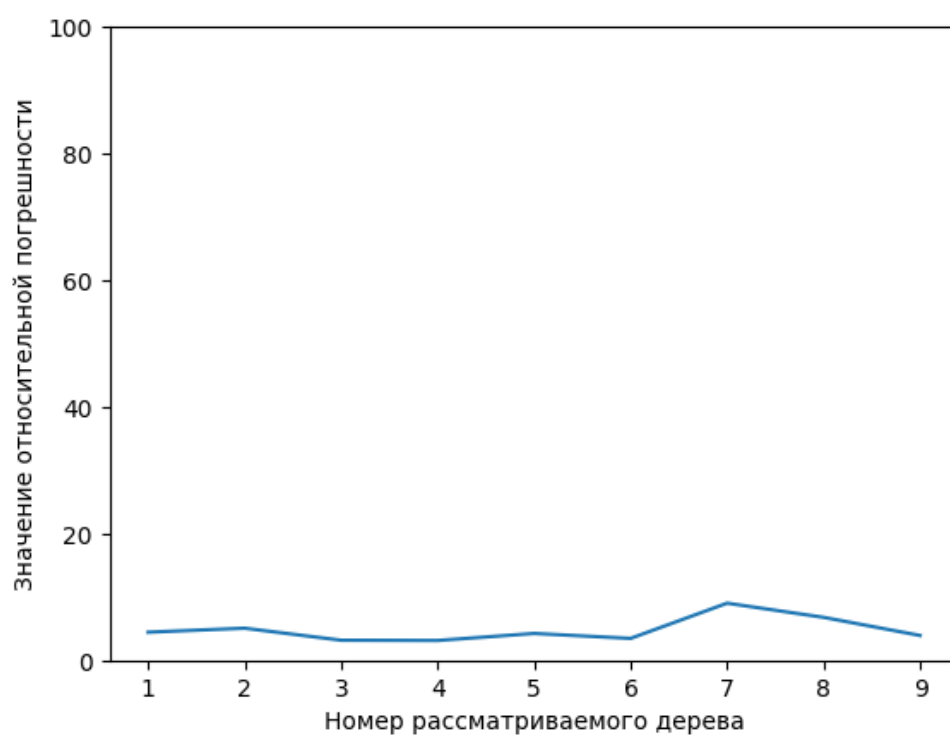


Рисунок 3 — Средняя относительная погрешность колеблется в диапазонах $3.16 \leq \delta \leq 9.01$

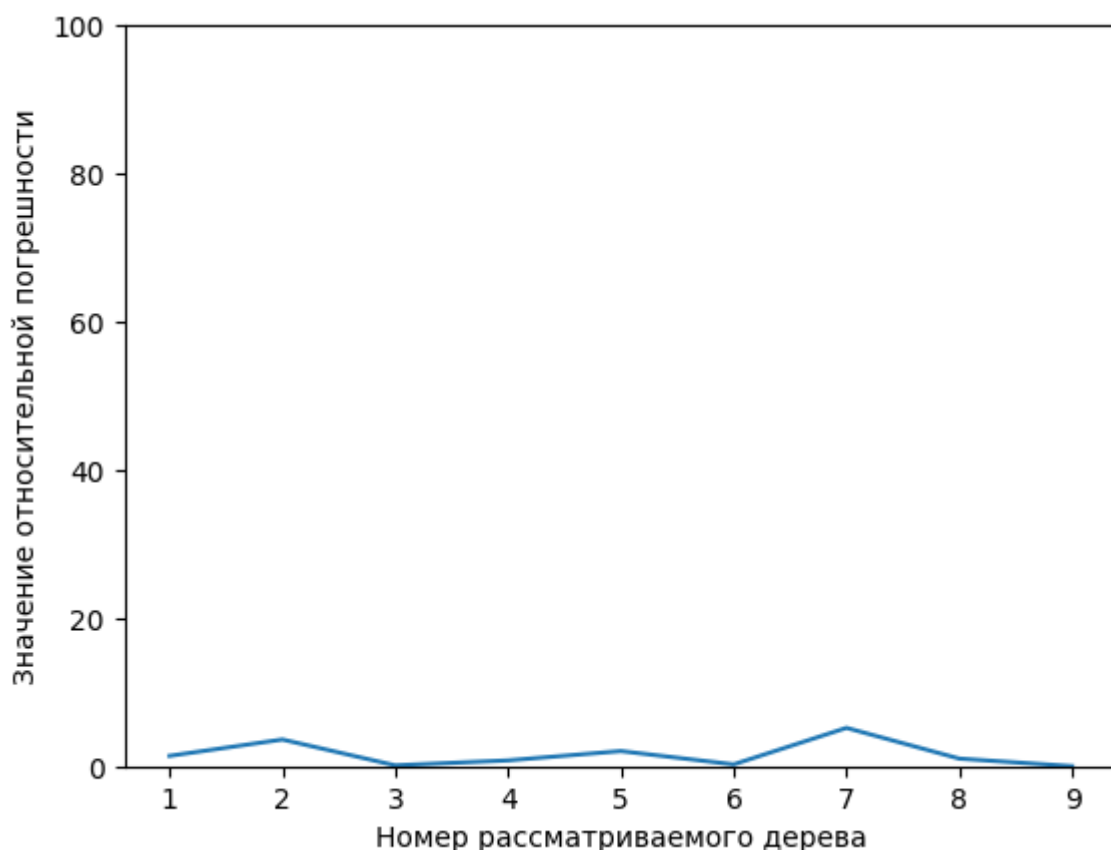


Рисунок 4 – Лучшая относительная погрешность принадлежит диапазону значений $0 \leq \delta \leq 5.17$

Полученные в ходе анализа результаты позволили выявить определенные частные случаи, в которых становится возможным точное определение изменения целевой функции при переходе к задаче CVRP содержащей один цикл.

Рассмотрим алгоритм потерь для частного случая, при добавления ребра между листьями.

Пусть z - корень поддерева смежный с листьями i и j . (i, j) - ребро, добавляемое между двумя листьями i и j , $c[i]$ - вес ребра (i, j) , Q - вместимость транспортного средства, $F(z)$ - функция вычисляющая минимальное количество транспортных средств, необходимых для обслуживания данного поддерева графа.

Исходя из анализа полученных данных, лучшая минимальная погрешность достигается, в случае, если сумма веса добавленного ребра и минимального из двух рёбер, которые соединяются данным ребром, отличается от веса максимального ребра не больше чем на добавленное ребро, данная погрешность равна 0.

Существуют два варианта:

1) $q[i] + q[j] < Q$:

Для первого случая сначала вычисляется функция $F(z)$. Затем мы смотрим на поддерево z' , где веса вершин $q[i]$ и $q[j]$ объединяются и рассматриваются как единая вершина с весом $q[i] + q[j]$. Затем вычисляется $F(z')$ и рассчитывается разность: $F(z') - F(z)$.

- Если $F(z') - F(z) > 0$ и $c[0, z] > \min(c[z, i], c[z, j])$, то снижение целевой функции равно:
$$2 * (c[z, i] + c[z, j]) - 2 * c[i, j] - 4 * \min(c[z, i], c[z, j]).$$
- Если $c[0, z] \leq \min(c[z, i], c[z, j])$, то это выражается как
$$2 * ((c[z, i] + c[z, j]) - (c[i, j] + \min(c[z, i], c[z, j]))) + 2 * c[0, z].$$
- Если $F(z') - F(z) = 0$, то целевая функция вычисляется как: $2 * ((c[z, i] + c[z, j]) - (c[i, j] + \min(c[z, i], c[z, j])))$.

2) $q[i] + q[j] \geq Q$:

- Если $\min(c[z, i], c[z, j]) + c[i, j] < \max(c[z, i], c[z, j])$, то снижение целевой функции равно:
$$2 * (c[z, i] + c[z, j]) - 2 * c[i, j] - 4 * \min(c[z, i], c[z, j]);$$
- Во всех остальных случаях целевая функция остается равной нулю.

В случае, когда сумма вместимостей вершин $q[i]$ и $q[j]$ меньше вместимости транспортного средства Q , проводится ряд операций. Первым шагом является расчет разности между функциями $F(z)$ и $F(z')$, где $F(z)$ вычисляет минимальное количество транспортных средств, необходимых для обслуживания поддерева, и $F(z')$ вычисляет то же для поддерева, в котором веса вершин $q[i]$ и $q[j]$ объединяются.

Этот расчет помогает определить, будет ли уменьшаться максимально возможное количество пар в поддереве, которое не превышает Q , если транспортное средство обслуживает клиентов i и j за одну поездку. Если количество пар уменьшается, встречаются два возможных сценария.

- 1) В первом сценарии расстояние от депо 0 до корня узла z превышает минимальное расстояние от корня узла до каждой из вершин i и j . С учетом свойств задачи CVRP о поиске минимального пути, маршрут будет спланирован так, чтобы минимизировать количество проходов по пути $(0, z)$. В результате количество транспортных средств, необходимых для обхода поддерева, не увеличится, и целевая функция уменьшится только на минимальное расстояние до одной из вершин i или j .

- 2) Во втором сценарии расстояние от депо до корня узла меньше или равно минимальному расстоянию от корня узла до каждой из вершин i и j . В этом случае маршрут будет спланирован так, чтобы минимизировать количество маршрутов пролегающих через ребро (z, i) или (z, j) . Следовательно, вершины i и j будут обслужены за один обход транспортного средства, и количество маршрутов увеличится на 1, так как $0 \leq F(z') - F(z) \leq 1$.

Если $F(z') - F(z) = 0$, то маршрут будет спланирован таким образом, чтобы за один обход обслужить вершины i и j , следовательно, маршрут укоротится на минимальное расстояние до одной из вершин i или j .

Предположим, что сумма вершин $q[i]$ и $q[j]$ превышает значение Q . В таком случае мы можем рассмотреть два потенциальных сценария:

- 1) Если сумма весов добавленного ребра и минимального веса из рёбер (z, i) и (z, j) окажется меньше максимального веса среди рёбер (z, i) и (z, j) , то это приведёт к уменьшению значения целевой функции. Специфически, снижение будет эквивалентно разности минимального расстояния до наиболее отдаленной из вершин i, j и веса ребра (i, j) .
- 2) В противоположном случае, если указанное условие не выполняется, значение целевой функции останется неизменным.

В ходе проведения исследования активно использовались численные эксперименты для проверки предложенного алгоритма. Набор данных, состоящий из множества различных маршрутов и соответствующих им ограничений по грузоподъемности, использовался для подробного анализа алгоритма. С целью подтверждения устойчивости и эффективности алгоритма, было проведено множество итераций численного моделирования. Данный процесс позволил качественно оценить производительность алгоритма, учитывая различные сценарии и условия задачи. Результаты численных экспериментов исчерпывающе подтверждают работоспособность предложенного алгоритма в контексте задачи CVRP.

Рассмотрим анализ второго сценария, при котором проводилась генерация n -арных деревьев с вариативностью связей, колеблющейся в диапазоне от 1 до 10. Это исследование было проведено с целью выявления корреляции между увеличением числа связей у каждого узла и относительной ошибкой. Исходя из полученных результатов, были идентифицированы сценарии, при которых целевая функция изменяется наиболее существенным образом, а также был разработан приблизительный алгоритм для оценки нижнего предела этой функции. Данный алгоритм демонстрирует наихудший из возможных сценариев изменения целевой функции, при этом не гарантируя достижения

определенной нижней границы. График, иллюстрирующий зависимость наибольшей относительной ошибки от количества связей на каждом узле, представлен на рисунке 5. Средняя относительная ошибка изменения целевой функции отображена на рисунке 6. Наибольшая ошибка была зафиксирована при $n = 2$ и составила 7.2%.

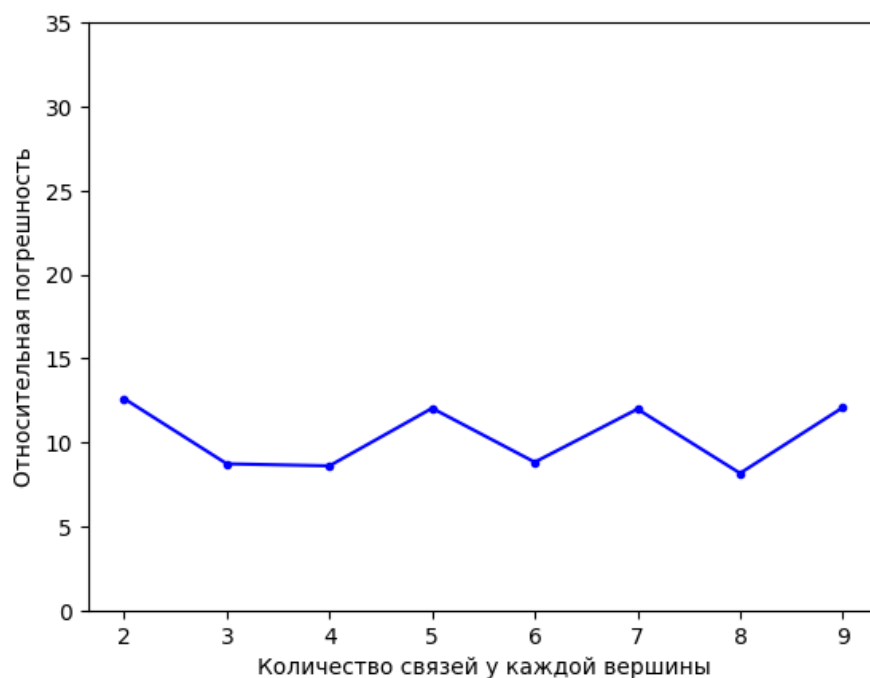


Рисунок 5 – Худшая относительная погрешность $8.16 \leq \delta \leq 12.6$

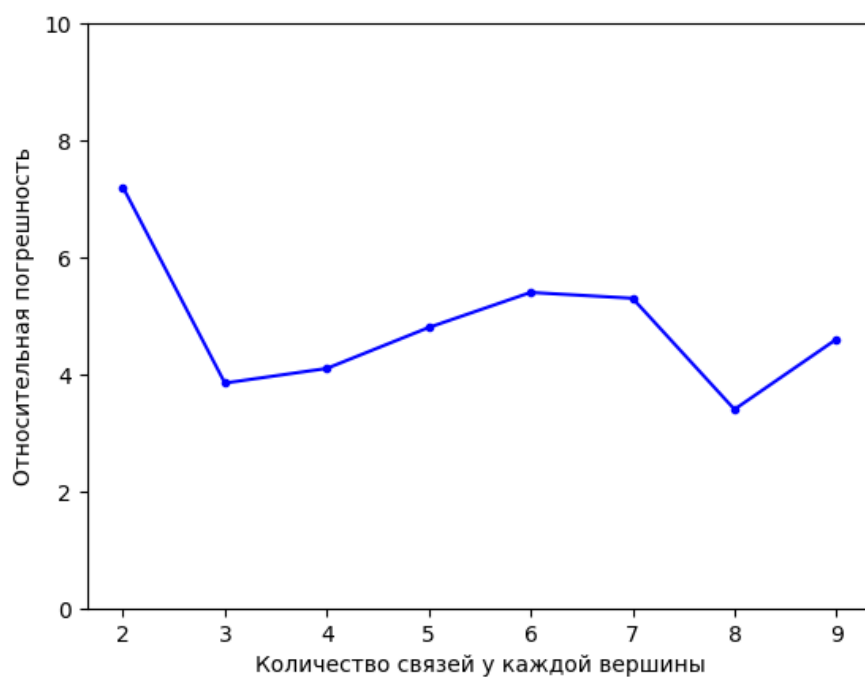


Рисунок 6 – Средняя относительная погрешность погрешность $3.4 \leq \delta \leq 7.2$

В ходе экспериментального исследования был идентифицирован наиболее неблагоприятный сценарий потенциального включения дополнительного ребра, при котором целевая функция испытывает максимальные изменения. Этот сценарий характеризуется ситуацией, когда поддерево с максимальной весовой характеристикой в графе становится смежным с депо посредством ребра с минимальным весом. Такое происходит после включения указанного ребра, либо когда ближайшее к депо ребро становится смежным с поддеревом, обладающим максимальной весовой характеристикой.

Приближенный алгоритм для оценки максимально возможного изменения функции.

Алгоритм можно описать следующим образом:

- 1) Инициализация процесса: Идентификация поддерева с наибольшим количеством вершин в пределах исследуемого графа, если данного поддерева не существует, то выбирается поддерево с наибольшим суммарным весом ребер.
- 2) На данной стадии алгоритма, вес ребра, соединяющего депо и найденного поддерева, устанавливается равным максимальному значению веса среди всех ребер, присутствующих в графе.
- 3) Рекурсивный поиск: Внутри обнаруженного поддерева проводится последовательный поиск. На каждом шаге итерации проверяется условие, что количество узлов в текущем максимальном поддереве не уступает числу узлов в остальной части поддерева, в рамках которой осуществляется поиск.
- 4) Вычисление отношения: После обнаружения целевого поддерева производится расчет отношения числа вершин в этом поддереве к общему количеству вершин в исходном графе.
- 5) Определение количества транспортных средств: Исходя из полученного отношения, рассчитывается пропорциональное количество транспортных средств, обслуживающих все маршруты.
- 6) Расчет приближенного изменения функции: Последний этап алгоритма предполагает расчет приближенного изменения функции в худшем случае. Для этого из кратчайшего расстояния до обнаруженного поддерева до момента добавления нового ребра вычитается вес добавленного ребра, умноженный на ранее определенное отношение. Полученное значение дополнительно умножается на два для получения окончательной оценки.

Аналитическое изучение полученных результатов и определенных специфических случаев обозначило главную тенденцию: при увеличении числа вершин и n в n -арных деревьях, относительная ошибка изменения целевой функции при добавлении и удалении ребер существенно сокращается. Для подтверждения выдвинутой гипотезы функция

попарных сравнений с применением критерия относительной погрешности была численно определена. Результаты этого исследования, включающие добавление и удаление одного ребра в дереве, представлены на рисунке 7 и рисунке 8 соответственно. Каждая отметка на графике представляет собой совокупность десяти примеров.

Также была оценена верхняя граница относительной погрешности, для n -арного графа с n вершинами $20 \leq n \leq 120$. Численные исследования показали, что при увеличении вершин в графе и пропорциональному увеличению степени каждой вершины, худшая относительная погрешность колеблется в пределах $7.9 \leq \delta \leq 10.5$

В ходе исследования количество вершин в графе колебалось в диапазоне от 20 до 150. Количество связей на каждой вершине оптимально увеличивалось.

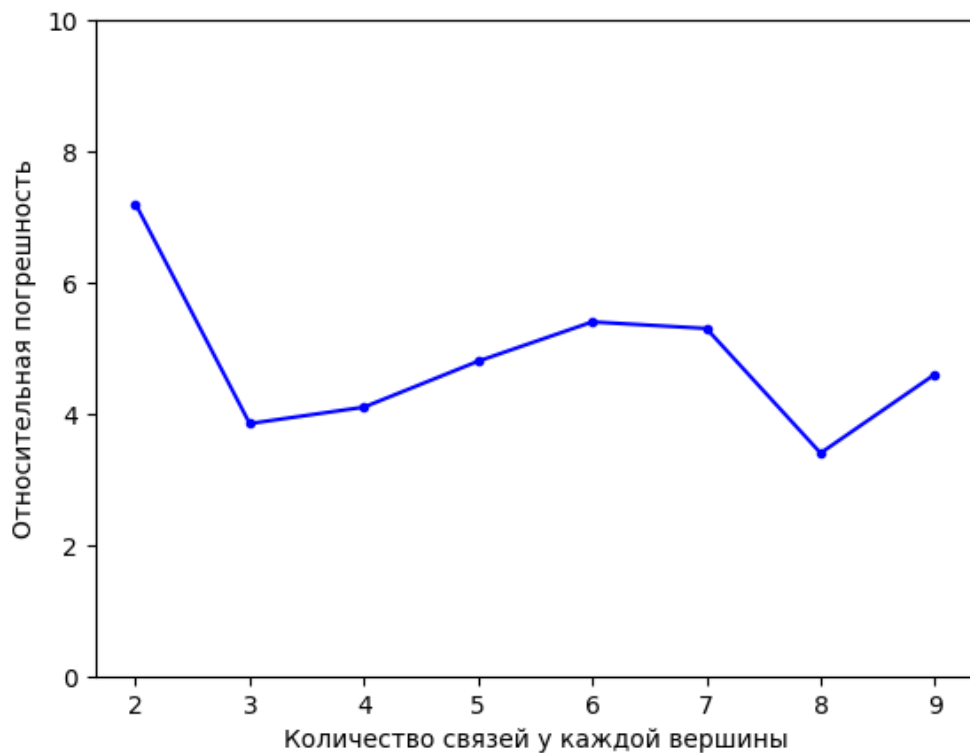


Рисунок 6 – Средняя относительная погрешность $3.4 \leq \delta \leq 7.2$

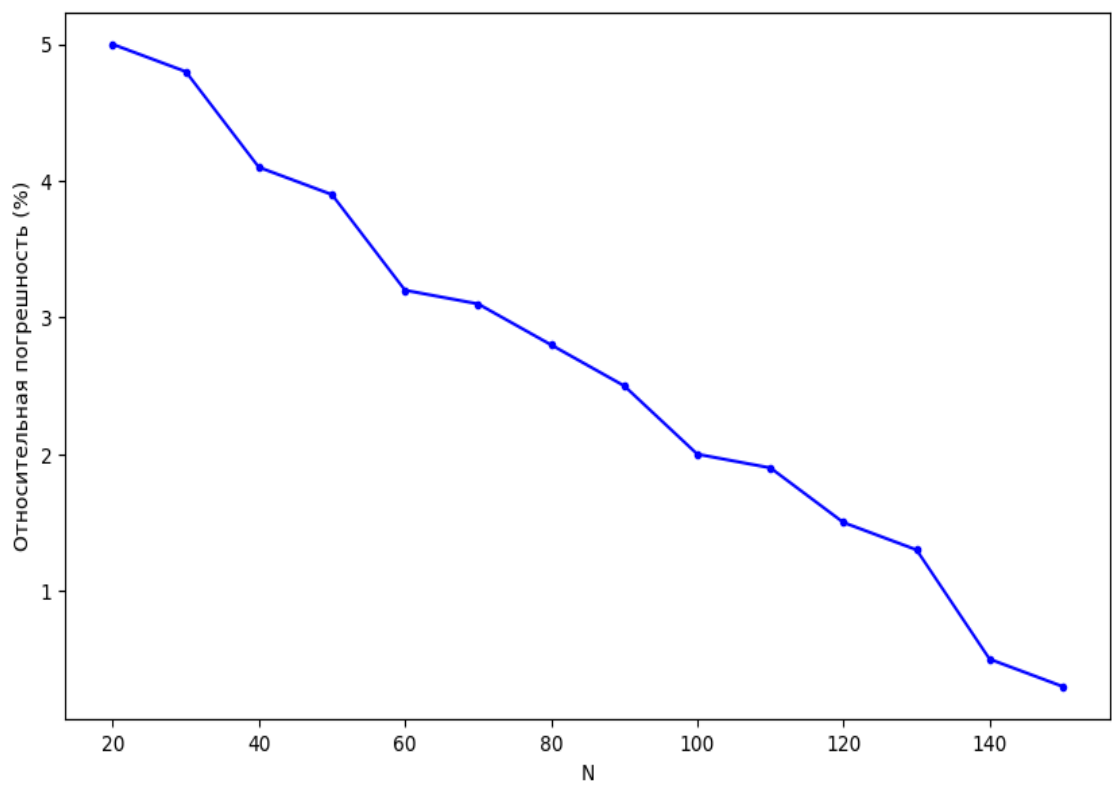


Рисунок 7 – Численная функция попарных сравнений с критерием относительной погрешности при добавлении случайного ребра с минимальным весом

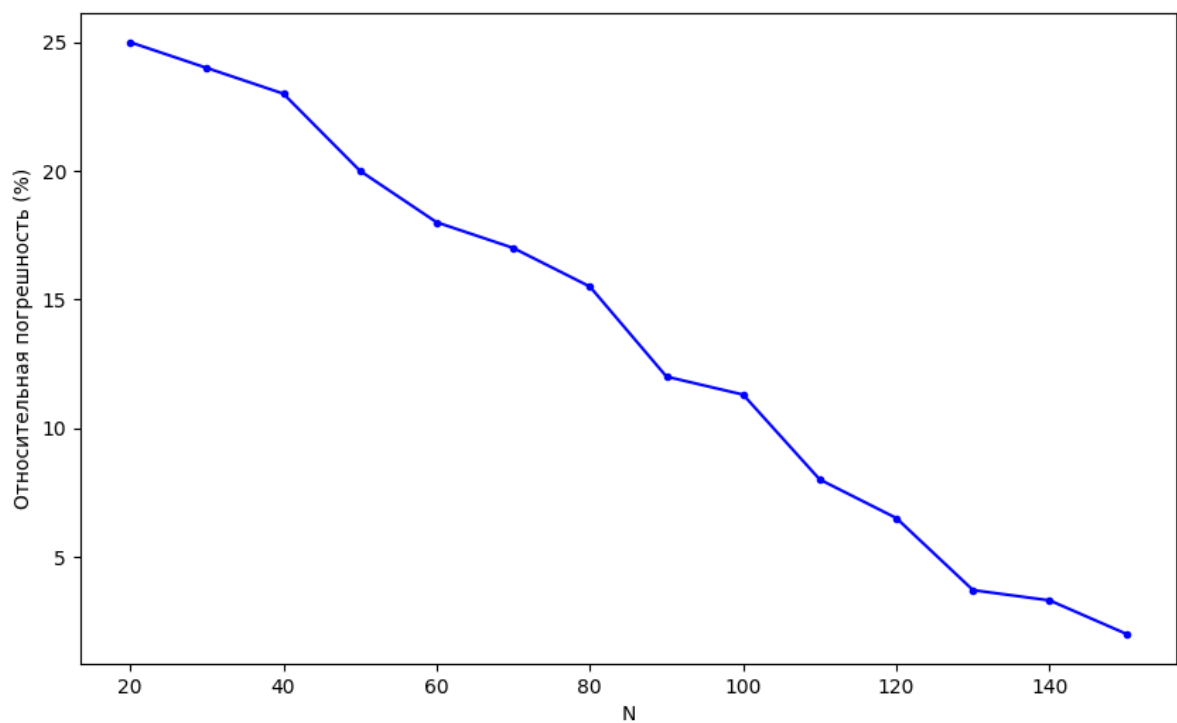


Рисунок 8 – Численная функция попарных сравнений с критерием относительной погрешности при удалении случайного ребра

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы бакалавра все поставленные цели и задачи были выполнены: нахождение частных случаев изменения целевой функции, численное определение функции попарных сравнений для задачи маршрутизации транспортных средств с ограниченной вместимостью.

Анализ полученных результатов показал, актуальность применения метода попарных сходств к решению задачи CVRP. Выводы, сделанные на основе проведенной работы, являются важным вкладом в понимание и решение задачи CVRP. Они также представляют ценный источник для дальнейшего развития методологии в этом направлении, а также для совершенствования теоретических и практических подходов к решению CVRP.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Toth P., Vigo D. Branch-and-bound algorithms for the capacitated VRP //The vehicle routing problem. – Society for Industrial and Applied Mathematics, 2002. – С. 29-51.
2. Kopman L. A new generic separation routine and its application in a branch and cut algorithm for the capacitated vehicle routing problem. – Cornell University, 1999.
3. Zhu W. et al. A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP //Computers & Operations Research. – 2012. – Т. 39. – №. 9. – С. 2178-2195.
4. Lin S. W. et al. Applying hybrid meta-heuristics for capacitated vehicle routing problem //Expert Systems with Applications. – 2009. – Т. 36. – №. 2. – С. 1505-1512.
5. Mazzeo S., Loiseau I. An ant colony algorithm for the capacitated vehicle routing //Electronic Notes in Discrete Mathematics. – 2004. – Т. 18. – С. 181-186.
6. Kaku I., Xiao Y., Xia G. The deterministic annealing algorithms for vehicle routing problems //International Journal of Smart Engineering System Design. – 2003. – Т. 5. – №. 4. – С. 327-339.
7. Vidal T. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood //Computers & Operations Research. – 2022. – Т. 140. – С. 105643.
8. Tao Y., Wang F. An effective tabu search approach with improved loading algorithms for the 3L-CVRP //Computers & Operations Research. – 2015. – Т. 55. – С. 127-140.
9. Hamaguchi S., Katoh N. A capacitated vehicle routing problem on a tree //Algorithms and Computation: 9th International Symposium, ISAAC'98 Taejon, Korea, December 14–16, 1998 Proceedings 9. – Springer Berlin Heidelberg, 1998. – С. 399-407.
10. Asano T., Katoh N., Kawashima K. A new approximation algorithm for the capacitated vehicle routing problem on a tree //Journal of Combinatorial Optimization. – 2001. – Т. 5. – №. 2. – С. 213-231.
11. Mathieu C., Zhou H. A PTAS for capacitated vehicle routing on trees //ACM Transactions on Algorithms. – 2023. – Т. 19. – №. 2. – С. 1-28.
12. Labbé M., Laporte G., Mercure H. Capacitated vehicle routing on trees //Operations Research. – 1991. – Т. 39. – №. 4. – С. 616-622.
13. Baldacci R., Maniezzo V. Exact methods based on node-routing formulations for undirected arc-routing problems //Networks. – 2006. – Т. 47. – №. 1. – С. 52-60.

ПРИЛОЖЕНИЕ А

Исходный код программы

QR-код репозитория с кодом представлен на рисунке А.1.



Рисунок А.1 — QR-код репозитория с кодом