

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ЮФУ»)

На правах рукописи



Хуссейн Фирас Айманович

**МЕТОДЫ РЕШЕНИЯ МНОГОАГЕНТНОЙ ЗАДАЧИ КОММИВОЯЖЕРА
НА ОСНОВЕ СОКРАЩЕНИЯ ПОИСКОВОГО ПРОСТРАНСТВА**

Специальность 2.3.1 – «Системный анализ,
управление и обработка информации, статистика»

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель: доктор технических наук, профессор
М.Ю. Медведев

Таганрог – 2025 г.

Содержание

Введение.....	4
Глава 1. Обзор и анализ методов и подходов к решению многоагентной задачи коммивояжёра	10
1.1. Подход одновременной оптимизации	16
1.2. Подход на основе предварительной кластеризации	26
1.3. Подход на основе предварительной маршрутизации	30
1.4. Выводы по главе 1	32
Глава 2. Разработка метода решения многоагентной задачи коммивояжера на основе предварительной маршрутизации.....	33
2.1. Этап 1: решение задачи коммивояжёра.	36
2.2. Этап 2: разделение супер-маршрута между агентами.....	40
2.3. Этап 3: улучшение маршрутов с помощью алгоритма локального поиска	44
2.4. Моделирование и результаты.....	46
2.4.1. Оценка влияния локальной оптимизации на эффективность предлагаемого метода.....	46
2.4.2. Сравнительное исследование предлагаемого метода с методами, реализующими другие подходы решения многоагентной задачи коммивояжёра.....	57
2.5. Выводы по главе 2.....	73
Глава 3. Разработка гибридного метода решения многоагентной задачи коммивояжёра.....	74
3.1. Предлагаемый метод «Кластеризация, Маршрутизация, Соединение, Разделение»	74
3.1.1. Определение количества кластеров.....	76

3.1.2. Определение узлов соединения кластеров	81
3.1.3. Определение внутрикластерных маршрутов.....	82
3.2. Оценка масштабируемости предлагаемого метода	84
3.3. Моделирование и результаты.....	90
3.4. Выводы по главе 3.....	103
Глава 4. Экспериментальные исследования разработанных методов.....	105
4.1. Моделирование.....	108
4.2. Результаты и сравнительный анализ	111
4.3. Статистическая значимость полученных результатов	125
4.4. Выводы по главе 4.....	130
Заключение	131
СПИСОК ЛИТЕРАТУРЫ.....	134
Приложения	144

Введение

Актуальность темы. Многоагентные системы представляют собой коллективы автономных агентов, способных координировать свои действия для решения общей задачи. В настоящее время эти системы находят широкое применение в разных областях, таких как промышленность, логистика, медицина и оборона.

В связи с расширением сферы применения многоагентных систем, всё большую значимость приобретают методы группового управления подвижными агентами – элементами этих систем, среди которых важнейшими являются задачи целераспределения, маршрутизации и балансировки нагрузки.

Решения подобных задач особенно критичны при наличии ограничений по времени, энергии и другим ресурсам. Одним из ключевых направлений здесь является оптимизация маршрутов агентов при выполнении пространственно-распределённых задач, что на практике сводится к задаче, известной как многоагентная задача коммивояжёра (МКВ)

Многоагентная задача коммивояжёра — это обобщение классической задачи коммивояжёра, в которой допускается присутствие более одного коммивояжёра (агента). Её целью является определение маршрута для каждого коммивояжёра таким образом, чтобы оптимизировать заданный функционал качества. При этом каждый город должен посетить ровно один коммивояжер. Таким образом, многие задачи целераспределения в мультиагентных системах сводятся к формулировке МКВ или её вариациям, что делает исследование соответствующих методов особенно актуальным.

Качество и эффективность решения МКВ во многом зависит от сложности задачи (количество агентов и городов, пространственное распределение, наличие ограничений) и от выбранного метода решения. Во-первых, метод должен оптимизировать заданную целевую функцию – например, общее пройденное агентами расстояние, время выполнения задачи. При этом требуется равномерная загрузка агентов, чтобы избежать ситуаций, когда некоторые агенты перегружены, а другие недогружены. Во-вторых, метод должен обладать приемлемой

вычислительной сложностью, что особенно влияет на степень его масштабируемости – его способность эффективно работать при увеличении числа задач и агентов.

Таким образом, разработка эффективных методов решения многоагентной задачи коммивояжёра является актуальной задачей для целого ряда приложений мультиагентных систем.

Степень разработанности темы. Многоагентная задача коммивояжёра получила широкое распространение при формализации проблем маршрутизации мобильных роботов, распределения задач между роботами, планирования логистических операций и др.

В научной литературе представлено большое количество методов и алгоритмов решения МКВ. Среди них – точные (например, метод ветвей и границ, методы динамического программирования), эвристические (жадные алгоритмы, методы кластеризации) и метаэвристические (генетические алгоритмы, рой частиц, муравьиные алгоритмы и др.). Особый интерес вызывают гибридные методы, сочетающие элементы кластеризации, построения маршрутов и локального поиска.

Однако, несмотря на значительное количество работ, проблема эффективного решения МКВ в условиях статически заданной среды, остаётся актуальной. Это связано с высокой вычислительной сложностью задачи, чувствительностью решений к параметрам задачи и гиперпараметрам алгоритмов решения, а также с недостаточной исследованностью стратегий, комбинирующих этапы маршрутизации, кластеризации и локальной оптимизации.

Различным аспектам проблемы целераспределения и МКВ посвящены работы отечественных (Германчук М.С., Бурховецкий В.В., Полупанова Е.Е., Рыбалко А.А., Козлова М.Г., Лукьяненко В.А., Макаров О.О., Матвиевская Т.Б., Тарков М.С., Дугаров Г.А. и др.) и зарубежных (Howard T.M., Khatib O., Khoufi I., Hadded M., Dorigo M., Kolmanovsky I., Latah M., Cheikhrouhou O., Koubaa A., Bennaceur H., и др.) ученых.

Объектом исследования является многоагентная система, представленная группой подвижных объектов.

Предметом исследования являются методы решения многоагентной задачи коммивояжера.

Целью диссертационной работы является повышение показателей качества решения многоагентной задачи коммивояжера, в частности минимизация суммарной длины маршрутов и максимальной длины индивидуального маршрута, а также сокращение времени расчёта.

Задачи, решаемые в диссертации:

1. Провести анализ методов и алгоритмов решения многоагентной задачи коммивояжера.
2. Определить целевые критерии, характерные для практических задач группового управления.
3. Разработать методы решения МКВ, сочетающие элементы кластеризации и маршрутизации, обеспечивающие компромисс между качеством решений и вычислительными затратами.
4. Провести экспериментальные исследования разработанных методов на тестовых задачах, оценить влияние числа агентов и заданий на качество решений, а также сравнить с существующими подходами по основным показателям качества.

Методы исследования. Для решения поставленных задач использованы методы оптимизации, математического и численного моделирования и прикладного программирования. Основные расчеты, моделирование и разработка программ выполнены с использованием программных продуктов: PyCharm, Anaconda Notebook. Язык программирования – Python.

Научная новизна заключается в разработке гибридного метода решения МКВ, отличающегося сочетанием подходов «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация», что позволяет:

- повысить равномерность распределения нагрузки между агентами системы;

- сократить время вычислений за счет уменьшения размера поискового пространства;
- обеспечить гибкий компромисс между качеством решения и временем вычислений за счёт регулирования количества.

Теоретическая и практическая значимость работы. Теоретическая значимость полученных результатов заключается в развитии методов решения МКВ, направленных на повышение эффективности в условиях статических сред. Алгоритмы, соответствующие предлагаемым методам, реализованы на языке Python в виде комплекса программ, обеспечивающего распределение задач и маршрутизации в группе подвижных объектов. Полученные результаты могут быть использованы для разработки систем группового управления автономными подвижными объектами и способствуют повышению их эффективности за счёт сокращения времени достижения цели.

Достоверность полученных результатов обеспечивается:

- строгими математическими обоснованиями, основанными на принципах и методах системного анализа;
- соответствием теоретических выводов результатам компьютерного моделирования, проведённого в специализированных программно-аппаратных средах;
- согласованием результатов работы с опубликованными результатами научных исследований и экспериментов других авторов.

Реализация и внедрение результатов работы. Теоретические и практические результаты, полученные в рамках работы, использованы при выполнении гранта РФФИ № 24-29-00492 «Разработка методов оптимального целераспределения в группе подвижных робототехнических комплексов», проведенного на базе АО «НКБ Робототехники и систем управления». Результаты, полученные в ходе исследования, использовались при выполнении работ по гранту УМНИК Фонда содействия инновациям. Результаты работы внедрены при разработке и исследовании сценариев применения групп робототехнических платформ при решении

типовых задач в рамках проекта «Экспериментально-теоретические исследования по отработке технологии автономного управления движением группы наземных робототехнических платформ», головной исполнитель АО «НПО Андроидная техника» (г. Магнитогорск).

Наиболее существенные научные результаты, полученные автором и обладающие научной новизной:

1. Метод решения многоагентной задачи коммивояжёра, основанный на подходе «сначала маршрутизация, затем кластеризация», отличающийся использованием алгоритма локального поиска для повышения качества решения.

2. Гибридный метод решения многоагентной задачи коммивояжёра, сочетающий подходы «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация», объединяющий их сильные стороны и устраняющий их недостатки, что, в свою очередь, помогает сократить время вычислений и улучшить равномерность распределения нагрузки между агентами.

Основные положения, выносимые на защиту:

1. Критерий максимального по длине пути среди агентов является более важным по сравнению с критерием суммарной длины маршрутов в многоагентной задаче коммивояжёра.

2. Применение алгоритма локального поиска (2-opt) в сочетании с подходом «сначала маршрутизация, затем кластеризация» повышает эффективность решения многоагентной задачи коммивояжёра по критерию минимизации общей (суммарной) длины маршрутов.

3. Гибридный метод решения многоагентной задачи коммивояжёра, основанный на объединении подходов «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация» позволяет достичь компромисса между временем расчета решения и качеством получаемого решения.

4. Предлагаемый гибридный метод позволяет управлять компромиссом между временем вычислений и качеством решения за счёт изменения количества кластеров на первом этапе.

Апробация результатов работы. Теоретические положения и практические результаты работы докладывались на XIX Всероссийской научно-практической конференции «Перспективные системы и задачи управления» 2024 (п. Домбай, Карачаево-Черкесская республика), Всероссийской научно-технической конференции с международным участием имени профессора О.Н. Пьявченко “Компьютерные и информационные технологии в науке, инженерии и управлении” «КомТех-2024» (г. Таганрог), Международной конференции 10th International Conference on Control, Decision and Information Technologies CoDIT 2024 (Валлетта, Мальта), XX Всероссийской научно-практической конференции «Перспективные системы и задачи управления» 2025 (п. Домбай, Карачаево-Черкесская республика), Всероссийской научно-технической конференции с международным участием имени профессора О.Н. Пьявченко “Компьютерные и информационные технологии в науке, инженерии и управлении” «КомТех-2025» (г. Казань), Международной конференции 11th International Conference on Control, Decision and Information Technologies CoDIT 2025 (Сплит, Хорватия).

Личный вклад автора. Все научные результаты диссертационной работы получены автором лично.

Публикации. Основные результаты исследований по теме диссертации изложены в 9 работах, в том числе: в 4 статьях в ведущих научных изданиях, рекомендованных ВАК для публикации результатов работ по диссертациям на соискание ученой степени кандидата технических наук; в 2 статьях в иностранных научных изданиях, включенных в систему цитирования Scopus; в 2 докладах на всероссийских и международных конференциях; а также получено одно свидетельство о государственной регистрации программы для ЭВМ.

Структура и объем диссертации. Работа состоит из введения, четырех глав, заключения, списка литературы из 77 наименований, содержания и двух приложений. Основная часть работы изложена на 147 страницах и включает в себя 58 рисунков и 16 таблиц.

Глава 1. Обзор и анализ методов и подходов к решению многоагентной задачи коммивояжёра

Проблема целераспределения заключается в нахождении бесконфликтного сопоставления заданного множества задач мощностью n и множества агентов мощностью m таким образом, чтобы оптимизировать некоторый глобальный функционал качества. Каждому агенту может быть назначено не более L_i задач, и считается, что целераспределение завершено после назначения $N_{\min} = \min\{n, mL_t\}$ задач [1]. Если каждая задача назначается только одному агенту, то распределение считается бесконфликтным. Предполагается, что глобальная целевая функция представляет собой сумму локальных целевых функций, каждая из которых зависит от задач, назначенных конкретному агенту. Описанная выше задача может быть сформулирована как целочисленная (возможно, нелинейная) задача оптимизации с бинарными переменными решения x_{ij} , указывающими, назначена ли задача j агенту i :

$$\max \sum_{i=1}^m \left(\sum_{j=1}^n c_{ij}(X_i, P_i) \cdot x_{ij} \right), \quad (1.1)$$

при условии:

$$\sum_{j=1}^n x_{ij} \leq L_i, \forall i \in I,$$

$$\sum_{i=1}^m x_{ij} \leq 1, \forall j \in J,$$

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = N_{\min},$$

$$x_{ij} \in \{0,1\}, \forall (i, j) \in I \times J,$$

где: $x_{ij} = 1$ если агенту i назначена задача j , $X_i \in \{0,1\}^m$ – вектор, в котором j -ый элемент является x_{ij} , I – набор индексов агентов $I \subseteq \{1, \dots, m\}$, J – набор индексов задач $J \subseteq \{1, \dots, n\}$, P_i – вектор, представляющий упорядоченную последовательность задач для агента i .

Сумма в скобках в формуле (1.1) представляет собой локальное вознаграждение для агента i . Предполагается, что функция оценки удовлетворяет $c_{ij}(X_i, P_i) \geq 0$.

Когда каждому агенту присваивается из набора пространственно-распределённых задач несколько для выполнения, возникает не только задача оптимального распределения задач между агентами, но и задача поиска оптимальной последовательности их выполнения каждым агентом. Таким образом проблема множественного распределения задач сводится к многоагентной задаче коммивояжёра (МКВ) (Multiple Travelling Salesman Problem, MTSP) [2, 3].

Одной из самых известных классических комбинаторных задач является задача коммивояжера (КВ) (Travelling salesman problem, TSP) [4], целью которой является минимизация значения целевой функции — обычно расстояния — при определении последовательности посещения городов (задач), каждый из которых посещается коммивояжёром (агентом) ровно один раз, который затем возвращается в начальный город. Задача КВ может быть определена как поиск оптимального гамильтонова цикла в графе $G = (V, E)$ с множеством вершин V , представляющих города, соединённых ребрами E , представляющими дороги или пути, ведущие из одного города в другой, с соответствующим значением стоимости.

Рисунок 1.1 представляет граф из пяти вершин $V = \{v_1, v_2, v_3, v_4, v_5\}$, каждая из которых соответствует отдельному городу, соединённому с другими вершинами рёбрами. Пути между вершинами имеют соответствующую стоимость, обозначенную числом рядом с ребром. Самым простым способом решения за-

дачи КВ является перебор всех возможных маршрутов с использованием исчерпывающего метода. Однако по мере увеличения количества рёбер этот подход становится чрезмерно трудоёмким и вычислительно затратным [5].

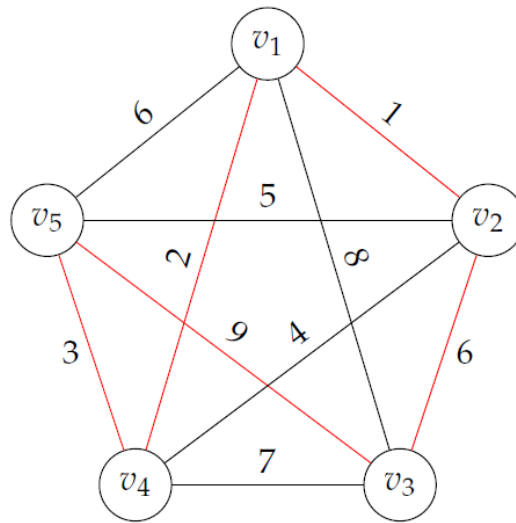


Рисунок 1.1. Пример графа, представляющего задачу КВ

Ввиду высокой временной сложности задачи КВ, при увеличении числа городов для её решения применяются эвристические и метаэвристические методы. Эвристика не гарантирует нахождение оптимального решения, однако позволяет получить достаточно качественный результат за приемлемое время, что особенно важно для практических приложений. С другой стороны, метаэвристика представляет собой общую модель, служащую руководством для построения эвристики. Многие из этих моделей основаны на природных явлениях [6].

В отличие от задачи коммивояжера, которая в последние годы получила широкое распространение и активно исследуется, многоагентная задача коммивояжёра изучена значительно менее подробно.

В зависимости от показателей качества решения различают следующие разновидности МКВ:

- MinSum МКВ: цель этого варианта — минимизировать общие затраты (например, расстояние или время) всех агентов [7]. Формально вариант MinSum моделируется следующим образом:

$$\min_{Tour_i \in TOURS} \left(\sum_{i=1}^m C(Tour_i) \right), \quad (1.2)$$

при условии: $Tour_i \cap Tour_j = \emptyset, \forall i \neq j, i \geq 1, j \leq m,$

где i, j – номер коммивояжера, $Tour_i$ – маршрут коммивояжера i , m – количество коммивояжеров, $TOURS$ – множество возможных решений данной МКВ, $C(.)$ – функция стоимости (длина маршрута).

• **MinMax МКВ:** цель в этом варианте — минимизировать стоимость самого затратного маршрута (например, с точки зрения расстояния или времени) среди всех агентов [8]. Этот случай широко используется в приложениях, направленных на сокращение времени выполнения миссии и балансировку нагрузки между агентами. Формальная модель этого варианта:

$$\min_{Tour_i \in TOURS} \left(\max_{j \in \{1, \dots, m\}} C(Tour_j) \right), \quad (1.3)$$

при условии: $Tour_i \cap Tour_j = \emptyset, \forall i \neq j, i \geq 1, j \leq m.$

Для оценки качества решения МКВ в данной работе используются оба критерия — **MinSum** и **MinMax**, поскольку применение каждого из них по отдельности может привести к нежелательным результатам [9]. В частности,

- использование только **MinSum** может привести к ситуации, где несколько агентов посещают лишь один город, в то время как остальные города присваиваются одному агенту. Это приводит к дисбалансу нагрузки между агентами.

- использование только **MinMax**, напротив, может привести к чрезмерному увеличению общей длины маршрутов.

Таким образом, совместное использование критериев **MinSum** и **MinMax** позволяет сбалансировать решения по двум важнейшим аспектам: равномерность распределения нагрузки и общая стоимость маршрутов.

Лемма. Суммарная длина маршрутов всех агентов при решении многоагентной задачи коммивояжера для заданного набора городов всегда превышает длину кратчайшего маршрута решения задачи одиночного коммивояжера (КВ) для того же набора, при условии, что все коммивояжеры начинают и заканчивают маршрут в одном и том же городе (депо).

Эта лемма показывает, что при решении МКВ нельзя добиться более короткого маршрута, чем в задаче КВ. Это логически обосновывает, почему не стоит использовать MinSum по отдельности и почему вводится компромиссный критерий MinMax.

Доказательство. Решение задачи МКВ предполагает разбиение гамильтонова цикла, определяющего оптимальное решение задачи КВ, на несколько частей, каждая из которых начинается и заканчивается в депо. Это достигается удалением некоторых рёбер из решения задачи КВ и добавлением двух новых рёбер, соединяющих точки разрыва с депо.

Рассмотрим одну такую замену, на рисунке 1.2 показан граф, представляющий решения задачи КВ, удаляется ребро длины y , соединяющее два соседних города Γ_1 и Γ_5 в маршруте КВ, и добавляются два рёбра длиной x_1 и x_2 , соединяющие каждый из этих городов с городом депо. Согласно неравенству треугольника, для любого треугольника с рёбрами x_1 , x_2 и y , выполняется нестрогое неравенство: $x_1 + x_2 \geq y$. Стоит отметить, что $x_1 + x_2 = y$ в том и только в том случае, если города задачи попадают на одну линию.

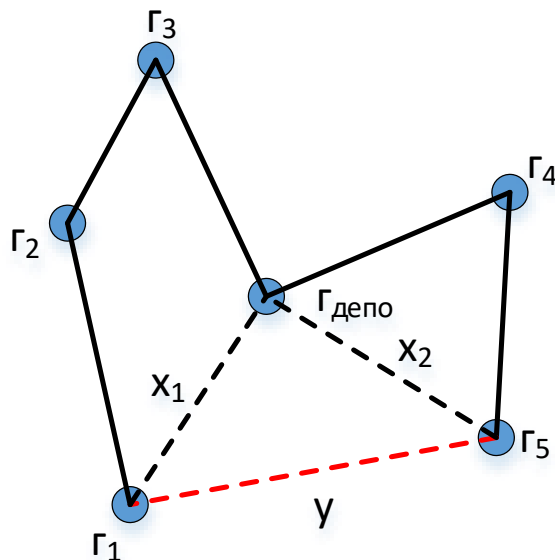


Рисунок 1.2. Пример решения задач КВ и МКВ

Следовательно, каждая операция замены одного ребра маршрута КВ на два ребра, ведущих в депо, может увеличить суммарную длину маршрутов. Таких замен выполняется не менее чем $m - 1$, где m — количество агентов МКВ. Таким образом, использование критерия MinSum в одиночку теряет практический смысл в контексте многоагентной постановки, поскольку минимальное значение целевой функции достигается тогда, когда все задачи выполняет один агент, что окончательно доказывает лемму. ■

Задача МКВ включает в себя две различные, однако взаимосвязанные подзадачи:

- распределение задач между агентами (назначение каждого города конкретному коммивояжёру);
- определение порядка выполнения задач в рамках маршрута каждого агента (определения порядка посещения городов для каждого коммивояжера).

В литературе выделяют три подхода к решению МКВ, основанные на различных последовательностях решения её двух составляющих подзадач:

1. Одновременная оптимизация — предполагает одновременное распределение задач между агентами и определение порядка их выполнения для каждого агента. Такой подход может обеспечить наилучшее качество решения, но является вычислительно наиболее трудоёмким.

2. Подход на основе предварительной кластеризации, также называемый «Сначала кластеризация, затем маршрутизация» (СКЗМ) (Cluster-First, Route-Second, CFRS) — сначала выполняется распределение всех задач между агентами (кластеризация), затем вопрос о порядке выполнения задач рассматривается независимо для каждого агента. В этом случае МКВ сводится к нескольким независимым задачам КВ.

3. Подход на основе предварительной маршрутизации, также называемый «Сначала маршрутизация, затем кластеризация» (СМЗК) (Route-First, Cluster-Second, RFCS) — сначала определяется порядок выполнения задач путём построения единого маршрута, охватывающего все задачи, аналогично задаче

КВ, а затем маршрут разбивается на части, которые распределяются между агентами. Это способствует более равномерной загрузке агентов. В этом случае МКВ сводится к задаче КВ.

1.1. Подход одновременной оптимизации

В этом подходе обе составляющие подзадачи МКВ — распределение задач между агентами и упорядочивание задач в маршруте каждого агента — алгоритмы или метаэвристики одновременно определяют, какой агент будет выполнять какую задачу, и каков порядок ее выполнения [10].

Подход одновременной оптимизации включает в себя множество методов, которые, в свою очередь, делятся на несколько классов: детерминированные методы, метаэвристические алгоритмы, рыночные механизмы распределения и другие. Классификация этих методов представлена на рисунке 1.3.

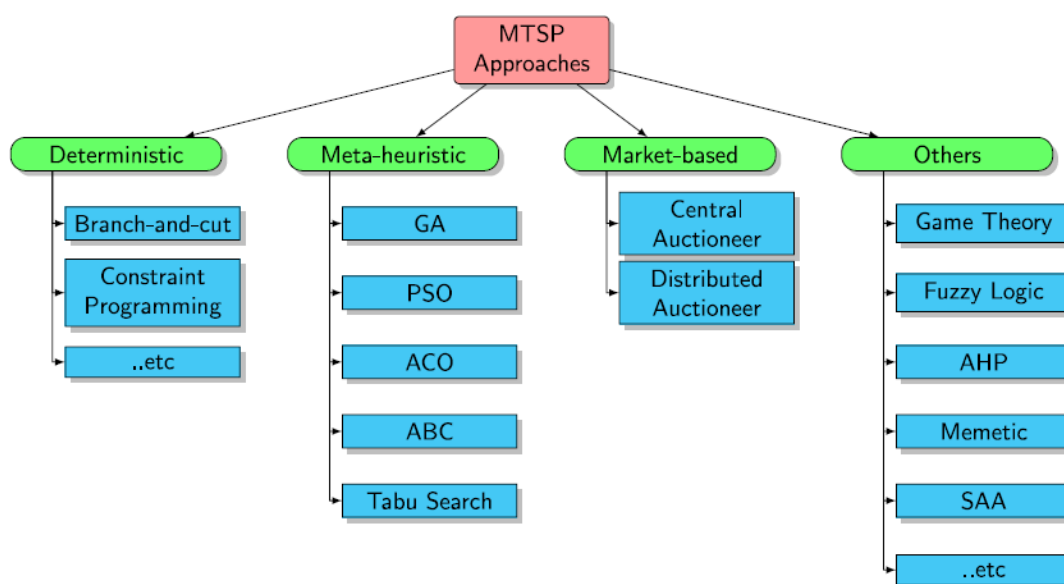


Рисунок 1.3. Классификация методов решения МКВ, определяющих подхода одновременной оптимизации [10]

Детерминированные методы

Предполагают применение точных алгоритмов, позволяющих найти оптимальное решение МКВ. Однако такие методы являются трудоёмкими и, как правило, применимы только к небольшим экземплярам МКВ. В связи с этим детерминированные подходы используются преимущественно в ограниченном числе работ, где рассматриваются задачи с малым числом агентов и городов.

Авторы в работе [11] представили формулировку МКВ в виде задачи целочисленного линейного программирования (ILP), а затем предложили настраиваемый алгоритм ветвления и отсечения. Полученное с его помощью субоптимальное решение было найдено за 300 секунд на экземпляре, содержащем 100 задач и 5 агентов.

В другой работе [12] для решения задачи МКВ применялось программирование с ограничениями (Constraint Programming, CP), с использованием глобальных ограничений, интервальных переменных и алгоритмов фильтрации доменов. Несмотря на высокую точность, подход оказался крайне ресурсоёмким: время расчёта составило более двух часов для задачи с 51 городом и 3 коммивояжёрами.

Метаэвристические методы

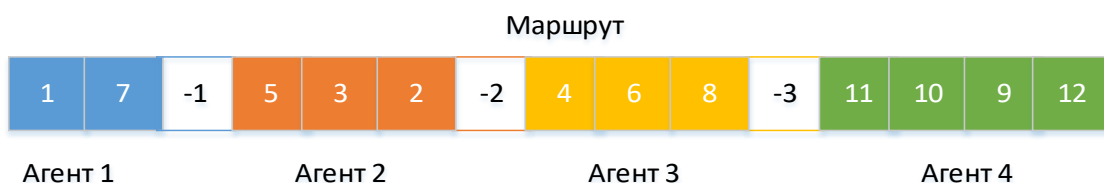
Наиболее часто используемые метаэвристические методы решения МКВ: генетический алгоритм (Genetic Algorithm, GA), муравьиный алгоритм (Ant Colony Optimization, ACO), оптимизация роя частиц (Particle Swarm Optimization, PSO) и методы искусственной пчелиной колонии. (Artificial Bee Colony, ABC).

Методы на основе генетических алгоритмов. Принцип этих методов заключается в применении механизмов естественного отбора и генетических операций, таких как кроссовер и мутация, для улучшения решений с каждым новым поколением. Процесс начинается с формирования начальной популяции случайных решений. Затем качество каждого решения оценивается с помощью функции пригодности. Далее выбираются пары наилучших решений, над которыми выполняются генетические операции, создавая новые решения (потомки) следующего поколения. Если потомки демонстрируют лучшие показатели функции пригодности, они заменяют родителей. Этот процесс отбора повторяется до достижения заданного числа поколений или пока качество решения не перестанет улучшаться.

В задаче МКВ решение кодируется в виде хромосомы. Основные способы представления хромосом, используемые в литературе, приведены на рисунке 1.4.

Авторы работ [13, 14] предложили двухчастное кодирование хромосом (рисунок 1.4. г) и разработали новый оператор кроссовера, который был сравнён с существующими операторами. Целями оптимизации были минимизация общей длины маршрутов и минимизация максимальной длины маршрута. Экспериментальный анализ на тестовых экземплярах различного размера из эталонной библиотеки TSPLIB [15] продемонстрировал эффективность предложенных подходов. Результаты показали, что разработанные операторы кроссовера обеспечивают более высокое качество решений по сравнению с существующими аналогами. Однако авторы не предоставили информацию о времени расчета, что ставит под сомнение эффективность предложенного решения.

В последние годы внимание ряда исследований было сосредоточено на партеногенетическом алгоритме (Partheno-Genetic Algorithm, PGA), который, в отличие от традиционных генетических алгоритмов, не использует операторы кроссовера [16]. В работе [17] были предложены два варианта партеногенетического алгоритма. Первый — PGA с элитарным отбором, в котором реализованы четыре новых типа операций мутации. Вторым вариантом — улучшенный алгоритм IPGA, в котором используется более универсальный оператор мутации. Авторы применили метод кодирования последовательности для представления популяции: хромосома состояла из двух частей (рисунок 1.4. в), где первая задаёт маршрут робота, а вторая — количество городов, принадлежащих каждому роботу. Эффективность предложенных решений была сравнена с методом оптимизации роя частиц (PSO) на тестовых задачах из библиотеки TSPLIB. Результаты моделирования показали, что IPGA демонстрирует наилучшую производительность среди рассмотренных алгоритмов, но за счет больших временных затрат. Методу требуется в среднем 21 секунда для решения MKB с 35 городами и 5 агентами.



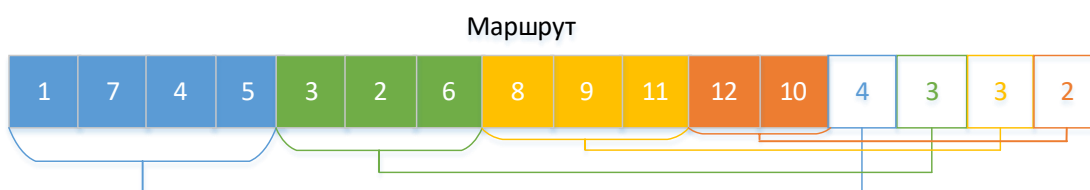
а) кодирование одной хромосомы – маршруты агентов представлены в одной хромосоме и разделены отрицательными цифрами



б) кодирование 2-х хромосом – первая содержит порядок посещения городов задачи, вторая содержит принадлежность каждого города определённому агенту



в) двухчастное кодирование хромосомы – первая часть содержит порядок посещения городов задачи, вторая часть содержит индексы конечных городов агентов



г) двухчастное кодирование хромосомы – первая часть содержит порядок посещения городов задачи, вторая часть содержит количество городов для каждого агента

Рисунок 1.4. Примеры представления хромосом в решениях МКВ с использованием генетического алгоритма

Авторы в [18] использовали недоминируемый генетический алгоритм (NSGA-II) для решения многоцелевой МКВ. Решение было направлено на оптимизацию двух критериев: минимизацию общего пройденного расстояния и балансировку времени в пути между коммивояжёрами. Однако в работе не приведены сравнительный анализ с другими методами и оценка баланса времени в пути.

Методы на основе алгоритма оптимизации роя частиц. Алгоритм оптимизации роя частиц является одной из самых известных метаэвристик и имеет много общего с генетическими алгоритмами. Процесс поиска решения начинается с инициализации популяции случайных решений, после чего алгоритм итеративно обновляет их, стремясь к достижению оптимума. В отличие от генетического алгоритма, PSO не использует операции кроссовера или мутации для генерации новых решений. Вместо этого возможные решения, называемые частицами и характеризующиеся своей скоростью, перемещаются по пространству решений, следуя за лучшими найденными решениями.

Алгоритм оптимизации роя частиц был применен в работе [19]. В этом исследовании авторы рассмотрели проблему распределения задач между несколькими роботами и сформулировали ее как МКВ. Решение направлено на минимизацию общей длины маршрутов и максимальной длины маршрута среди роботов. Авторы расширили стандартный PSO и предложили две стратегии: стратегия уточнения фронта Парето, которая удаляет худшие решения, и стратегия выбора лидера на основе вероятности. Авторы сравнили предложенный подход с известными многоцелевыми подходами, такими как улучшенная многокритериальная оптимизация OMOPSO [20], усовершенствованный эволюционный алгоритм Парето SPEA2[21], NSGA-II [22] и SMPSO [23], и продемонстрировали его превосходство. Однако, авторы не привели оценку временных затрат предложенного метода.

Методы на основе муравьиных алгоритмов. Оптимизация колонии муравьев – это метаэвристика на основе популяции для решения задач комбинаторной оптимизации. АСО имитирует способность настоящих колоний муравьев

эффективно организовывать поведение колонии при поиске пищи, используя химические феромонные следы для общения между муравьями. В популяции каждый индивидум постепенно и стохастическим образом конструирует решение заданной задачи оптимизации. Далее рассматривается вклад, основанный на алгоритме АСО для решения МКВ [24].

Авторы исследования [25] смоделировали задачу целераспределения для нескольких автономных необитаемых подводных аппаратов (АНПА) как многокритериальную задачу МКВ, в которой минимизируется общее пройденное расстояние и суммарный угол поворотов, что способствует снижению энергопотребления аппаратов [26]. Предложенный метод включает в себя два этапа, сначала определяется количество целей, назначаемых каждому аппарату, затем решается задача МКВ с помощью предложенного метода Multiple Ant Colonies System (MACS). Сравнительный анализ результатов показал, что данная система превосходит классический алгоритм АСО.

В работе [27] авторы предложили алгоритм МОАТ-АСО (Mission-Oriented Ant Team АСО), направленный на минимизацию общего пройденного расстояния и обеспечение балансировки нагрузки задачи МКВ. В предложенном алгоритме был реализован феромон миссии, имитирующий биологический инстинкт, что позволило снизить пересечения между маршрутами муравьёв, следовательно, минимизировать одну из целевых функций – общее расстояние. Для оптимизации второй целевой функции – баланса нагрузки между агентами – предложено правило стрельбы муравьёв, которое позволяет только одному муравью с наименьшим текущим пройденным расстоянием выбирать следующий город. Данный механизм способен обеспечить равномерное распределение нагрузки. Однако результаты не включают оценку решения по критерию общего расстояния.

Авторы в работе [28] предложили алгоритм на основе АСО для решения многоцелевой версии МКВ. Алгоритм АСО был интегрирован с процедурой локального поиска с чередующимися окрестностями (Sequential Variable Neighborhood Descent) для локального улучшения решений. Целью алгоритма

является одновременная минимизация общего пройденного расстояния и максимальной длины маршрута. Метод сравнивался с алгоритмом NSGA-II, многокритериальным эволюционным алгоритмом АСО на основе декомпозиции MOEA/D-АСО [29] и с методом на основе нечеткой логики FL-MTSP [30]. Хотя предложенный метод превосходил эти методы по качеству решений, он демонстрировал более медленную работу по сравнению с FL-MTSP.

Методы на основе алгоритма искусственной пчелиной колонии. Искусственная пчелиная колония – это алгоритм оптимизации, построенный на основе поведения роя медоносных пчёл при поиске пищи. ABC также относится к популяционным методам, в которых местоположение источника пищи в пространстве решений представляет собой потенциальное решение задачи оптимизации, а количество нектара данного источника пищи соответствует пригодности решения.

Венкатеш и соавторы [31] использовали алгоритм ABC для решения задачи МКВ с одним депо, в которой необходимо минимизировать суммарную длину маршрутов агентов и максимальную длину маршрута. Авторы предложили использовать локальный поиск для улучшения полученных результатов. Хотя авторы доказали преимущества предложенного метода, они не привели оценку времени расчета.

В работе [32] рассматривается «цветная» задача МКВ, где городам назначены цвета, и каждый агент имеет право посещать города определённых цветов. Авторы работы предложили решение на основе модифицированного алгоритма ABC.

Рыночные методы. Также известны как аукционные методы, где предметом аукциона являются города задачи МКВ, а агенты являются участниками аукциона. Аукционы могут быть либо централизованными с центральным аукционистом, который берёт на себя задачу проведения аукциона и назначает города агентам с наибольшей ставкой, либо распределёнными, где процесс аукциона разделен между разными аукционистами для снижения нагрузки на каналы связи.

Авторы в работе [33] предложили рыночный распределенный и динамический алгоритм решения задачи МКВ, в котором каждый робот выбирает города для посещения следующим образом. Во-первых, каждый робот использует евклидово расстояние в качестве функции стоимости для выбора соответствующих городов посещения и планирует путь их обхода. Затем он объявляет аукцион с одним городом своего маршрута. Выбор наиболее подходящего робота для конкретной задачи осуществляется с помощью протокола на основе аукциона, называемому CNP (Contract Net Protocol). Результаты моделирования показали, что предложенное решение эффективно с точки зрения масштабируемости, общей длины пути и затрат на связь.

В [34] задача целераспределения в многоагентной системе, представленной несколькими роботами, была сформулирована как задача МКВ и решена с помощью метода кластеризации К-средних с аукционным процессом. Целью решения является минимизация общего пройденного расстояния и максимизация равномерности длины маршрутов роботов. Сначала с помощью алгоритма К-средних формируются n групп задач, где n — количество роботов. Затем каждый робот вычисляет стоимость посещения каждого кластера, сформированного на предыдущем шаге. На этапе аукциона роботы делают ставки на кластеры, и каждый кластер назначается роботу с самой высокой ставкой. Однако сложность алгоритма достаточно высока, так как рассматриваются все возможные комбинации назначений кластеров роботам. Поэтому данный подход может быть неэффективен для решения крупномасштабных МКВ. Кроме того, метод работает только в случае, если роботы находятся в разных позициях, в противном случае все роботы будут иметь одинаковые ставки на каждый кластер.

На основе алгоритма консенсусного пакета (СВВА) [35] и метода с агентами прогнозирования (MALA) [36], авторы [37] предложили метод решения задачи МКВ. В основе метода лежит повторяющаяся рыночная процедура, в ходе которой роботы на каждой итерации выполняют следующие шаги: рыночный аукцион, торговля между агентами, переключение агентов и отказ от агентов. На

этапе рыночного аукциона каждый агент выбирает задачи, исходя из их стоимости. Во время торговли агенты проверяют задачи других агентов, чтобы определить, смогут ли выполнить их с меньшими затратами. Фаза переключения агентов направлена на исследование решений, выходящих за пределы окрестности найденного локального минимума. Алгоритм завершается после серии итераций без улучшения качества найденного решения.

Cheikhrouhou и другие в работах [38, 39] предложили рыночный подход, называемый Move-and-Improve для решения задачи целераспределения и маршрутизации. Подход Move-and-Improve состоит из четырех основных фаз: (1) первоначальное распределение целей, (2) построение маршрута, (3) согласование конфликтующих целей, (4) улучшение решения. Move-and-Improve был смоделирован с использованием MATLAB и симулятора Webots и развернут на реальных роботах с применением операционной системы робота (ROS).

Другие методы. Далее рассматриваются некоторые подходы к решению задачи МКВ, в которых применяются различные методы, такие как теория вероятностей, теория игр, нечеткая логика и другие.

Хуфи и соавторы [40] предложили многоцелевую задачу оптимизации для определения маршрутов роботов, которые собирают данные с беспроводных сенсорных узлов и доставляют их в депо, принимая во внимание такие ограничения, как задержки доставки данных, ограниченные энергетические ресурсы роботов и ограниченное количество самих роботов. Оптимизация осуществляется с использованием теории игр. Предложенная модель основана на игре формирования коалиций, которая направлена на минимизацию времени обхода, оптимизацию числа задействованных роботов и балансировку нагрузки между ними. Предложенный метод сравнивался с методом на основе генетического алгоритма и показал свое преимущество, однако время расчета решения оказалось высоким.

Для решения многоцелевой задачи МКВ Тригуи и соавторы [41] предложили метод, основанный на нечеткой логике (FL-MTSP). В работе учитываются две цели – минимизация общей длины маршрутов и минимизация максимальной

длины маршрута. Эффективность FL-MTSP была подтверждена сравнением с методом на основе генетического алгоритма [42].

Cheikhrouhou и соавторы [43, 44] предложили метод АНР-MTSP, основанный на аналитическом иерархическом процессе АНР [45]. В этом методе сначала вычисляются веса для каждого города, с использованием метода АНР. Затем города распределяются между агентами. Наконец, третий этап включает в себя совершенствование решений, полученных на втором этапе. Сравнение АНР-MTSP с несколькими другими методами как FL-MTSP [46] показывает его превосходство.

Объем пространства решений МКВ с n задачами и m агентами можно рассчитать с помощью числа Лаха – представляет собой коэффициент, выражающий возрастающие факториалы через падающие факториалы и наоборот. Они были обнаружены Иво Лахом в 1954 году [47]. Беззнаковые числа Лаха имеют важное применение в комбинаторике: они подсчитывают количество способов разделить множество из n элементов на m непустых упорядоченных подмножеств (то есть учитывается порядок элементов внутри каждого подмножества). Беззнаковое число Лаха вычисляется по следующей формуле:

$$L(n, m) = \left(\frac{n!}{m!} \right)^2 \cdot \frac{m}{n(n-m)!}. \quad (1.4)$$

На рисунке 1.5 представлены значения размера пространства решений при использовании подхода одновременной оптимизации для экземпляров с $n \in \{0, 1, 2, \dots, 10\}$ городами и $m \in \{0, 1, 2, \dots, 10\}$ агентами, полученные с помощью формулы 4.

Оценка размера пространства решений позволяет понять, насколько трудоёмким является процесс решения задачи МКВ, особенно для точных и метаэвристических методов.

$n \setminus m$	0	1	2	3	4	5	6	7	8	9	10
0	1										
1	0	1									
2	0	2	1								
3	0	6	6	1							
4	0	24	36	12	1						
5	0	120	240	120	20	1					
6	0	720	1800	1200	300	30	1				
7	0	5040	15120	12600	4200	630	42	1			
8	0	40320	141120	141120	58800	11760	1176	56	1		
9	0	362880	1451520	1693440	846720	211680	28224	2016	72	1	
10	0	368800	16329600	21772800	12700800	3810240	635040	60480	3240	90	1

Рисунок 1.5. Количество решений МКВ при использовании похода одновременной оптимизации

1.2. Подход на основе предварительной кластеризации

Второй подход к решению задачи МКВ основан на двухэтапной стратегии. На первом этапе множество городов задачи разбивается на кластеры, число которых, как правило, равно числу агентов задачи. На втором этапе определяется порядок посещения городов для каждого агента таким образом, чтобы оптимизировать заданный критерий качества решения [48]. Данный подход сводит задачу МКВ к нескольким задачам КВ, что позволяет существенно сократить размер пространства решений [49-51].

Одной из первых работ, в которой для решения задачи МКВ была использована кластеризация, является работа авторов [52], где была применена схема аттрактора соседства в сочетании с различными эвристическими и эволюционными алгоритмами. Полученные комбинации были протестированы на трёх различных задачах, а результаты открыли новое направление исследований в области решения МКВ. С тех пор было предложено множество методов, объединяющих кластеризацию с одним или несколькими эвристическими алгоритмами оптимизации.

Чандра и др. [53] разработали собственный алгоритм кластеризации для разделения набора городов МКВ. Однако этот алгоритм не оценивался с точки зрения того, насколько он повышает качество решения задачи МКВ; его анализ

ограничился оценкой качества кластеров с использованием одной метрики: относительного расстояния городов до центра кластера.

В работе [54] Ма и др. разработали алгоритм кластеризации с методом целочисленного линейного программирования, который использовался для решения задачи МКВ. Производительность разработанного алгоритма кластеризации оценивалась только по критерию суммарной длины маршрутов.

Авторы в [55] также рассмотрели проблему целераспределения, сведя ее к постановке задачи МКВ и предложили для ее решения алгоритм динамического и распределённого PSO. Метод решения состоит из двух фаз: на первой фазе задачи группируются в кластеры, а на второй — кластеры назначаются агентам. Предложенный алгоритм сравнивался с распределённым алгоритмом PSO и генетическим алгоритмом.

В работе [56] задача МКВ решается с использованием комбинации метода К-средних и кроссовера АСО, основанного на подходе Лахата [57]. Для повышения производительности муравьиного алгоритма реализован кроссовер феромонов — оператор генетического алгоритма, применяемый для улучшения следа феромонов. Это позволяет более сбалансированно сочетать эксплуатацию и исследование, когда муравьи выбирают следующий город.

Кластерный рыночный алгоритм (CM-MTSP) [58] предложен для решения МКВ и состоит из трёх этапов: кластеризация, аукцион и улучшение решения. На этапе кластеризации центральный агент (аукционер) формирует кластеры с помощью алгоритма k-средних. Затем остальные агенты делают ставки на каждый кластер, рассчитывая стоимость и отправляя её аукционеру. В итоге аукционер назначает каждый кластер агенту с максимальной ставкой. На этапе улучшения решения происходит преобразование кластеров с целью получения более качественного решения по критериям общей длины маршрутов, максимальной длины маршрута и времени выполнения миссии. Однако необходимость последнего отдельного этапа кластеризации не обоснована, поскольку перестановка существенно изменяет исходное распределение задач и может сде-

лать предварительное формирование кластеров избыточным. Таким образом, эффективность и целесообразность разделения на эти этапы требуют дополнительного анализа.

Проблема использования подхода СКЗМ для решения задачи МКВ заключается в том, что разделение городов между агентами, выполняемое на этапе кластеризации, может быть неоптимальным с точки зрения критерия общей длины маршрутов, поскольку алгоритмы кластеризации оценивают кластеры на основе критерия суммарного расстояния городов в кластере от центра кластера, что не является показателем для получения минимальной длины внутрикластерного маршрута. Кроме того, данный подход, как правило, не включает в себя механизм оптимизации критерия MinMax.

Размер пространства решений МКВ с n городами и m агентами для данного подхода можно оценить с помощью чисел Стирлинга второго рода [59], которые определяют количество способов разбиения множества n элементов на m непустых подмножеств (в отличие от числа Лаха, порядок элементов этих подмножеств не имеет значения), следующим образом:

$$S(n, m) = \frac{1}{m!} \sum_{i=0}^m \left((-1)^i \frac{m!}{i!(m-i)!} (m-i)^n \right). \quad (1.5)$$

На рисунке 1.6 представлены значения числа Стирлинга для $n \in \{0, 1, 2, \dots, 10\}$ и $m \in \{0, 1, 2, \dots, 10\}$.

Следовательно, размер пространства решений МКВ представляет собой сумму количества способов разделения множества n на m непустых подмножеств (см. формула 1.5) и количества способов упорядочить элементы количеством B_i внутри каждого подмножества m :

$$R(n, m) = S(n, m) + \sum_{i=1}^m B_i!, \quad (1.6)$$

$n \backslash m$	0	1	2	3	4	5	6	7	8	9	10
0	1										
1	0	1									
2	0	1	1								
3	0	1	3	1							
4	0	1	7	6	1						
5	0	1	15	25	10	1					
6	0	1	31	90	65	15	1				
7	0	1	63	301	350	140	21	1			
8	0	1	127	966	1701	1050	266	28	1		
9	0	1	255	3025	7770	6951	2646	462	36	1	
10	0	1	511	9330	34105	42525	22827	5880	750	45	1

Рисунок 1.6. Количество способов разделения множество размером $n \in \{0,1,2,\dots,10\}$ на $m \in \{0,1,2,\dots,10\}$ непустых подмножеств.

Заметим, что количество способов упорядочить элементы внутри каждого подмножества зависит от числа этих элементов, которое, в свою очередь, является переменной, зависящей от результата этапа кластеризации. Однако можно оценить это значение сверху и снизу.

Нижняя оценка достигается, когда алгоритм кластеризации разделяет множество $n - 1$ (исключаем города депо) на m подмножеств равного размера. Таким образом, количество возможных маршрутов в каждом кластере равно

$\left(\frac{n-1}{m}\right)!$, и так как имеется m кластеров, то нижняя оценка количество возможных маршрутов равна $m \left(\frac{n-1}{m}\right)!$.

Верхняя оценка достигается, когда $m - 1$ подмножеств содержат по одному элементу, а одно подмножество содержит оставшиеся $n - 1 - m + 1 = n - m$ элементов. Таким образом, количество способов упорядочить элементы внутри кластеров попадает в промежуток:

$$\sum_{i=1}^m B_i! \in \left[m \left(\frac{n-1}{m}\right)!, (n-m)! \right]. \quad (1.7)$$

Подход «Сначала кластеризация, затем маршрутизация» является популярной стратегией решения МКВ, особенно для задач большого размера. Его ос-

новное преимущество заключается в уменьшении размера пространства решений за счёт разбиения задачи МКВ на более мелкие подзадачи КВ. Однако, несмотря на практическую эффективность, этот подход имеет важные ограничения. Качество решения сильно зависит от исходного этапа кластеризации, который, как правило, не учитывает целевую функцию задачи. Это может приводить к неоптимальному распределению задач между агентами и, следовательно, к снижению качества итогового маршрута по критериям MinSum и MinMax. Кроме того, отсутствует встроенный механизм балансировки маршрутов между агентами. В связи с этим, подход требует доработок — например, внедрение итеративных стратегий улучшения или интеграция кластеризации с глобальной оптимизацией распределения задач.

1.3. Подход на основе предварительной маршрутизации

Третий подход к решению МКВ реализуется следующим образом. Сначала определяется порядок посещения городов, в результате формируется маршрут, проходящий через все города задачи (супер-маршрут) с исключением города депо. Затем решается задача распределения городов между агентами, путём разделения супер-маршрута на несколько частей, количество которых равно количеству агентов, не меняя последовательности обхода городов. Таким образом, задача МКВ сводится к задаче одного коммивояжёра. Этот подход известен в литературе как «Route First – Cluster Second» [60]. Он использовался для решения задачи маршрутизации транспортных средств (Vehicle Routing Problem, VRP), которая, в отличие от задачи МКВ, учитывает вместимость и грузоподъёмность агентов.

Автор работы [61] использовал алгоритм Беллмана для направленных ациклических графов с целью разделения супер-маршрута. Два вложенных цикла с индексами i и j перебирают каждую часть супер-маршрута вида $(T_i, T_{i+1}, \dots, T_j)$ и вычисляют для неё общую загрузку и стоимость поездки, если в

ней нарушается ограничение на пропускную способность агента, то она отбрасывается.

В работе [62] авторы использовали жадные алгоритмы для разделения супер-маршрута: начиная с одного города, маршрут отрезается каждый раз, когда следующий узел не вписывается в остаточную вместимость агента. Такой подход приводит к неоптимальному результату.

Авторы [63] предложили нейросетевой подход, который был смоделирован как Марковский процесс принятия решений, в рамках которого нейронная сеть последовательно выбирает и разделяет рёбра супер-маршрута.

Авторы в работе [64] обучили нейронную сеть на основе модели внимания (Attention Model, AM) с использованием методов обучения с подкреплением, в результате чего, получили политику разбиения супер-маршрута π (называемой политикой SplitNet).

На основе проведённого обзора и анализа существующих методов решения многоагентной задачи коммивояжёра определена цель диссертационной работе – повышение показателей качества её решения, в частности минимизация суммарной длины маршрутов и максимальной длины индивидуального маршрута, а также сокращение времени расчёта.

Также сформулированы задачи диссертационного исследования:

1. Определить целевые критерии, характерные для практических задач группового управления.
2. Разработать методы решения задачи МКВ, сочетающие элементы кластеризации и маршрутизации, обеспечивающие компромисс между качеством решений и вычислительными затратами.
3. Провести экспериментальные исследования разработанных методов на тестовых задачах, оценить влияние числа агентов и заданий на качество решений, а также сравнить с существующими подходами по основным показателям качества.

1.4. Выводы по главе 1

В главе показано, что распределение пространственно-распределенных задач в многоагентной системе сводится к многоагентной задаче коммивояжера, где каждая задача определяется городом и каждый агент является коммивояжером. Соответственно, проведён обзор подходов, методов и алгоритмов решения задачи МКВ. Выявлено, что задача МКВ включает в себя 2 подзадачи: распределение городов между агентами и определение порядка посещения городов каждым агентом. Определены три основных подхода к решению задачи МКВ с точки зрения порядка выполнения ее подзадач: подход одновременной оптимизации, подход «Сначала кластеризация, затем маршрутизация» (СКЗМ) и подход «Сначала маршрутизация, затем кластеризация» (СМЗК).

Подход одновременной оптимизации позволяет найти оптимальное решение, но за счет значительных временных затрат.

Подход СКЗМ существенно уменьшает время расчета за счет разделения исходной задачи на более простые подзадачи, однако он не подразумевает конкретного механизма оптимизации критерия MinMax.

Подход СМЗК обладает более гибким механизмом оптимизации критерия MinMax, но за счет больших временных затрат, поскольку задача МКВ сводится к классической задаче КВ, а размер пространства решений составляет примерно $(n-1)!$, где n – количество городов задачи.

Глава 2. Разработка метода решения многоагентной задачи коммивояжера на основе предварительной маршрутизации

«Сначала Маршрутизация, Затем Кластеризация» (СМЗК) — это широко используемый эвристический подход для решения комбинаторных задач оптимизации, таких как многоагентная задача коммивояжера и задача маршрутизации транспортных средств. Ключевые шаги в подходе СМЗК включают:

1. Построение маршрута: решается задача коммивояжера или аналогичная задача маршрутизации с целью построения одного маршрута (супер-маршрута), охватывающего все города задачи, кроме города депо. Результатом является гамильтонов цикл, соединяющий все города в эффективном по заданному критерию порядке. Как правило, этот этап направлен на минимизацию длины гамильтонова цикла.

2. Разбиение маршрута: супер-маршрут разбивается на несколько сегментов (кластеров), каждый из которых назначается отдельному агенту. На этом этапе происходит минимизация целевых критериев, таких как максимальная длина индивидуального маршрута (MinMax) или общее время выполнения миссии.

Данный подход имеет ряд проблем и ограничений:

- компромисс между качеством и скоростью: получение оптимального (или субоптимального) решения задачи коммивояжера может быть вычислительно затратным при больших размерах задачи, поскольку количество возможных решений составляет $(n - 1)!$, где n — число городов.
- сложность разбиения: фаза кластеризации не всегда приводит к сбалансированным или эффективным подмаршрутам. Разделение супер-маршрута и присоединение каждого подмаршрута к городу-депо может привести к появлению неэффективных маршрутов, пересекающихся с самими собой из-за геометрического распределения городов и расположения города депо.

Для устранения проблемы разбиения предлагается этап улучшения маршрута с использованием алгоритма 2-opt [65] – эвристика локального поиска, применяемая для оптимизации решений задачи коммивояжёра и других задач маршрутизации путём устранения пересечений в маршруте. Алгоритм итеративно удаляет два пересекающихся ребра в маршруте и повторно соединяет образовавшиеся участки иным способом, сокращая общую длину пути и повышая эффективность маршрута.

Чтобы реализовать первую фазу подхода СМЗК – построение супермаршрута – необходимо использовать эффективный алгоритм для решения задачи КВ. Поскольку точные методы непригодны для крупных задач МКВ из-за высокой сложности проблемы, на практике применяются эвристические и метаэвристические алгоритмы. Одним из таких является муравьиный алгоритм, который подходит для построения начального маршрута за ограниченное время и обеспечивает требуемое качество решения. Ниже рассматривается его применение в контексте предлагаемого метода.

Рисунок 2.1 представляет блок схему алгоритма предлагаемого метода решения задачи МКВ.

Предлагаемый метод, реализующий подход СМЗК, будет обозначаться как (СМКЗ-МА), поскольку для решения этапа маршрутизации используется муравьиный алгоритм.

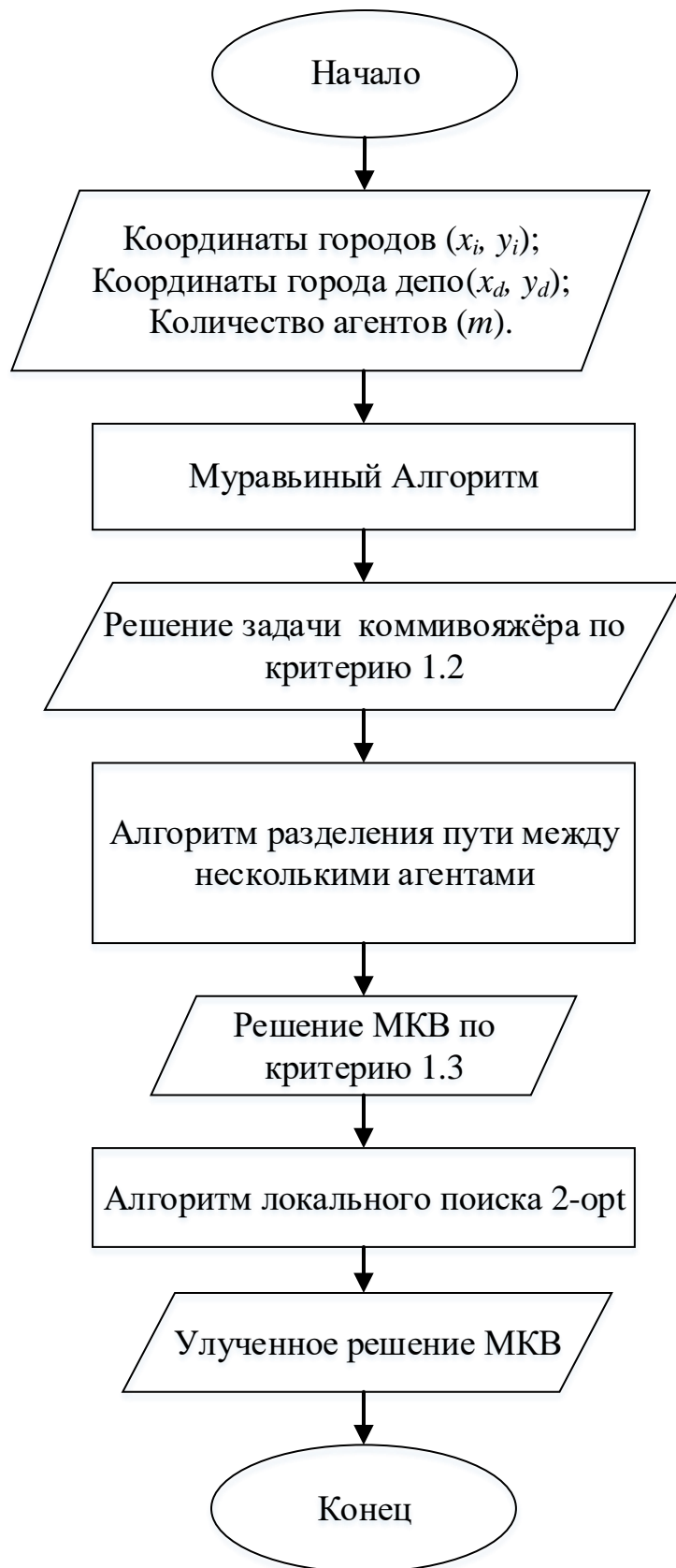


Рисунок 2.1. Блок схема алгоритма модифицированного метода решения МКВ

2.1. Этап 1: решение задачи коммивояжера.

В литературе представлено множество алгоритмов для решения задачи коммивояжера, включая генетические алгоритмы, биоинспирированные методы и другие подходы. В рамках данного исследования для решения задачи коммивояжера выбран муравьиный алгоритм, который доказал свою эффективность при решении комбинаторных задач маршрутизации.

Муравьиный алгоритм (МА) основан на поведении муравьёв при поиске пищи, которое представляет собой сложное коллективное явление, обусловленное выделением феромонов – собирательное название веществ, выделяемых некоторыми видами животных и обеспечивающих коммуникацию между особями одного вида. Феромоны привлекают других муравьёв, указывая путь к источнику пищи. Чем больше муравьёв проходит по определённому маршруту, тем выше концентрация феромонов на нём, и тем выше вероятность, что последующие муравьи выберут именно этот путь. Таким образом, кратчайший путь к цели определяется максимальной концентрацией феромонов, поскольку большее число муравьёв следует по нему [66]. Для устранения неоптимального пути (локальных минимумов), алгоритм предусматривает испарение феромона с течением времени. Уровни феромона на самом коротком пути остаются высокими, так как скорость осаждения феромона будет больше, чем скорость его испарения.

Алгоритм реализуется следующим образом. На рисунке 2.2 изображен пример среды передвижения, где каждая точка – это узел. От заданного узла i муравей имеет возможность перехода в другой узел j , если есть путь к нему, т.е. если i и j соединены пунктирной линией, исключая узлы, которые муравей уже посетил. В муравьином алгоритме действует вероятностно-пропорциональное правило, задающее вероятность перехода k -го муравья из узла i к узлу j , она зависит от уровня феромона между узлами i и j (τ_{ij}) и от заданной эвристики (η_{ij}), которая в основном выбирается как обратная величина расстояния между узлами i и j . Влияние τ_{ij} и η_{ij} на вероятности перехода регулируется двумя параметрами α и β последовательно [67]. Таким образом, и с учётом того, что $J_{i,k}$

– это список еще не посещенных узлов муравьем k , $P_{ij,k}$ – вероятность перехода с узла i в узел j рассчитывается следующей формулой:

$$P_{ij,k} = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in J_{i,k}} \tau_{ij}^\alpha \eta_{ij}^\beta} . \quad (2.1)$$

При $\alpha = 0$ эффект феромона будет проигнорирован в расчёте вероятностей следующего шага, на вероятности выбора какого-либо узла будет влиять только эвристика. Если эвристика есть расстояние между узлами, то выбор ближайшего узла будет самым вероятным, что соответствует жадному алгоритму в классической теории оптимизации. Если $\beta = 0$, то работает лишь феромонное усиление, что влечет за собой быстрое вырождение маршрутов к одному субоптимальному решению и достижение локального минимума.

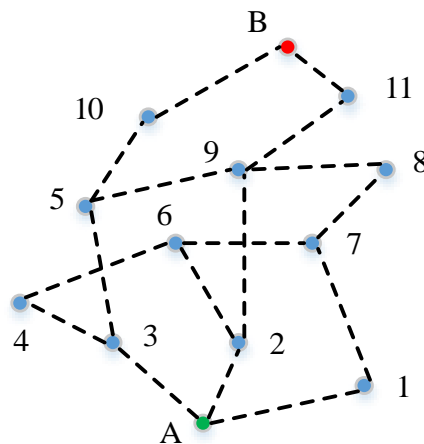


Рисунок 2.2 – Пример среды работы муравьиного алгоритма, узлы и пути между ними

Выбор следующего узла происходит по принципу колеса рулетки: каждый узел имеет свой сектор колеса, площадь которого пропорциональна вероятности $P_{ij,k}$, рассчитанной по формуле 2.1.

При достижении целевого узла, муравей k оставляет феромон $\Delta\tau$ между всеми узлами его маршрута T_k величиной, равной отношению константы интенсивности феромона Q к длине траектории L_k как показано в следующей формуле:

$$\Delta\tau_{ij,k} = \begin{cases} \frac{Q}{L_k} & \text{если } (i,j) \in T_k, \\ 0 & \text{если } (i,j) \notin T_k. \end{cases} \quad (2.2)$$

Чтобы исследовать все пространство решений и избежать попадания в локальный минимум, феромон, поставленный всеми муравьями K , должен испаряться в каждом цикле алгоритма t относительно заданного коэффициента испарения ρ в соответствии с выражением:

$$\tau_{ij}(t+1) = (1-\rho)t + \sum_{k=1}^K \Delta\tau_{ij,k}. \quad (2.3)$$

Входными параметрами алгоритма являются список координат местоположения городов C (задач) и координат города-депо ($c_d \in C$). Далее город-депо исключается из списка городов. Таким образом образуется список городов с исключением города депо C^- . Затем рассчитывается матрица расстояний каждого узла до остальных узлов $D = \|d_{ij}\|$, следовательно, рассчитывается параметр эвристики муравьиного алгоритма $\|n_{ij}\| = \left\| \frac{1}{d_{ij}} \right\|$.

Далее запускается муравьиный алгоритм, позволяющий получить супермаршрут (решение задачи коммивояжёра).

Рисунок 2.3 представляет блок схему муравьиного алгоритма, предназначенного для решения задачи коммивояжёра и получения супермаршрута.

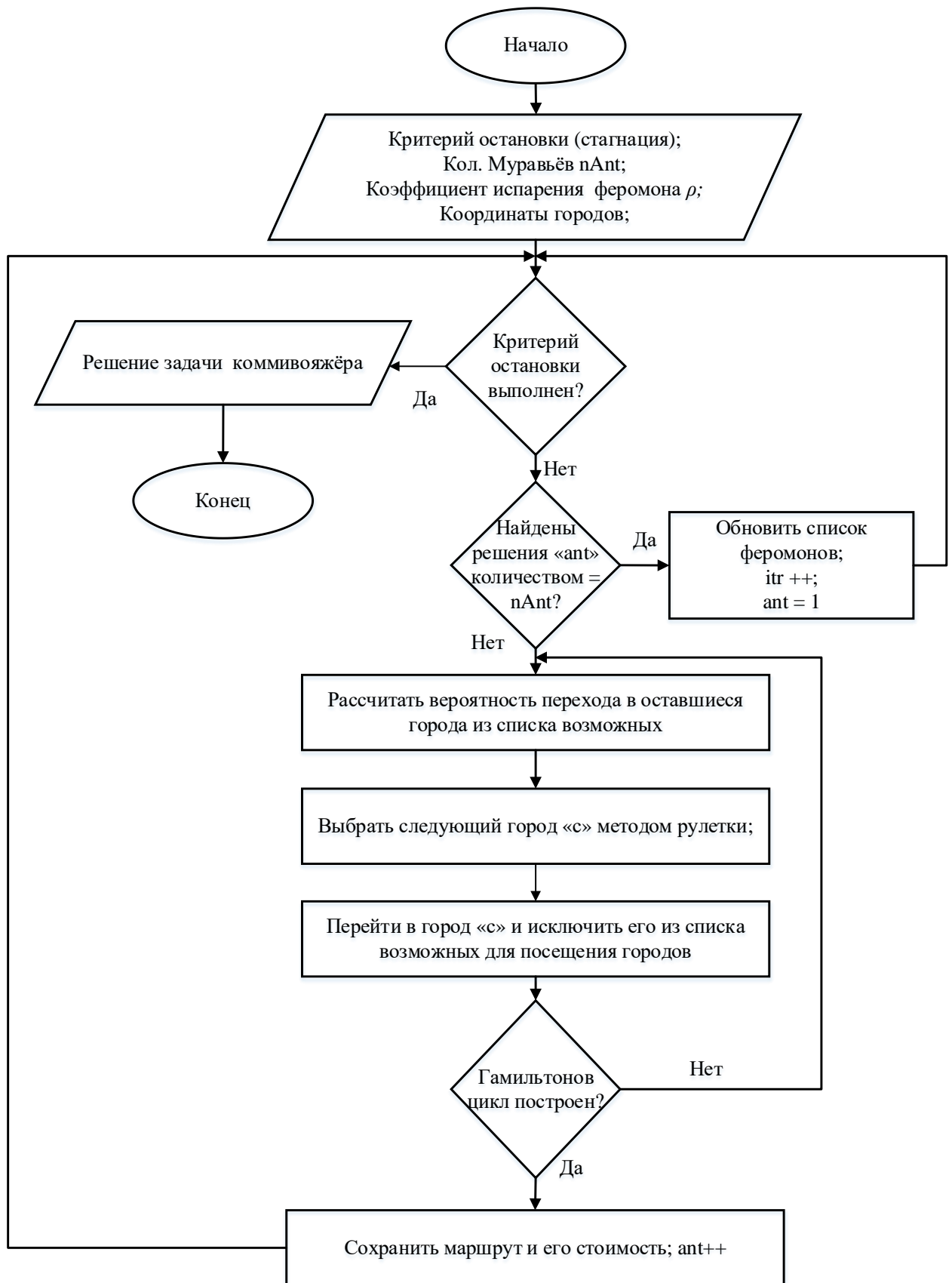


Рисунок 2.3. Блок схема Муравьиного алгоритма

2.2. Этап 2: разделение супер-маршрута между агентами

Разделение супер-маршрута между несколькими агентами представляет собой ключевой этап в подходе «Сначала Маршрутизация, Затем Кластеризация» при решении задачи МКВ. После построения супер-маршрута, необходимо разделить его между агентами, соблюдая заданные ограничения и оптимизируя целевые функции, такие как минимизация максимальной длины маршрута.

Как было показано в обзоре, для разделения супер-маршрута используются точные алгоритмы, позволяющие перебрать все возможные варианты разделения маршрута и выбрать оптимальный. Такие алгоритмы действительно способны найти оптимальное разбиение супер-маршрута, однако для этого требуется перебор большого количества вариантов, что делает задачу вычислительно затратной. Альтернативой являются жадные алгоритмы, где маршрут последовательно обрывается при превышении допустимой нагрузки каждого агента. Такие алгоритмы менее ресурсоёмки, но приводят к ухудшению качества решений из-за отсутствия глобального поиска.

В данной работе предлагается авторский алгоритм разбиения супер-маршрута, который объединяет идеи жадного подхода и метода перебора, что позволяет достичь баланса между качеством решений и вычислительной эффективностью [68].

Предлагаемый алгоритм разделения супер-маршрута реализуется следующим образом.

1) Сначала формируется матрица расстояний D^* между городами, в порядке их посещения в супер-маршруте. Например, на рисунке 2.4 представлен супер-маршрут, где c_i – города, p_i – расстояние между ними: $i \in [1,5]$.

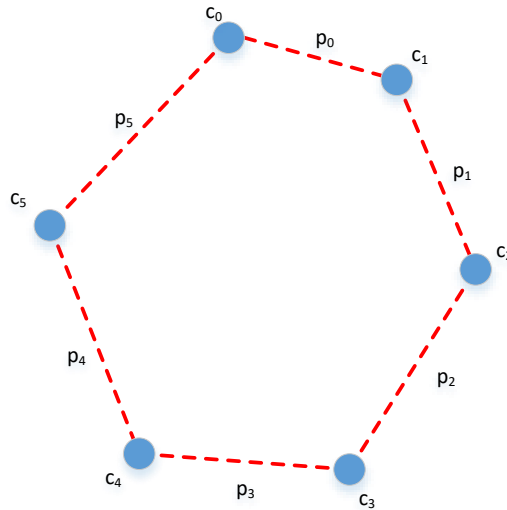


Рисунок 2.4. Пример супер-маршрута, используемого для построения матрицы расстояний

Матрица расстояний D^* для этого примера определяется следующим образом:

	c0	c1	c2	c3	c4	c5
c0	0	p_0	p_0+p_1	$p_0+p_1+p_2$	$p_0+p_1+p_2+p_3$	$p_0+p_1+p_2+p_3+p_4$
c1	$p_1+p_2+p_3+p_4+p_5$	0	p_1	p_1+p_2	$p_1+p_2+p_3$	$p_1+p_2+p_3+p_4$
c2	$p_2+p_3+p_4+p_5$	$p_2+p_3+p_4+p_5+p_0$	0	p_2	p_2+p_3	$p_2+p_3+p_4$
c3	$p_3+p_4+p_5$	$p_3+p_4+p_5+p_0+p_1$	$p_3+p_4+p_5+p_0+p_1$	0	p_3	p_3+p_4
c4	p_4+p_5	$p_4+p_5+p_0+p_1+p_2$	$p_4+p_5+p_0+p_1+p_2$	$p_4+p_5+p_0+p_1+p_2$	0	p_4
c5	p_5	p_5+p_0	$p_5+p_0+p_1$	$p_5+p_0+p_1+p_2$	$p_5+p_0+p_1+p_2+p_3$	0

2) Затем рассчитывается приблизительная длина маршрута d^* , которую должен пройти каждый агент, как отношение общей длины супер-маршрута к количеству агентов.

3) Далее выполняется последовательный перебор городов для первого агента, начиная с первого города (Γ_1 шаг 1 рисунок 2.5), с целью найти такой город, до которого расстояние по кусочно-ломаной траектории максимально

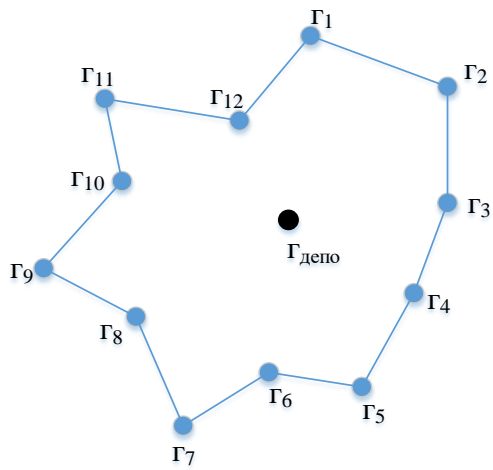
близко к d^* (Γ_4 шаг 1 рисунок 2.5). Для этого используется ранее созданная матрица расстояний между всеми парами городов.

Первый город для следующего агента выбирается как следующий за последним городом предыдущего агента (Γ_5 шаг 2 рисунок 2.5), и процесс повторяется до тех пор, пока все города не будут распределены (шаг 3 на рисунке 2.5). После этого вычисляется и сохраняется сумма длин маршрутов всех агентов. На этом завершается первый цикл процесса перебора. В следующем цикле в качестве начального города для первого агента выбирается второй город из супермаршрута, и описанный ранее процесс повторяется, формируя новые маршруты, отличающиеся от полученных на предыдущем цикле.

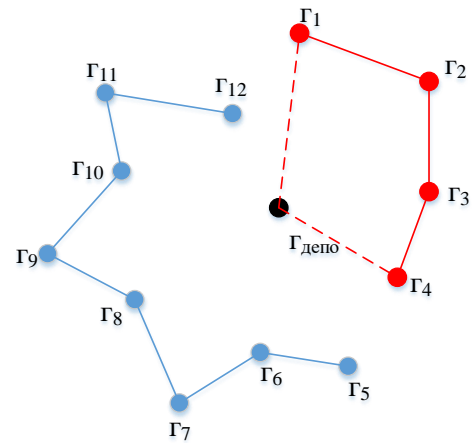
Начальный город первого агента изменяется циклически до тех пор, пока не будут рассмотрены все возможные варианты разбиения. В результате сохраняется то решение, которое обеспечивает минимальную максимальную длину среди маршрутов.

Таким образом, размер пространства решений задачи МКВ при использовании предлагаемого алгоритма разделения в задачи с n городами составляет $(n-1)! + n - 1$, поскольку существует $(n-1)!$ способов построения супер маршрута и $n - 1$ способов разделения его (алгоритм разделения повторяется $n - 1$ раз).

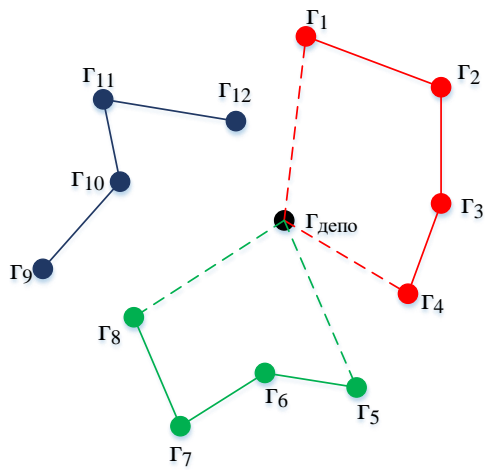
Рисунок 2.6 представляет блок схему алгоритма разделения супермаршрута между агентами.



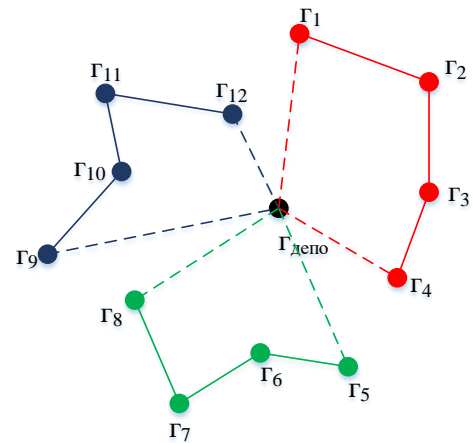
Шаг 0



Шаг 1



Шаг 2



Шаг 3

Рисунок 2.5. Пример работы одной итерации алгоритма разделения пути между тремя коммивояжёрами

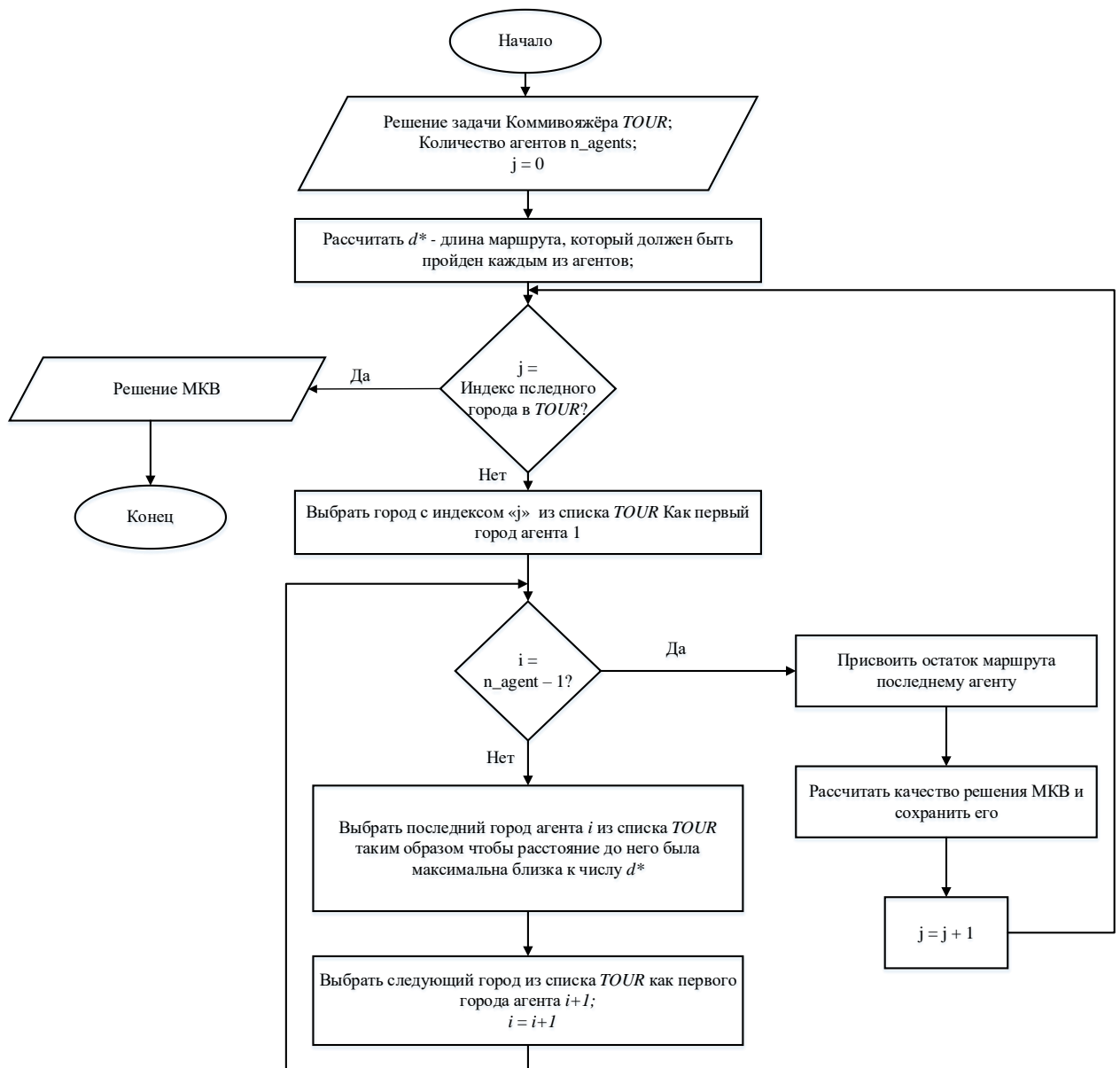


Рисунок 2.6. Блок схема алгоритма разделения супермаршрута

2.3. Этап 3: улучшение маршрутов с помощью алгоритма локального поиска

Изначально сформированные решения с помощью муравьиного алгоритма не всегда являются оптимальными. Это связано с вероятностной природой муравьиного алгоритма. Кроме того, после разделения супер-маршрута и присоединения каждого сегмента к городу депо может возникнуть ситуация, когда маршрут пересекается с самим собой. Такое поведение обусловлено геометрическим расположением городов и положением города депо.

Подобная ситуация показана на рисунке 2.7, где точки и пунктирные линии одного цвета обозначают города и маршрут одного агента. Красные круги выделяют участки, в которых маршрут пересекается с самим собой, что указывает на его отклонение от оптимального решения.

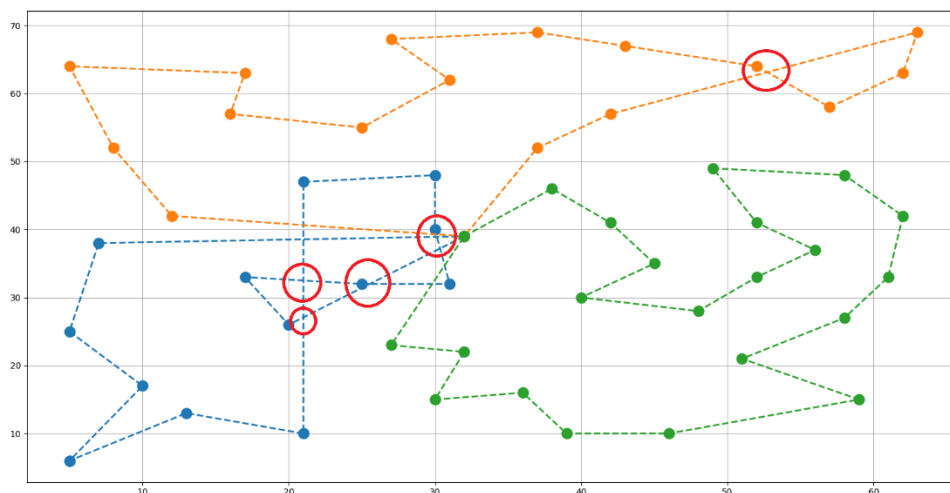


Рисунок 2.7. Пример где маршрут агента пересекается с самим собой

Для устранения проблемы пересечений в маршрутах агентов и повышения качества решений после этапа разбиения супер-маршрута предлагается использовать алгоритм 2-opt – классическая эвристика локального поиска, широко применяемая в задачах планирования пути и маршрутизации. Алгоритм направлен на поэтапное улучшение маршрутов путём устранения самопересечений и сокращения общей длины пути. Применяя 2-opt к каждому подмаршруту отдельно, можно получить более короткие маршруты движения агентов, пример работы алгоритма 2-opt показан на рисунке 2.8. алгоритм 1 представляет псевдокод алгоритма 2-opt.

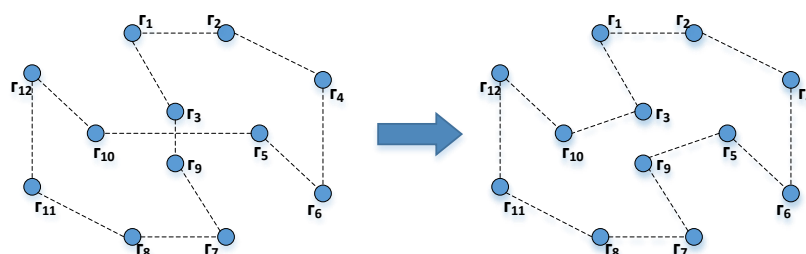


Рисунок 2.8. Пример работы алгоритма 2-opt по устранению пересечений маршрута

Алгоритм 1 алгоритм 2-opt

```
1 Процедура 2-opt(решение):
2   Лучший маршрут ← решение
3   для каждой пары рёбер  $(i, i+1)$  и  $(j, j+1)$  в текущем лучшем маршруте
   выполнить:
4     если рёбра  $(i, i+1)$  и  $(j, j+1)$  пересекаются тогда
5       обратить порядок узлов на участке маршрута от  $i+1$  до  $j$ 
6       Лучший маршрут ← обновлённый маршрут
7       рекурсивно вызвать 2-opt (Лучший маршрут) для дальнейшего
   улучшения
8     конец если
9   вернуть Лучший маршрут
10  конец процедуры
```

2.4. Моделирование и результаты

В данном параграфе представлены результаты применения предлагаемого метода при решении нескольких примерах МКВ, которые входят в библиотеку эталонных задач коммивояжёра (TSPLIB) [15]. Основное внимание уделяется оценке эффективности разработанного метода с точки зрения качества найденных решений и времени вычислений. Для оценки производительности предлагаемого метода проведено два сравнительных анализа результатов. В первом сравнивались две версии метода, описанного в данной главе: первая версия с дополнительным этапом улучшения решения, вторая версия без него. Во втором предлагаемый метод сравнивался с методами, реализующими подходы одновременной оптимизации и СКЗМ.

2.4.1. Оценка влияния локальной оптимизации на эффективность предлагаемого метода

Для количественной оценки эффективности предлагаемой модификации проведено сравнительное исследование двух версии предлагаемого метода СМЗК-МА: одна включает этапы, описанные в параграфах 2.1, 2.2 и 2.3 (СМЗК-

МА-2opt), вторая включает этапы, описанные в параграфах 2.1 и 2.2 с исключением 2.3 (СМЗК-МА). Цель исследования – выявить влияние дополнительного этапа локальной оптимизации на ключевые метрики: суммарную длину маршрутов, максимальную длину индивидуального маршрута и время расчёта [68].

Для проведения вышеуказанного исследования были выбраны три примера из библиотеки эталонных задач коммивояжёра TSPLIB — eil51, kroA100 и kroA150, содержащие соответственно 51, 100 и 150 городов. Для каждой из этих задач были заданы три сценария с числом агентов 3, 5 и 10. Каждый сценарий решался по 100 раз обеими версиями исследуемого метода, что позволило получить репрезентативную выборку.

Количество муравьёв выбрано на основе предварительного тестирования на подмножестве задач [69]. В качестве критерия останова муравьиного алгоритма использовалась стагнация — 20 последовательных итераций алгоритма без улучшения найденного решения.

Таблицы 2.1, 2.2 и 2.3 последовательно представляют результаты решения задач eil51, kroA100 и kroA150 с 3, 5 и 10 агентами (m). В таблицах приведены среднее по 100 экспериментам: время расчета решения в секундах (μ_t), сумма длин маршрутов в метрах (μ_{sum}) и максимальная длина маршрута среди агентов в метрах (μ_{max}). Соответствующие наилучшие значения выделены жирным шрифтом.

Таблица 2.1. Результаты решения задачи eil51.

m	3 агента		5 агентов		10 агентов	
Метод	СМЗК-МА	СМЗК-МА-2opt	СМЗК-МА	СМЗК-МА-2opt	СМЗК-МА	СМЗК-МА-2opt
μ_t	5.106	5.122	5.318	5.334	10.352	10.377
μ_{sum}	523.482	513.671	607.640	596.320	840.117	828.541
μ_{max}	177.462	176.570	126.495	127.267	93.342	92.903

Таблица 2.2. Результаты решения задачи kroA100.

m	3 агента		5 агентов		10 агентов	
Метод	СМЗК- МА	СМЗК- МА-2opt	СМЗК- МА	СМЗК- МА-2opt	СМЗК- МА	СМЗК- МА-2opt
μ_t	49.023	49.094	42.970	43.037	60.069	60.145
μ_{sum}	27133.520	26828.499	32732.657	32299.114	45052.645	44314.791
μ_{max}	9217.113	9227.610	6906.940	7138.400	5342.015	5377.809

Таблица 2.3. Результаты решения задачи kroA150.

m	3 агента		5 агентов		10 агентов	
метод	СМЗК- МА	СМЗК- МА-2opt	СМЗК- МА	СМЗК- МА-2opt	СМЗК- МА	СМЗК- МА-2opt
μ_t	182.097	182.480	173.852	174.246	207.991	208.386
μ_{sum}	32481.369	32227.959	37511.698	37308.209	50295.477	49573.774
μ_{max}	10959.406	11069.340	7741.485	8102.608	5759.024	5773.038

Ниже представлены результаты статистического анализа 100 экспериментов для каждого из сценариев в виде преимущества использования предлагаемого метода с алгоритмом 2-opt по сравнению с его версией без алгоритма 2-opt [68].

Преимущество Δ при использовании версии метода с алгоритмом 2-opt m_p в отношении версии без 2-opt m_2 в процентах по каждому из трех критериев оценки Ψ_x (по среднему времени вычисления среди 100 запусков по времени расчета решения в секундах (μ_t), по сумме длин путей в метрах (μ_{sum}) и по максимальной длине пути среди агентов в метрах (μ_{max})), можно рассчитать следующим образом:

$$\Delta_{\Psi_x} = \left(1 - \frac{\Psi_x^{(m_p)}}{\Psi_x^{(m_2)}} \right) \cdot 100. \quad (2.4)$$

Скрипичный график – это сочетание диаграммы размаха и графика плотности, развернутых и расположенных по обе стороны для отображения формы распределения данных.

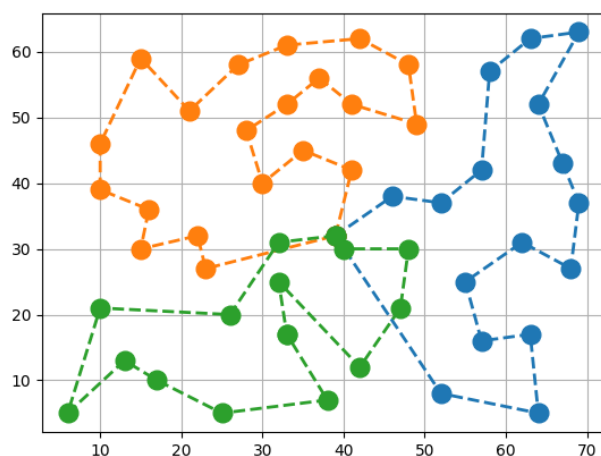
На рисунках 2.10-2.12, 2.14-2.16, 2.18-2.20 представлены скрипичные диаграммы, отражающие процентное преимущество метода с 2-opt по времени расчёта, суммарной длине маршрута и максимальной длине маршрута (слева направо) по сравнению с методом без 2-opt. Ширина диаграммы отражает количество экспериментов с соответствующим значением преимущества. Отрицательное значение преимущества означает, что в некоторых случаях предлагаемая модификация не улучшает, а, напротив, ухудшает результат по сравнению с методом без модификации.

eil51

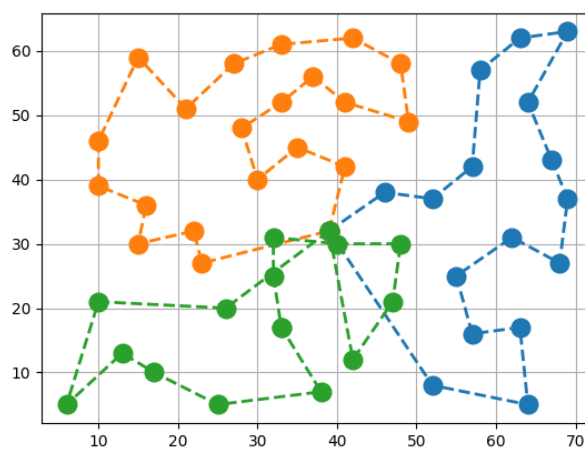
Статистический анализ 100 экспериментов задачи eil51 с 3 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.78, -0.11]$ %, в среднем составляя -0.34% ;
- суммарная длина пути агентов изменяется в диапазоне $[-1.2, 5.94]$ %, в среднем уменьшаясь на 2.02% ;
- максимальная длина пути агентов изменяется в диапазоне $[-3.56, 5.39]$ %, при среднем уменьшении на 0.6% .

Рисунок 2.9 иллюстрирует примеры работы версии метода, реализующего подхода СМЗК: а) с 2-opt, б) без 2-opt — в сценарии с 3 агентами и 51 городом.



а) СМЗК-МА-2opt



б) СМЗК-МА

Рисунок 2.9. Примеры решения сценария с 3 агентами и 51 городом

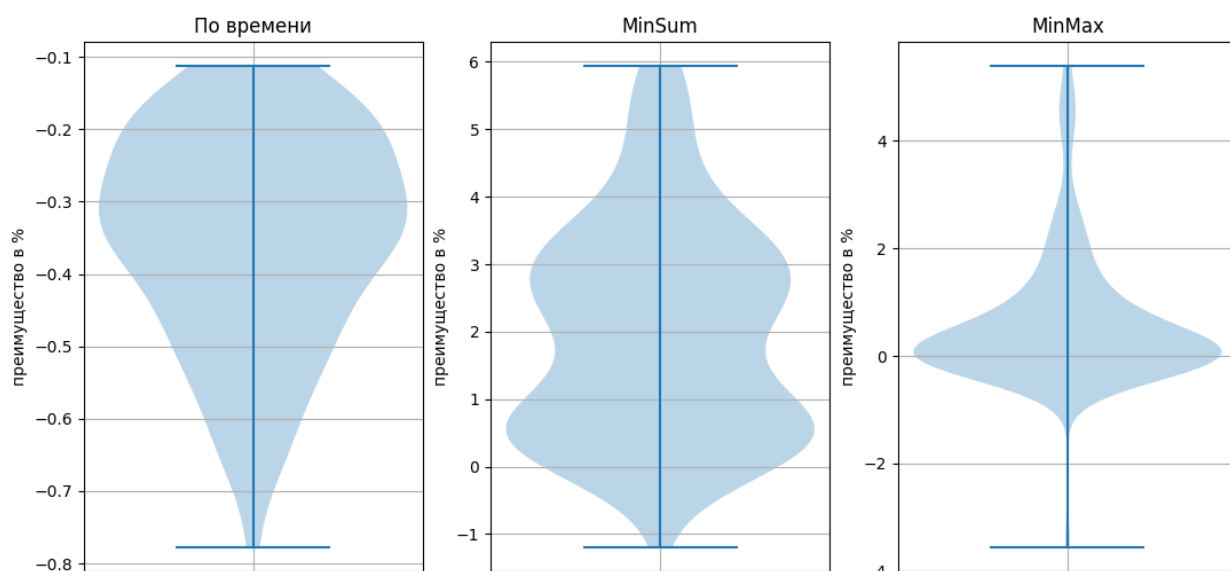


Рисунок 2.10. Скрипичные диаграммы преимуществ метода СМЗК-МА-2opt по сравнению с СМЗК-МА в задачи eil51 с 3 агентами

Статистический анализ 100 экспериментов задачи eil51 с 5 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.85, -0.09]$ %, в среднем составляя -0.31% ;
- суммарная длина пути агентов изменяется в диапазоне $[-2.88, 8.74]$ %, в среднем уменьшаясь на 2.69% ;
- максимальная длина пути агентов изменяется в диапазоне $[-13.27, 5.36]$ %, при среднем уменьшении на 0.48% .

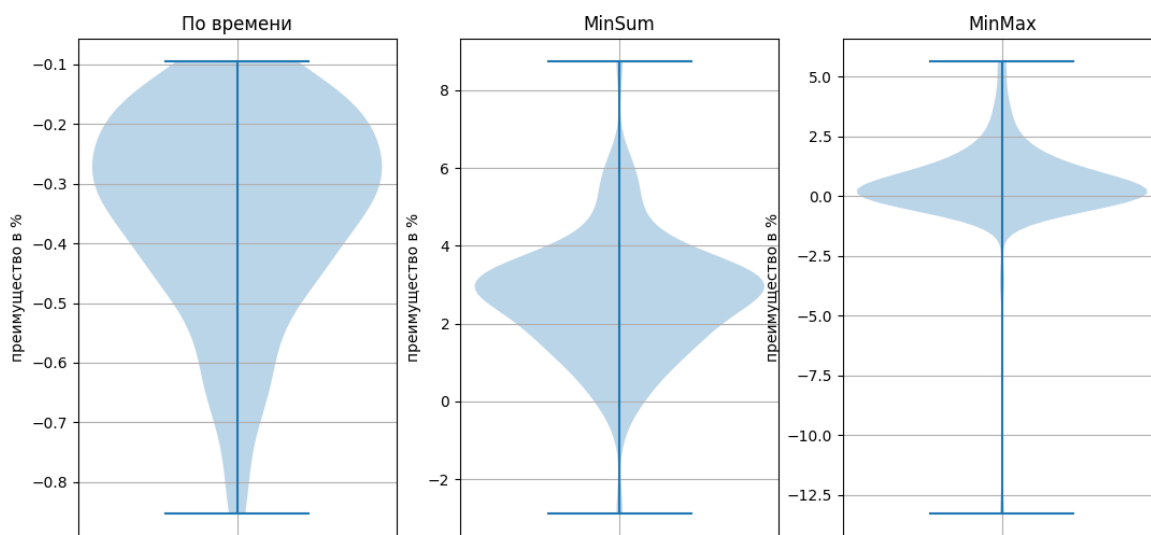


Рисунок 2.11. Скрипичные диаграммы преимуществ метода СМЗК-МА-2opt по сравнению с СМЗК-МА в задачи eil51 с 5 агентами

Статистический анализ 100 экспериментов задачи eil51 с 10 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-1.12, 0.96]$ %, в среднем составляя -0.25% ;
- суммарная длина пути агентов изменяется в диапазоне $[-2.49, 3.84]$ %, в среднем уменьшаясь на 1.81% ;
- максимальная длина пути агентов изменяется в диапазоне $[-8.77, 5.39]$ %, при среднем уменьшении на 0.69% .

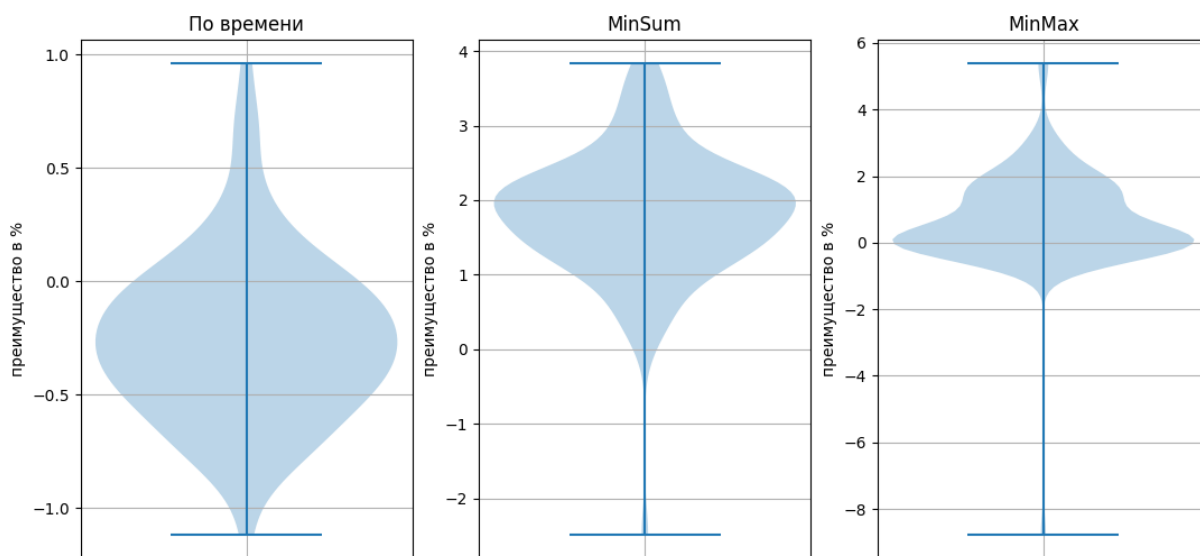


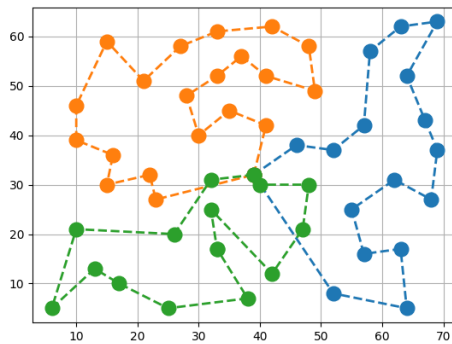
Рисунок 2.12. Скрипичные диаграммы преимуществ метода СМЗК-МА-2opt по сравнению с СМЗК-МА в задачи eil51 с 10 агентами

КроА100

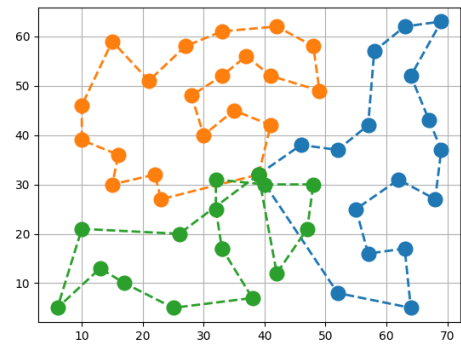
Статистический анализ 100 экспериментов задачи КроА100 с 3 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.4, -0.04]$ %, в среднем составляя -0.16 %;
- суммарная длина пути агентов изменяется в диапазоне $[-0.74, 4.48]$ %, в среднем уменьшаясь на 1.3 %;
- максимальная длина пути агентов изменяется в диапазоне $[-2.13, 4.96]$ %, при среднем уменьшении на 0.27 %.

Рисунок 2.13 иллюстрирует примеры работы версии метода, реализующего подхода СМЗК: а) с 2-опт, б) без 2-опт — в сценарии с 3 агентами и задачей на 100 городов.



а) СМЗК-МА-2опт



б) СМЗК-МА

Рисунок 2.13. Примеры решения сценария с 3 агентами и 100 городами

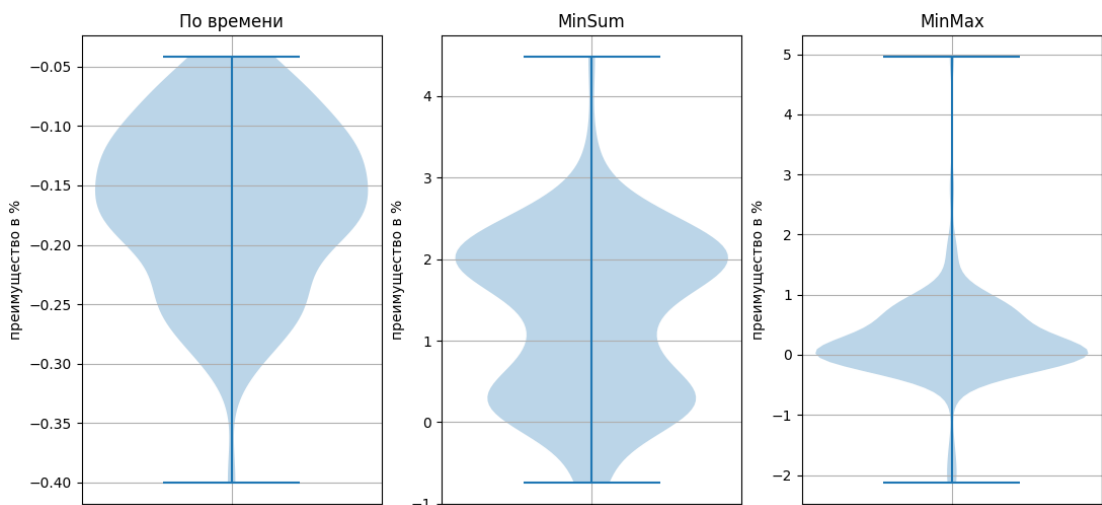


Рисунок 2.14. Скрипичные диаграммы преимуществ метода СМЗК-МА-2опт по сравнению с СМЗК-МА в задачи КроА100 с 3 агентами

Статистический анализ 100 экспериментов задачи KroA100 с 5 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.34, -0.05]$ %, в среднем составляя -0.16% ;
- суммарная длина пути агентов изменяется в диапазоне $[-2.08, 5.55]$ %, в среднем уменьшаясь на 2.09% ;
- максимальная длина пути агентов изменяется в диапазоне $[-11.47, 3.03]$ %, при среднем уменьшении на 0.83% .

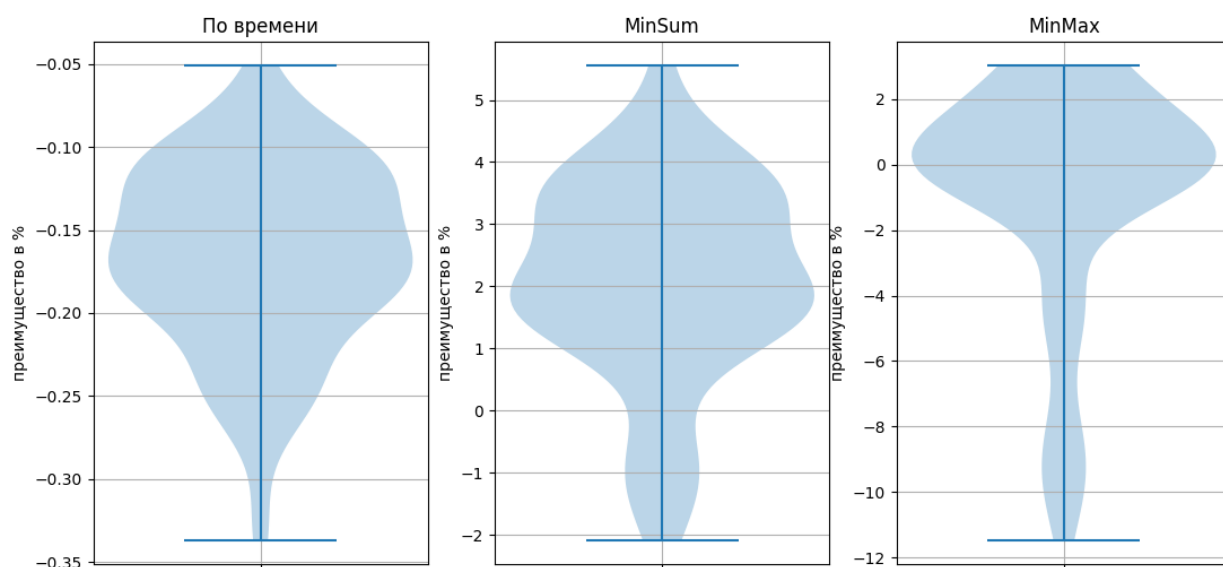


Рисунок 2.15. Скрипичные диаграммы преимуществ метода СМЗК-МА-2opt по сравнению с СМЗК-МА в задачи KroA100 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA100 с 10 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.63, 0.42]$ %, в среднем составляя -0.13% ;
- суммарная длина пути агентов изменяется в диапазоне $[-0.57, 3.79]$ %, в среднем уменьшаясь на 1.92% ;
- максимальная длина пути агентов изменяется в диапазоне $[-6.06, 0.44]$ %, при среднем увеличении на 0.22% .

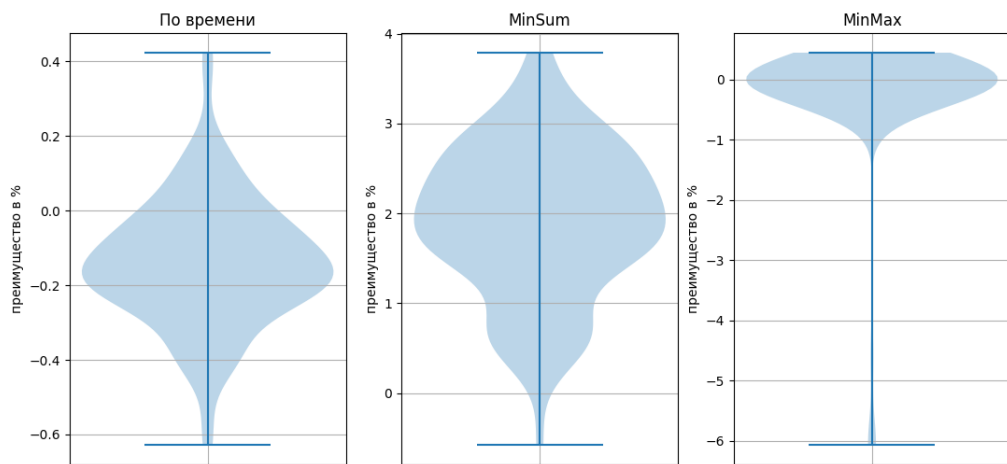


Рисунок 2.16. Скрипичные диаграммы преимуществ метода СМЗК-МА-2opt по сравнению с СМЗК-МА в задаче KroA100 с 10 агентами

KroA150

Статистический анализ 100 экспериментов задачи KroA150 с 3 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.43, -0.11]$ %, в среднем составляя -0.22% ;
- суммарная длина пути агентов изменяется в диапазоне $[-4.91, 3.49]$ %, в среднем уменьшаясь на 1.19% ;
- максимальная длина пути агентов изменяется в диапазоне $[-14.72, 3.74]$ %, при среднем увеличении на 0.26% .

Рисунок 2.17 иллюстрирует примеры работы версии метода, реализующего подхода СМЗК: а) с 2-opt, б) без 2-opt — в сценарии с 3 агентами и задачей на 150 городов.

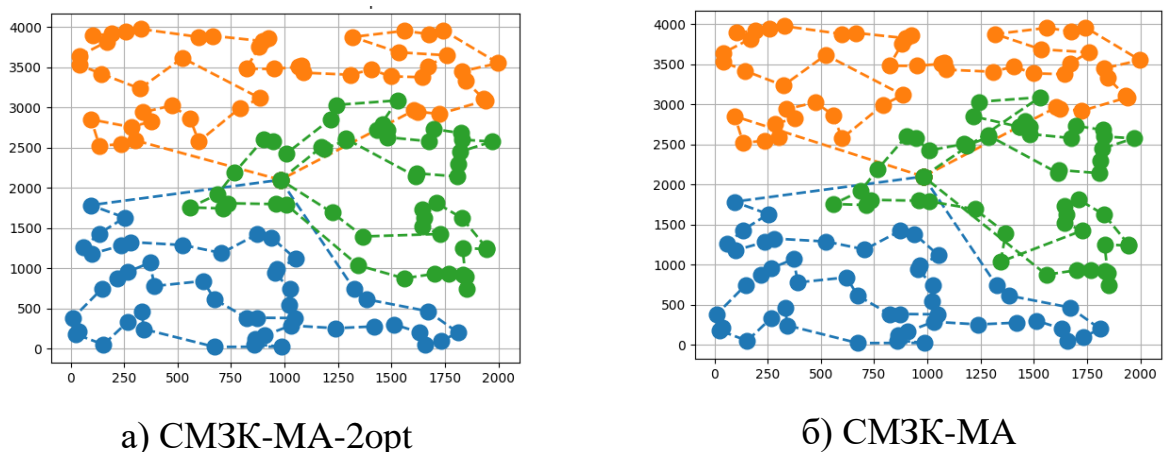


Рисунок 2.17. Примеры решения сценария с 3 агентами и 150 городами

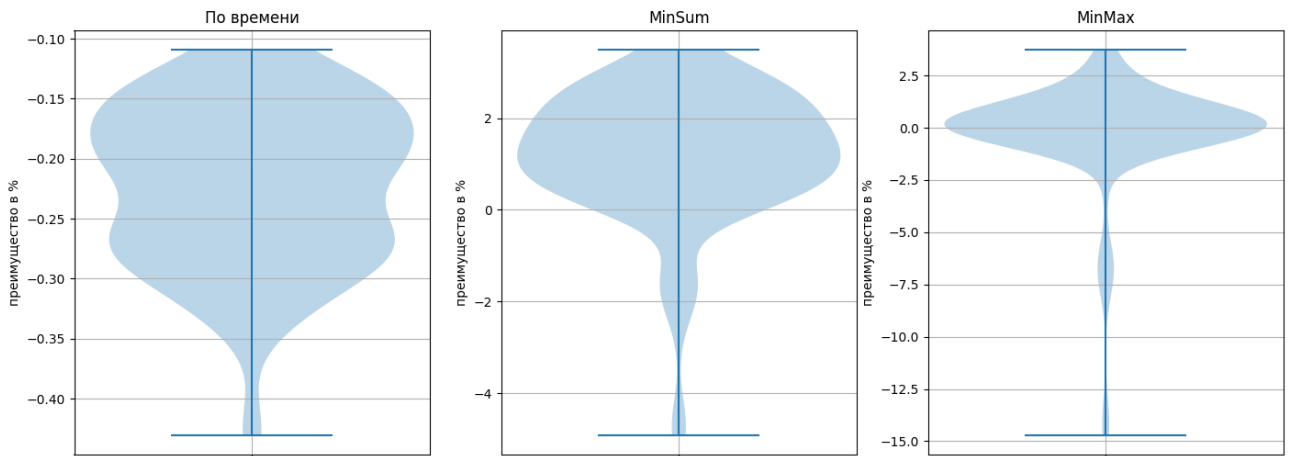


Рисунок 2.18. Скрипичные диаграммы преимуществ метода SM3K-MA-2opt по сравнению с SM3K-MA в задачи KroA150 с 3 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 5 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.42, -0.08]$ %, в среднем составляя -0.24% ;
- суммарная длина пути агентов изменяется в диапазоне $[-4.14, 5.83]$ %, в среднем уменьшаясь на 1.72% ;
- максимальная длина пути агентов изменяется в диапазоне $[-29.13, 5]$ %, при среднем увеличении на 0.79% .

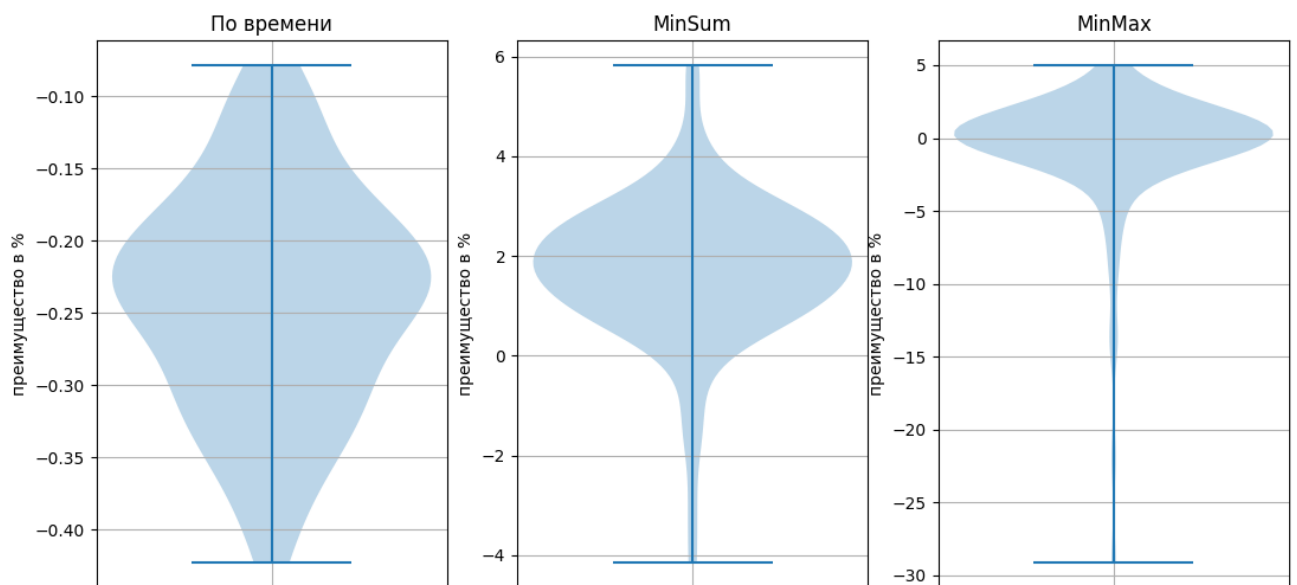


Рисунок 2.19. Скрипичные диаграммы преимуществ метода SM3K-MA-2opt по сравнению с SM3K-MA в задачи KroA150 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 10 агентами показали, что в результате использования предлагаемой модификации:

- время расчёта решения изменяется в диапазоне $[-0.46, 0.22]$ %, в среднем составляя -0.19% ;
- суммарная длина пути агентов изменяется в диапазоне $[-1.78, 5.27]$ %, в среднем уменьшаясь на 1.91% ;
- максимальная длина пути агентов изменяется в диапазоне $[-5.48, 3.85]$ %, при среднем уменьшении на 0.31% .

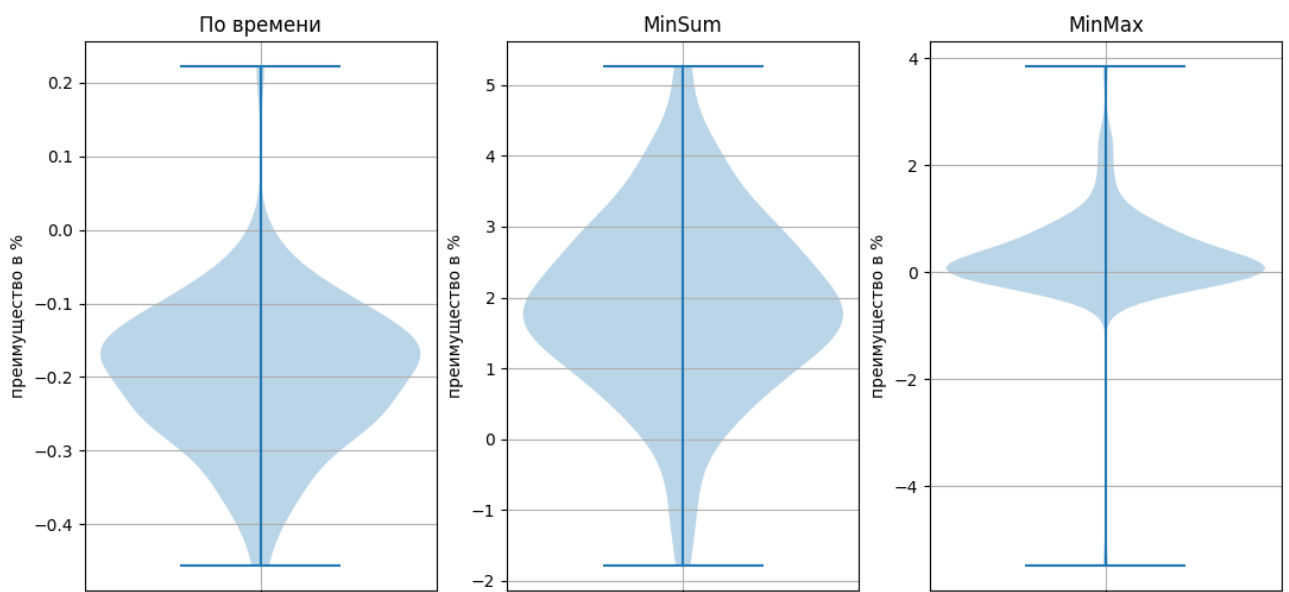


Рисунок 2.20. Скрипичные диаграммы преимуществ метода СМЗК-МА-2орт по сравнению с СМЗК-МА в задачи KroA150 с 10 агентами

Из представленных скрипичных диаграмм можно заметить некоторые случаи, где суммарная длина маршрутов и максимальный по длине маршрут ухудшаются при использовании алгоритма 2-орт. Такие случаи появляются, когда маршрут пересекается с собой несколько раз. Так как алгоритм 2-орт локально обрабатывает эти пересечения, то результирующий маршрут может оказаться длиннее исходного. Однако появления такой ситуации является редким, их доля в выборке составила 9%, следовательно, можно их считать аномалиями.

Рисунок 2.21 представляет возникновения такой ситуации при решении задачи eil51 с пятью агентами. Слева представлен результат решения с помощью

предлагаемого метода без алгоритма 2-opt, фиолетовый маршрут пересекается с самим собой 1 раз, а оранжевый 4 раза, длина этих маршрутов составляет 131.11 и 126.68 единиц измерения соответственно. Справа представлено решение с помощью предлагаемого метода с алгоритмом 2-opt, как можно видеть алгоритм избавился от пересечений маршрутов. Длина фиолетового маршрута составила 127.76 единиц измерения, а длина оранжевого маршрута составила 133.7 единиц измерения.

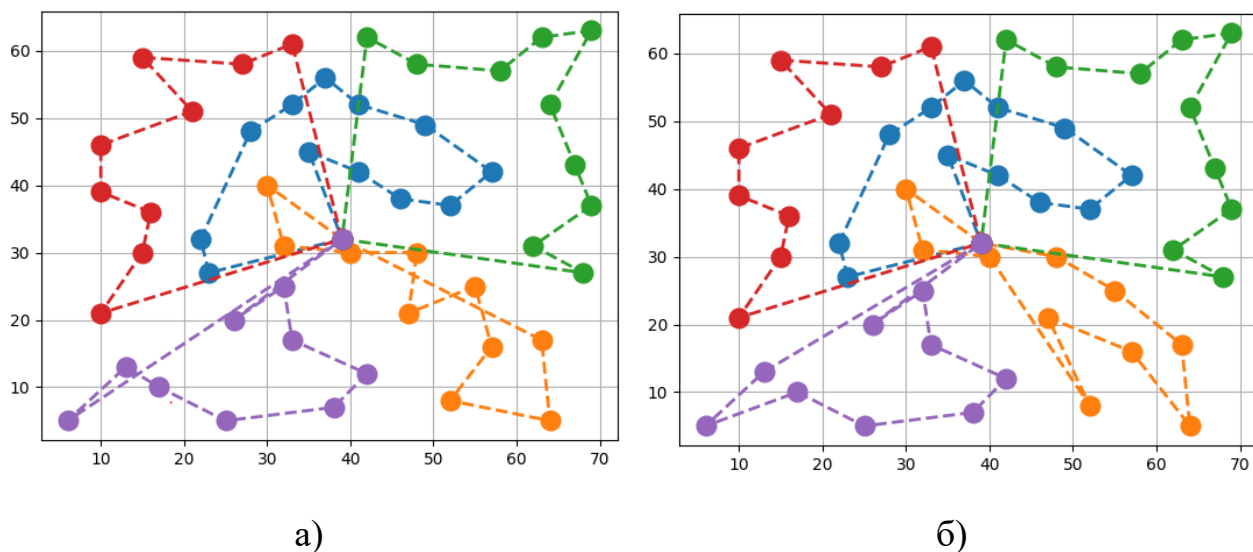


Рисунок 2.21. Результат решения задачи eil51 с пятью агентами методами а) SM3K-MA-2opt, б) SM3K-MA

Подводя итоги, можно отметить, что при использовании дополнительного этапа улучшения качество решения, полученного методом, реализующим подход SM3K, суммарная длина маршрута уменьшается в среднем на 1.3%. При этом время расчета увеличивается в среднем на 0.22%, что делает предлагаемую модификацию актуальной для улучшения качества решения МКВ.

2.4.2. Сравнительное исследование предлагаемого метода с методами, реализующими другие подходы решения многоагентной задачи коммивояжёра

Для проведения сравнительного анализа производительности предлагаемого метода выбирались по одному методу из подходов одновременной оптимизации и сначала-кластеризация, затем маршрутизация.

В числе методов, относящихся к подходу одновременной оптимизации, выделяется метод муравьиного пикета для решения МКВ (ACO-VmTSP) [70], в которой реализован муравьиный алгоритм с существенной модификацией его применения. Вместо использования одного муравья, как в классическом МА, в методе ACO-VmTSP для поиска решения применяется группа муравьев (муравьиный пикет) численностью m , равной количеству агентов задачи.

Первым шагом алгоритма, реализующего данный метод, является инициализация параметров задачи и гиперпараметров муравьиного алгоритма (строка 1:3 алгоритма 2). Все муравьи из пикета располагаются в городе депо $u_j = \{c_d\}$. Множество непосещённых городов S^- включает все города, кроме депо. Каждый муравьиный пикет представляет собой решения МКВ, а каждый муравей j из пикета представляет один агент, хранит маршрут в списке u_j и имеет оценку качества маршрута, определяемой длиной маршрута f_j .

В алгоритме 2 маршруты формируются в цикле следующим образом: муравей пикета, который прошёл минимальное расстояние (строка 5) выбирает город из списка S^- на основе правил МА, указанных в формулах (2.1) и (2.2) (строка 6). Затем он обновляет свой набор посещенных городов (строка 7), увеличивает стоимость маршрута с учётом расстояния d_{eiej} между последним и текущим городами (строка 8), а также обновляет S^- (строка 9). Таким образом, данный метод минимизирует суммарную длину маршрутов и способствует сбалансированному распределению нагрузок между агентами. Распространение феромонов происходит аналогично оригинальному муравьиному алгоритму (формула 10). Затем шаги 5–9 повторяются до тех пор, пока множество S^- не станет пустым. Каждое решение, сформированное на итерации муравьиного пикета, оценивается с помощью целевой функции.

Алгоритм 2 алгоритм ACO-BmTSP

```
1   $f_j = 0, \forall j \in \{1, \dots, m\}$ 
2   $u_j = \{c_d\}, \forall j \in \{1, \dots, m\}$ 
3   $C^- = \{1, \dots, n\} \setminus \{c_d\}$ 
4  While  $C^- \neq \emptyset$  do
5      Найти муравья с минимальным значением функции пригодности
6      Выбрать  $c_i \in C^-$  по правилам отбора
7       $u_j = u_j \cup c_i$ 
8       $f_j = f_j + d_{eiej}$ 
9       $C^- = C^- \setminus \{c_j\}$ 
10 end while
11 Вернуть решения
```

Представителем подхода «Сначала кластеризация, затем маршрутизация» (СКЗМ) является работа Латаха [57] (KM-CACO), в которой города группируются в кластеры, число которых соответствует количеству агентов задачи, с использованием алгоритма K-Means. После этого в каждом кластере применяется муравьиный алгоритм для поиска оптимального маршрута.

Для решения МКВ с использованием кластеризации применяются следующие этапы:

1. Создается C^- — список координат городов, за исключением города депо $C = C^- \cup \{c_d\}$.
2. Список C^- передаётся алгоритму кластеризации, который возвращает принадлежность каждого города из списка C^- к определённому кластеру (соответствующему одному из агентов).
3. В каждом кластере добавляем город депо c_d .

4. Для каждого кластера применяется муравьиный алгоритм, который упорядочивает города внутри кластера, формируя маршрут для соответствующего агента.

Рисунок 2.22 иллюстрирует этапы решения МКВ с тремя агентами с использованием метода КМ-АСО.

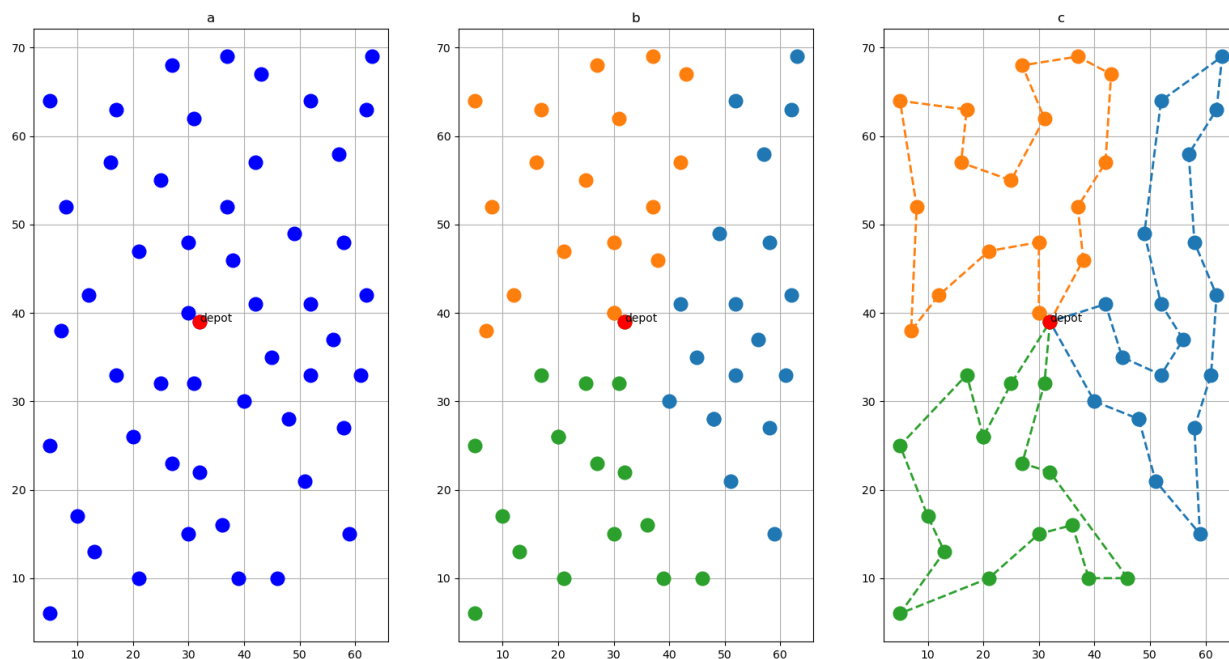


Рисунок 2.22. Этапы решения МКВ с использованием метода КМ-АСО: а) синими точками обозначены города из списка C^- , красной – город депо; б) разделения списка C^- на 3 кластера, соответствующих числу коммивояжеров; в) применение муравьиного алгоритма в каждом кластере, чтобы определить порядок посещения городов коммивояжерами

Для проведения сравнительного анализа вышеуказанных методов с предлагаемым методом были выбраны три задачи eil51, kroA100 и kroA150. Для каждой задачи были заданы три сценария с количеством агентов 3, 5 и 10. Каждый из этих сценариев решался по 100 раз каждым из рассматриваемых методов [71].

Критерии оценки эффективности изучаемых методов остаются теми же, что и в параграфе 2.4.1 (время расчета, MinSum, MinMax).

Таблица 2.4 содержит значения гиперпараметров муравьиного алгоритма, использованных в методах АСО-ВМТSP, КМ-АСО и в предлагаемом методе, где n – количество городов задачи.

Количество муравьёв и итераций в методе АСО-ВМТSP выбрано на основе рекомендаций авторов метода [77] и предварительного тестирования на подмножестве задач. В методах КМ-САСО и в предлагаемом методе количество муравьёв выбрано на основе предварительного тестирования на подмножестве задач [86]. При этом число итераций явно не задавалось: в качестве критерия остановки муравьиного алгоритма использовалось 20 последовательных итераций без улучшения найденного решения.

Таблица 2.4. Значения гиперпараметров муравьиного алгоритма.

Гиперпараметр	АСО-ВМТSP	КМ-САСО	СМЗК-МА-2opt
Количество муравьёв (K)	$2n$	n	n
Число итераций (itr)	$0.5n$	до стагнации	до стагнации
Критерий остановки	Фиксированное количество итерации	20 итерации без улучшения качества решения	20 итерации без улучшения качества решения
Феромонное испарение (ρ)	0.5	0.5	0.5

Исследуемые методы реализованы на языке Python 3.8. Моделирование проводилось на персональном компьютере с процессором Intel Core i7-9700KF CPU @ 3.60GHz, с оперативной памятью 32GB, под управлением операционной системы Windows 10 Pro 64-bit.

Ниже представлены результаты статистического анализа 100 экспериментов для каждого из сценариев в виде преимущества использования предлагаемого метода по сравнению с методами АСО-ВМТSP и КМ-САСО.

На рисунках 2.23-2.31 представлены скрипичные диаграммы, отражающие процентное преимущество предлагаемого метода по критериям оценки качества решения по сравнению с методами АСО-ВМТSP и КМ-САСО, реализующие

подходы одновременной оптимизации и СКЗМ последовательно. Ширина диаграммы отражает количество экспериментов с соответствующим значением преимущества. Отрицательное значение преимущества означает, что в некоторых случаях предлагаемый метод не улучшает, а, напротив, ухудшает результат по сравнению с остальными методами [68].

Eil51

Статистический анализ 100 экспериментов задачи eil51 с 3 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с АСО-ВМТСП:
 - время расчёта решения изменяется в диапазоне [-145.85, 35.75] %, в среднем составляя -5.30%;
 - суммарная длина маршрута изменяется в диапазоне [3.47, 21.35] %, в среднем составляя 14.07%;
 - максимальная длина среди маршрутов изменяется в диапазоне [4.25, 26.52] %, в среднем составляя 16.29%;
- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [-1473, -181.73] %, в среднем составляя -719.31%;
 - суммарная длина маршрута изменяется в диапазоне [-10.45, 7.51] %, в среднем составляя 0.13%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-5.11, 26.72] %, в среднем составляя 8.38%;

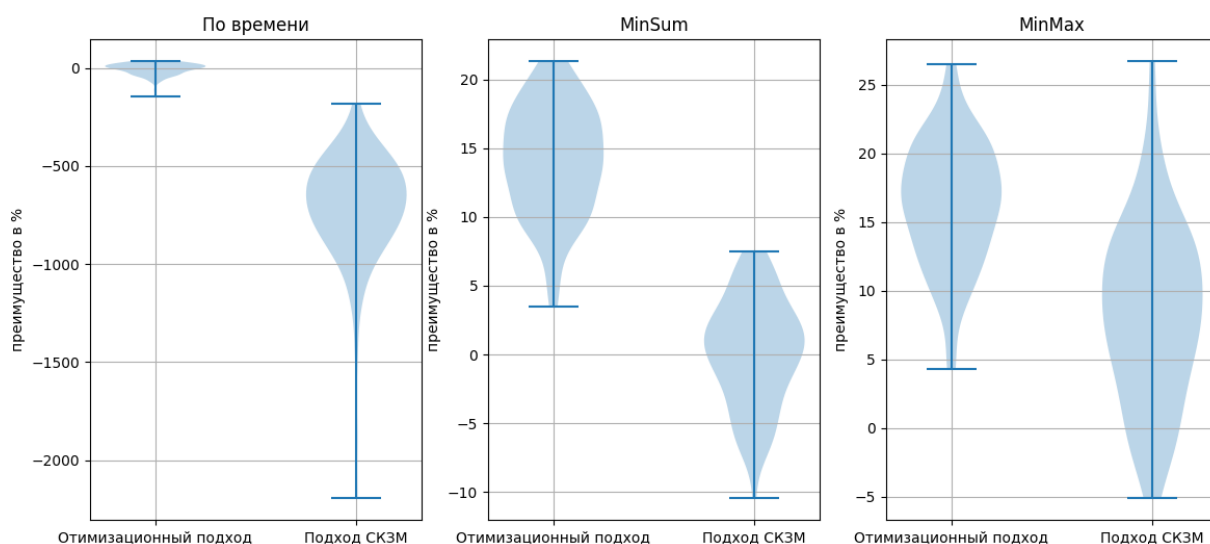


Рисунок 2.23. Скрипичные диаграммы преимуществ предлагаемого метода в задаче eil51 с 3 агентами

Статистический анализ 100 экспериментов задачи eil51 с 5 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с ACO-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-88.64, 32.95]$ %, в среднем составляя -5.19% ;
 - суммарная длина маршрута изменяется в диапазоне $[0.39, 21.04]$ %, в среднем составляя 12.80% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-14.90, 33.13]$ %, в среднем составляя 16.95% ;
- по сравнению с KM-CACO:
 - время расчёта решения изменяется в диапазоне $[-836.27, -186.98]$ %, в среднем составляя -465.98% ;
 - суммарная длина маршрута изменяется в диапазоне $[-21.04, 1.22]$ %, в среднем составляя -6.26% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-34.31, 21.34]$ %, в среднем составляя 4.24% ;

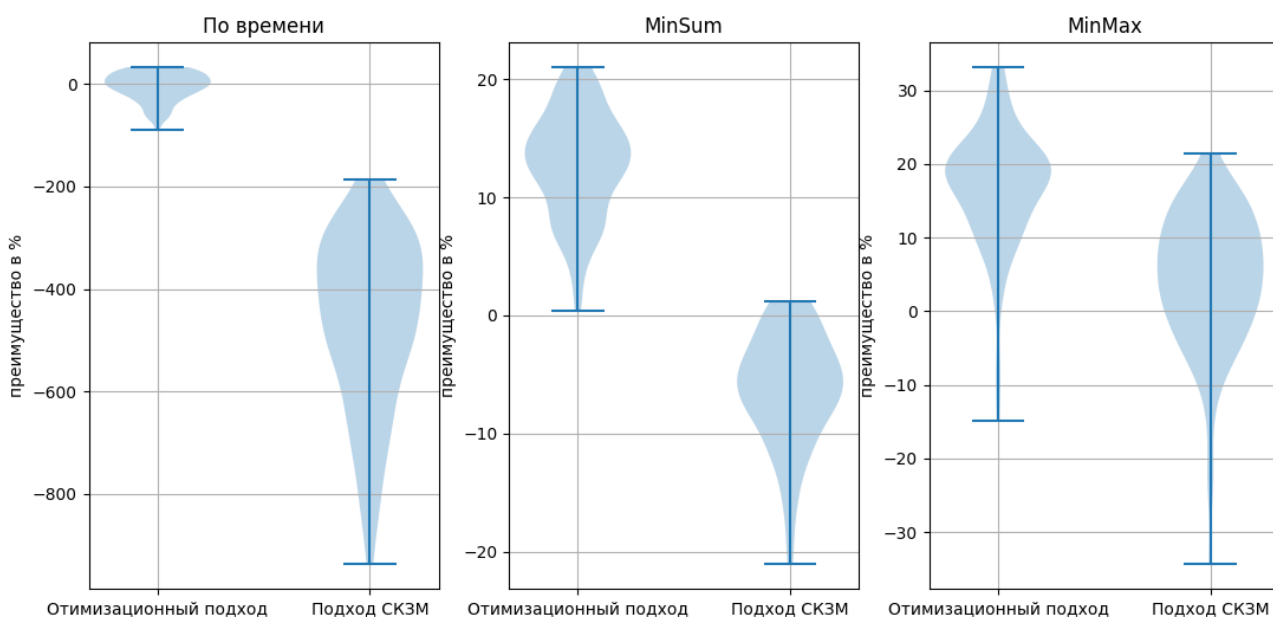


Рисунок 2.24. Скрипичные диаграммы преимуществ предлагаемого метода в задаче eil51 с 5 агентами

Статистический анализ 100 экспериментов задачи eil51 с 10 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с АСО-BMTSP:
 - время расчёта решения изменяется в диапазоне [-160.83, -54.55] %, в среднем составляя -91.68%;
 - суммарная длина маршрута изменяется в диапазоне [1.19, 13.92] %, в среднем составляя 8.50%;
 - максимальная длина среди маршрутов изменяется в диапазоне [1.61, 38.01] %, в среднем составляя 17.00%;
- по сравнению с подходом СКЗМ:
 - время расчёта решения изменяется в диапазоне [-794.04, -330.74] %, в среднем составляя -528.96%;
 - суммарная длина маршрута изменяется в диапазоне [-15.72, 0.49] %, в среднем составляя -7.57%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-4.10, 20.29] %, в среднем составляя 6.38%;

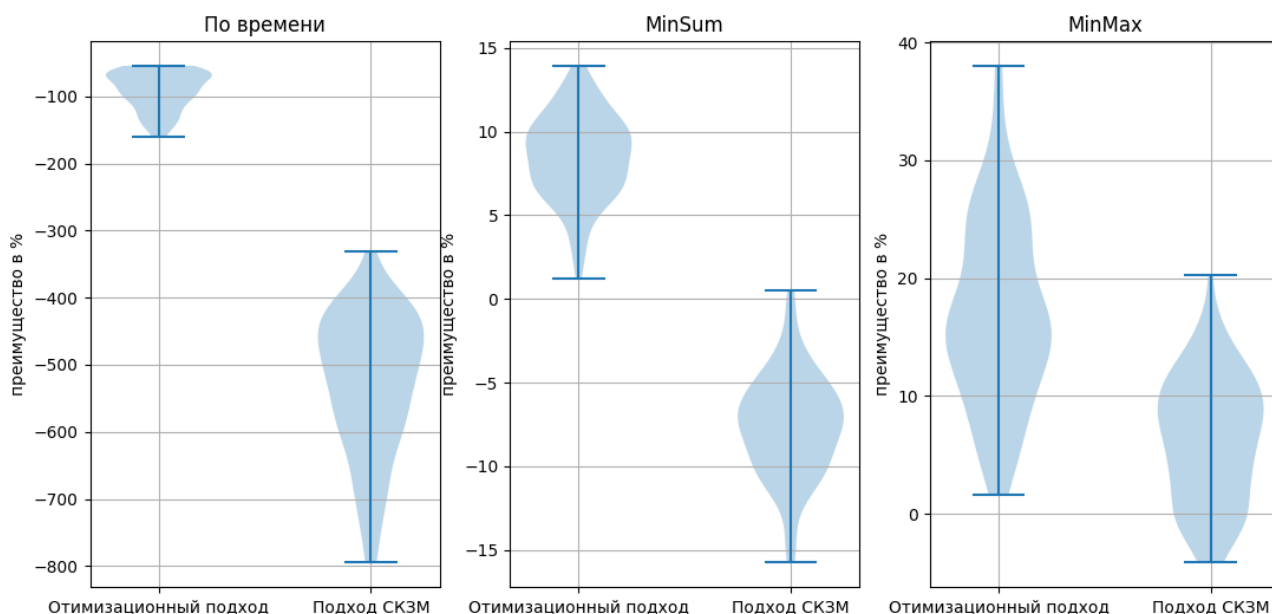


Рисунок 2.25. Скрипичные диаграммы преимуществ предлагаемого метода в задачи eil51 с 10 агентами

KroA100

Статистический анализ 100 экспериментов задачи KroA100 с 3 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с АСО-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-115.02, 39.48]$ %, в среднем составляя 6.26%;
 - суммарная длина маршрута изменяется в диапазоне $[5.98, 22.22]$ %, в среднем составляя 17.46%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[6.40, 37.01]$ %, в среднем составляя 19.55%;
- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне $[-3743.32, -783.67]$ %, в среднем составляя -1829.03%;
 - суммарная длина маршрута изменяется в диапазоне $[-3.98, 12.10]$ %, в среднем составляя 4.10%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[10.73, 34.91]$ %, в среднем составляя 22.86%;

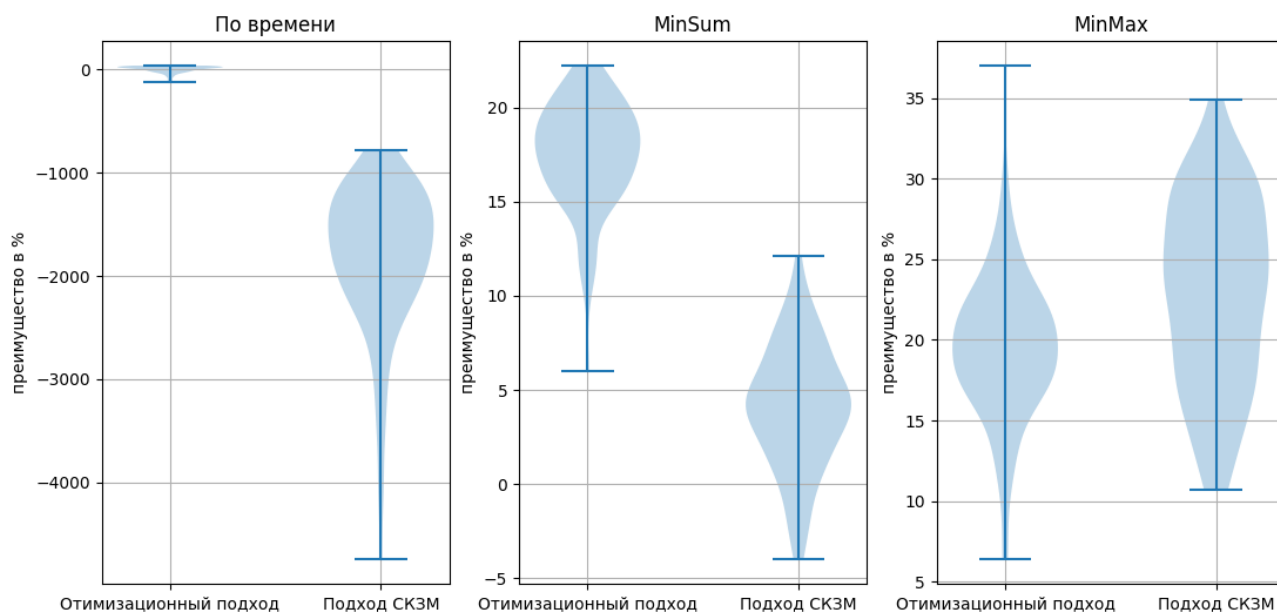


Рисунок 2.26. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA100 с 3 агентами

Статистический анализ 100 экспериментов задачи KroA100 с 5 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с ACO-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-89.96, 41.85]$ %, в среднем составляя 14.23%;
 - суммарная длина маршрута изменяется в диапазоне $[6.22, 21.36]$ %, в среднем составляя 15.21%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-6.63, 30.75]$ %, в среднем составляя 16.28%;
- по сравнению с KM-CACO:
 - время расчёта решения изменяется в диапазоне $[-1419.41, -531.30]$ %, в среднем составляя -1051.01%;
 - суммарная длина маршрута изменяется в диапазоне $[-14.57, 2.60]$ %, в среднем составляя -5.61%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-22.21, 34.44]$ %, в среднем составляя 9.10%;

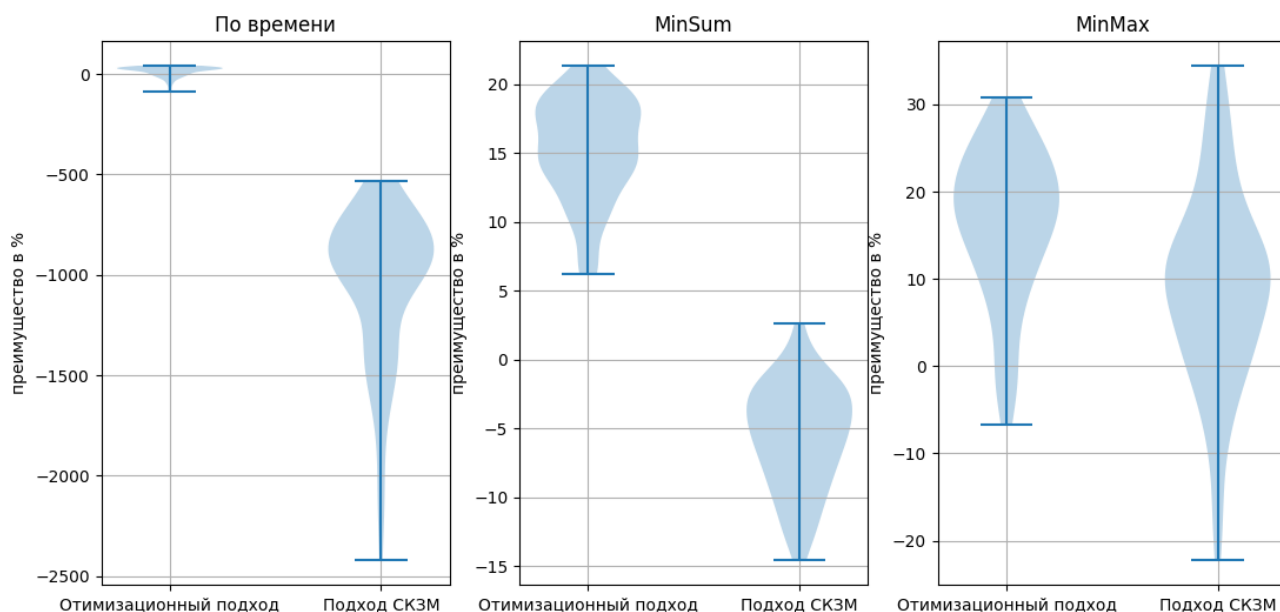


Рисунок 2.27. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA100 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA100 с 10 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с ACO-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-99.73, 21.06]$ %, в среднем составляя -5.12% ;
 - суммарная длина маршрута изменяется в диапазоне $[9.51, 21.47]$ %, в среднем составляя 15.35% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-13.33, 32.84]$ %, в среднем составляя 14.13% ;
- по сравнению с KM-CACO:
 - время расчёта решения изменяется в диапазоне $[-1220.96, -434.33]$ %, в среднем составляя -713.89% ;
 - суммарная длина маршрута изменяется в диапазоне $[-17.13, -0.18]$ %, в среднем составляя -9.92% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-30.57, 20.36]$ %, в среднем составляя 2.04% ;

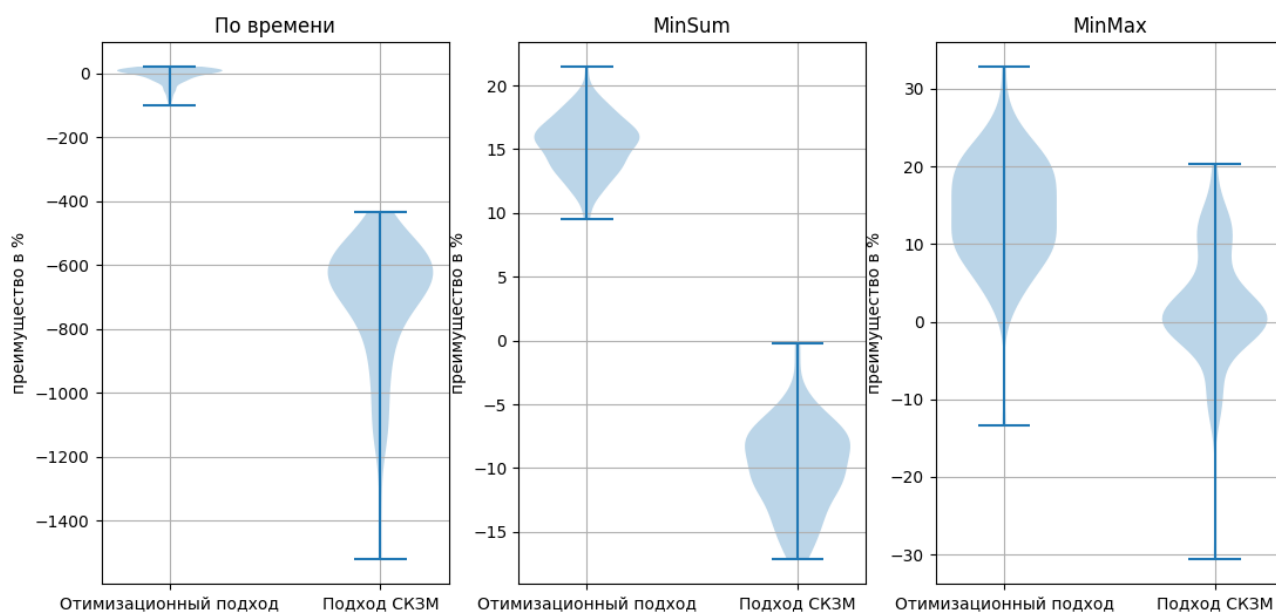


Рисунок 2.28. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA100 с 10 агентами

KroA150

Статистический анализ 100 экспериментов задачи KroA150 с 3 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с ACO-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-33.44, 47.59]$ %, в среднем составляя 22.58%;
 - суммарная длина маршрута изменяется в диапазоне $[8.30, 27.53]$ %, в среднем составляя 20.99%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-3.86, 32.63]$ %, в среднем составляя 22.12%;
- по сравнению с KM-CACO:
 - время расчёта решения изменяется в диапазоне $[-4104.54, -1395.14]$ %, в среднем составляя -2630.27%;
 - суммарная длина маршрута изменяется в диапазоне $[-6.54, 16.58]$ %, в среднем составляя 6.94%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[8.72, 38.39]$ %, в среднем составляя 27.19%;

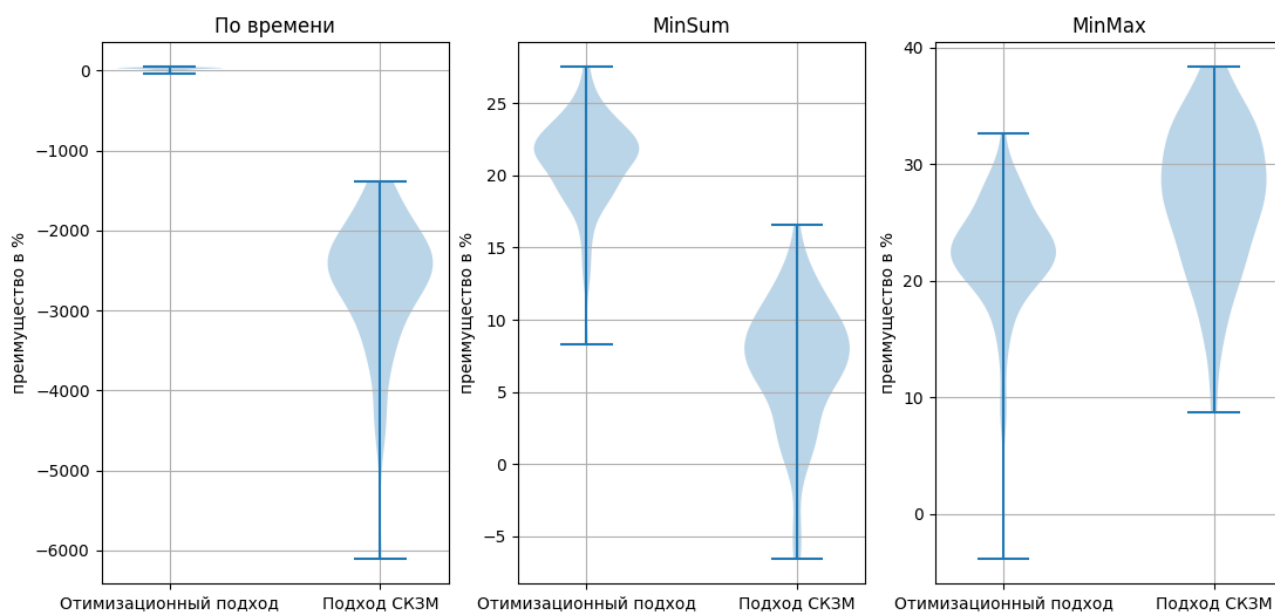


Рисунок 2.29. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA150 с 3 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 5 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с АСО-BMTSP:
 - время расчёта решения изменяется в диапазоне $[-64.78, 47.31]$ %, в среднем составляя 25.41%;
 - суммарная длина маршрута изменяется в диапазоне $[11.42, 28.68]$ %, в среднем составляя 20.06%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-10.60, 33.48]$ %, в среднем составляя 20.60%;
- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне $[-2679.43, -845.49]$ %, в среднем составляя -1723.83%;
 - суммарная длина маршрута изменяется в диапазоне $[-10.58, 7.73]$ %, в среднем составляя -2.69%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-24.46, 35.57]$ %, в среднем составляя 10.27%;

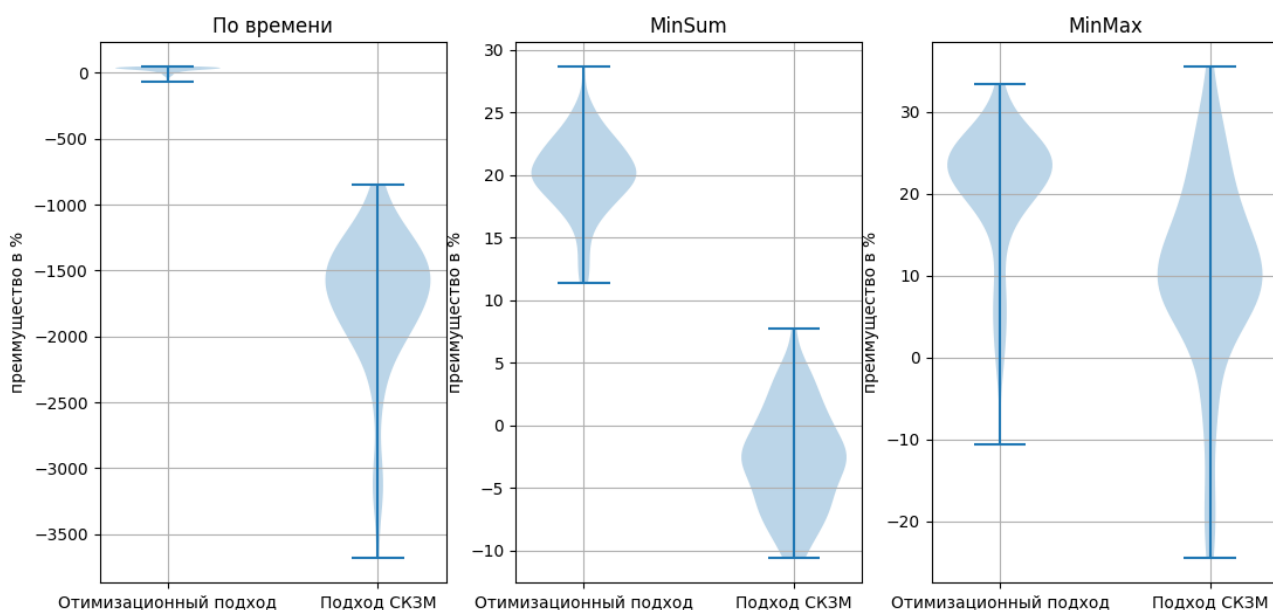


Рисунок 2.30. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA150 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 10 агентами показали, что в результате использования предлагаемой модификации:

- по сравнению с АСО-ВМТСП:
 - время расчёта решения изменяется в диапазоне $[-79.75, 39.41]$ %, в среднем составляя 17.69%;
 - суммарная длина маршрута изменяется в диапазоне $[12.08, 23.87]$ %, в среднем составляя 17.79%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-15.80, 36.74]$ %, в среднем составляя 18.33%;
- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне $[-1471.20, -614.64]$ %, в среднем составляя -981.48%;
 - суммарная длина маршрута изменяется в диапазоне $[-16.39, -2.05]$ %, в среднем составляя -8.46%;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-23.09, 22.37]$ %, в среднем составляя 6.36%;

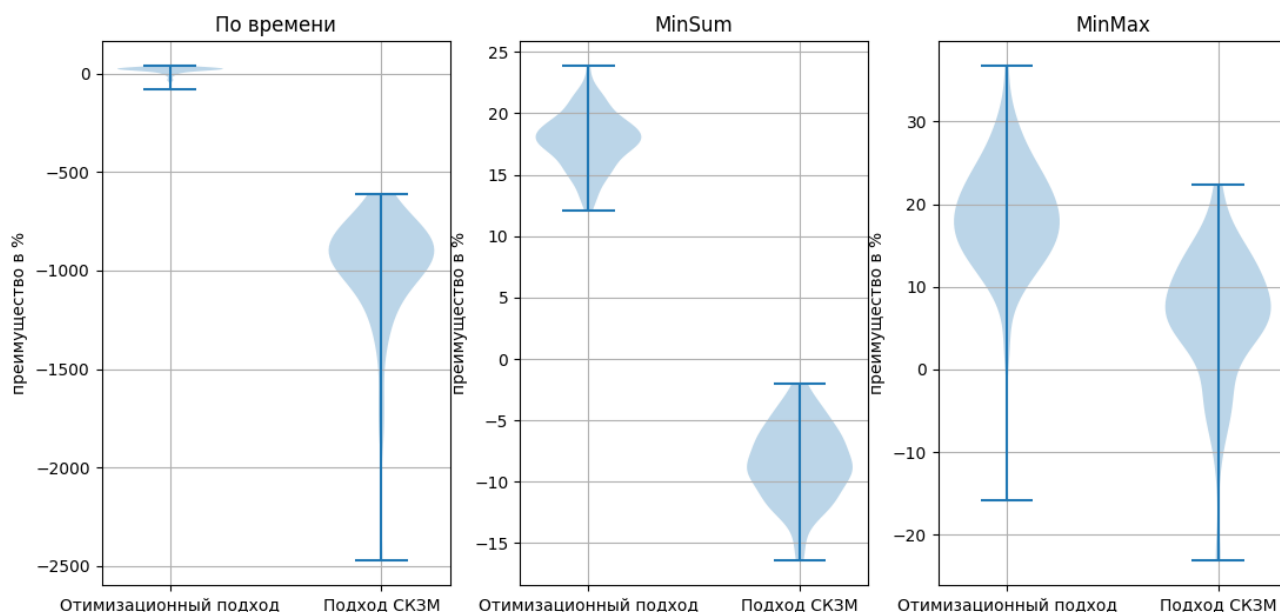


Рисунок 2.31. Скрипичные диаграммы преимуществ предлагаемого метода в задаче KroA150 с 10 агентами

Таблицы 2.5, 2.6 и 2.7 последовательно представляют результаты решения задач eil51, kroA100 и kroA150 с 3, 5 и 10 агентами (m). В таблицах приведены среднее значения по 100 запускам: время расчета решения в секундах (μ_t), сумма длин маршрутов в метрах (μ_{sum}) и максимальная длина маршрута среди коммивояжеров в метрах (μ_{max}) [72]. Соответствующие наилучшие значения выделены жирным шрифтом.

Таблица 2.5. Результаты решения задачи eil51.

m	3			5			10		
Метод	ACO-BMTP	KM-CAC	CM3K-MA-2opt	ACO-BMTP	KM-CAC	CM3K-MA-2opt	ACO-BMTP	KM-CAC	CM3K-MA-2opt
μ_t	3.99	0.519	4.5	4.06	0.76	4.27	4.3	1.31	8.24
μ_{sum}	597.9	514.18	513.43	683.5	560.6	595.4	904.7	769.5	827.5
μ_{max}	211.5	190.87	177.43	153.6	133.33	127.08	113.2	99.78	93.16

Таблица 2.6. Результаты исследования задачи kroA100.

m	3			5			10		
Me- тод	АСО- BMTS P	KM- CAC O	CMЗК -МА- 2opt	АСО- BMTS P	KM- CAC O	CMЗК -МА- 2opt	АСО- BMTS P	KM- CAC O	CMЗК -МА- 2opt
μ_t	43.44	2.11	37.52	44.86	3.41	38.47	46.97	6.1	49.38
μ_{sum}	32533	28054	26855	38205	30658	32360	52462	40412	44391
μ_{max}	11500	11775	9233	8542	7916	7111	6339	5550	5417

Таблица 2.7. Результаты исследования задачи kroA150.

m	3			5			10		
Me- тод	АСО- BMTS P	KM- CAC O	CMЗК -МА- 2opt	АСО- BMT SP	KM- CAC O	CMЗК -МА- 2opt	АСО- BMT SP	KM- CAC O	CMЗК -МА- 2opt
μ_t	196.2	5.87	152.6	203	8.41	151.4	197.7	15.21	162.8
μ_{sum}	40808	34671	32006	46489	36192	37146	60040	45507	49344
μ_{max}	14219	15355	10981	10099	8998	8002	7074	6169	5749

Анализ полученных результатов, показывает, что предлагаемый метод превосходит АСО-BMTSP по всем критериям оценки. В частности, время расчета при использовании предлагаемого метода снижается в среднем на 9–54.5 % по сравнению с АСО-BMTSP. При этом суммарная длина маршрутов сокращается в среднем на 8.9 – 19.2 %, а максимальная длина маршрута среди всех коммивояжеров — на 12.4 – 20.2 %.

В сравнении с методом KM-CACO предлагаемый метод проигрывает по времени расчета и суммарной длине маршрутов. Однако стоит отметить, что он превосходит KM-CACO по критерию минимизации максимальной длины маршрута среди агентов. При этом время расчёта увеличивается в среднем на 561-867% для задачи eil51, на 809-1777% для задачи kroA100 и на 1070-2589% для

задачи kroA150, а суммарная длина маршрутов — в среднем на 2–9 % для сценариев с 5 и 10 агентами. В то же время, для сценариев с 3 агентами суммарная длина маршрутов уменьшается в среднем на 0.1–7 %, а максимальная длина маршрута среди агентов — на 2–28.4 % [73].

2.5. Выводы по главе 2

1- Модифицирован метод решения многоагентной задачи коммивояжёра, основанный на подходе СМЗК, что позволило повысить качество решения по критерию MinSum. Модификация заключается в добавлении дополнительного этапа улучшения найденного решения с помощью алгоритма локального поиска 2-opt.

2- Проведено сравнение эффективности данного метода с исходным методом (без модификации), выявлено, что предлагаемая модификация улучшает качество решения по критерию MinSum в среднем на 1.3% за счет увеличения время расчета в среднем на 0.22%.

3- Проведено сравнение эффективности данного метода с методами, реализующими подходы одновременной оптимизации и сначала кластеризация, затем маршрутизация. Установлено, что предлагаемый метод обладает более гибким механизмом оптимизации по критерию MinMax.

4- Метод СМЗК имеет более длительное время расчёта по сравнению с методом, основанным на подходе сначала кластеризация, затем маршрутизация, который достигает высокой скоростью за счёт уменьшения размера пространства решений.

Глава 3. Разработка гибридного метода решения многоагентной задачи коммивояжёра

В предыдущей главе показано, что подход СМЗК с предложенной модификацией демонстрирует наилучшие результаты по критерию минимизации максимальной длины маршрута, что приводит к более равномерному распределению нагрузки между агентами. Однако, данный подход имеет существенный недостаток – высокую вычислительную сложность по сравнению с подходом «Сначала кластеризация, затем маршрутизация» [75]. Это обусловлено тем, что в подходе СМЗК алгоритм маршрутизации ищет решение в пространстве размером $(n-1)!$, где n – количество городов МКВ, что превышает размер пространства решений при использовании подхода СКЗМ в его худшем случае — $(n-m)!$, где m – количество городов МКВ, который достигается, когда $m-1$ кластеров содержат по одному городу, а один кластер содержит оставшиеся $n-m$ городов, и значительно превышает размер пространства решений при равномерном распределении n задач между m агентами — $m \left(\frac{n-1}{m} \right)!$

Чем больше размер пространства решений, тем ниже вероятность того, что муравьиный алгоритм найдёт оптимальное решение, и тем выше риск попадания в локальный минимум. В результате методы, реализующие подход СМЗК требуют значительно больше времени для расчета решения. В связи с этим предлагается гибридный метод, сочетающий элементы обоих подходов СМЗК и СКЗМ – с целью уменьшения размера пространства решений МКВ и сохранить механизм оптимизации критерия MinMax [74].

3.1. Предлагаемый метод «Кластеризация, Маршрутизация, Соединение, Разделение»

Предлагается гибридный метод решения многоагентной задачи коммивояжёра, объединяющий сильные стороны двух ранее рассмотренных подходов:

СКЗМ и СМЗК. Метод получил название «Кластеризация, Маршрутизация, Соединение, Разделение» (КМСР), поскольку алгоритмическая его реализация включает четыре последовательных этапа:

- **кластеризация:** задачи (города) делятся на кластеры. Цель этапа — распределить города таким образом, чтобы минимизировать размер пространства решений и упростить последующую маршрутизацию;
- **маршрутизация:** для каждого кластера строится оптимальный маршрут с использованием эвристик, таких как муравьиный алгоритм;
- **соединение:** все кластерные маршруты объединяются в единый маршрут (супер-маршрут), охватывающий все города задачи;
- **разделение:** полученный супер-маршрут разбивается на подмаршруты, каждый из которых начинается и заканчивается в городе депо и закрепляется за конкретным агентом. При этом учитываются такие ограничения, как максимальная длина маршрута, балансировка нагрузки и ограничение времени выполнения.

Суть предлагаемого метода заключается в объединении преимуществ подходов СКЗМ и СМЗК: с одной стороны – в возможности разбиения задачи на подзадачи, что уменьшает размер пространства решений и снижает время расчёта (СКЗМ); с другой – в способности минимизировать максимальную длину маршрута среди всех агентов (СМЗК) [74].

Для реализации предлагаемого метода необходимо сначала ответить на несколько ключевых вопросов:

1. На сколько кластеров следует разделить города на первом этапе?
2. Как соединить кластеры между собой для формирования супер-маршрута? (Какие города внутри кластера должны выступать в роли узлов соединения с другими кластерами?)
3. Как следует решать задачу маршрутизации внутри каждого кластера с учётом того, что эти маршруты будут объединены с маршрутами других кластеров?

В следующих разделах будут подробно рассмотрены эти вопросы и предложены соответствующие решения [75].

3.1.1. Определение количества кластеров

В подходе СКЗМ количество кластеров является фиксируемым и всегда равным количеству агентов задачи. В то время как в предлагаемом методе КМСР число кластеров может быть каким угодно (не превышающим количества городов задачи) и не обязательно должно совпадать с количеством агентов.

Чем больше количество кластеров, на которые разделяются города задачи, тем меньше число городов в каждом кластере, следовательно, тем меньше временные затраты на поиск оптимального маршрута внутри кластера. Однако, с другой стороны, по мере увеличения числа кластеров результирующий супермаршрут всё дальше отстоит от оптимального и всё ближе приближается к результату жадной эвристики – особенно в случае, когда в каждом кластере остаётся не более двух городов.

Для исследования зависимости времени вычислений, с одной стороны, и качества решения по критериям общей длины маршрутов и максимальной длины маршрута, с другой стороны, от количества кластеров и числа агентов, были использованы три задачи *eil51*, *kroA100* и *kroA150*. Для каждой из этих задач были заданы три сценария с 3, 5 и 10 агентами. Каждый сценарий запускался с изменением количества кластеров метода решения, изменяющимся от 2 до 10, при этом каждая комбинация (задача, количество городов, количество кластеров) запускалась по 100 раз для получения репрезентативной выборки.

На рисунках 3.1, 3.2 и 3.3 представлены результаты исследования для задач *eil51*, *kroA100* и *kroA150* соответственно. Каждый график соответствует определённому количеству агентов. По горизонтальной оси откладывается количество кластеров, по вертикальной оси — значения суммарной длины маршрутов (красная линия), максимальной длины маршрута (зелёная линия) и времени расчёта (синяя линия). Как видно из графиков, с увеличением количества кластеров

время расчёта снижается, но качество решения падает. При этом хорошим приближением к компромиссу между временем расчёта и качеством решения является использование 4 кластеров.

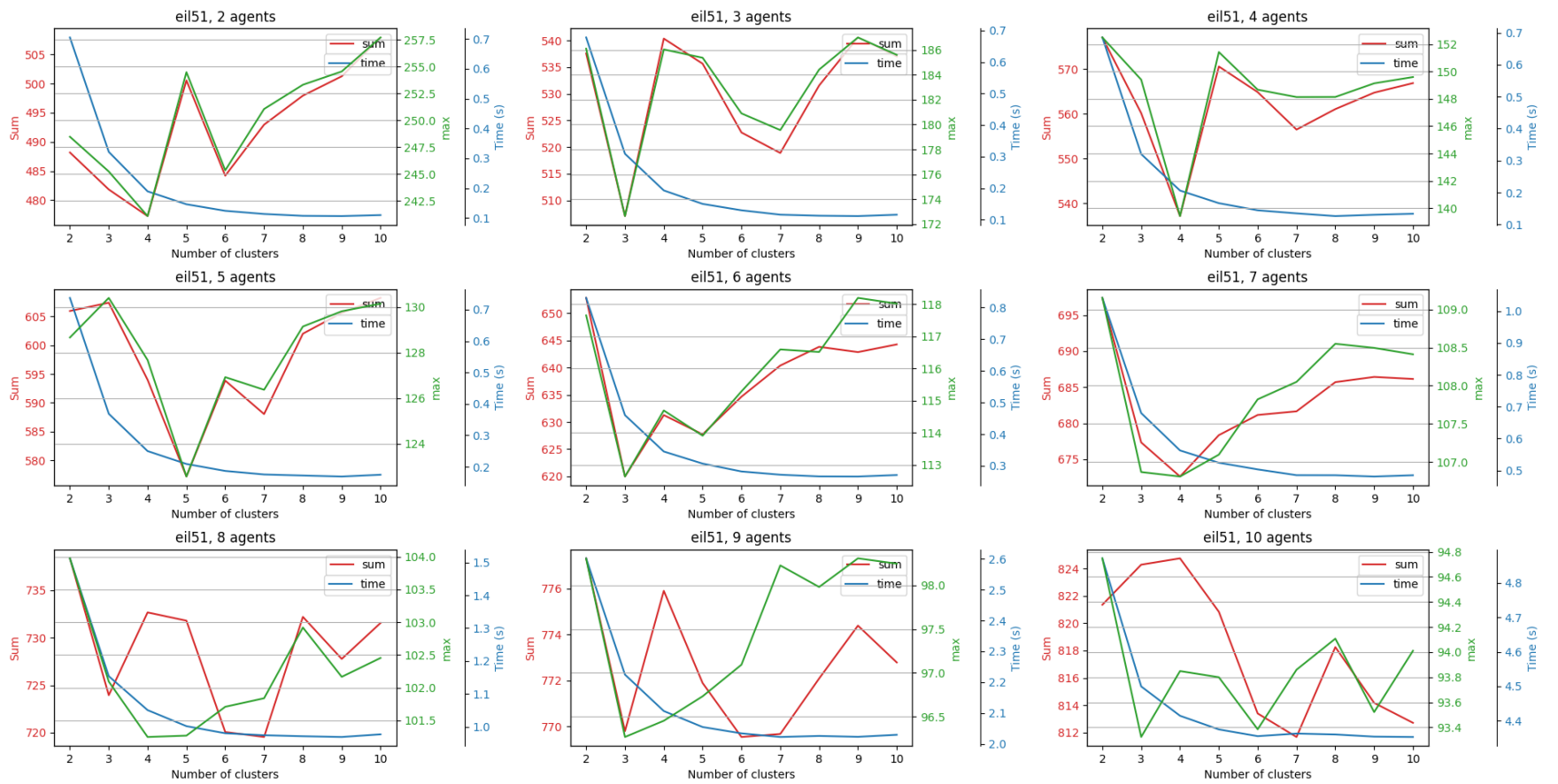


Рисунок 3.1. Влияние количества кластеров на время расчёта и качество решения при изменении количества агентов для задачи eil51

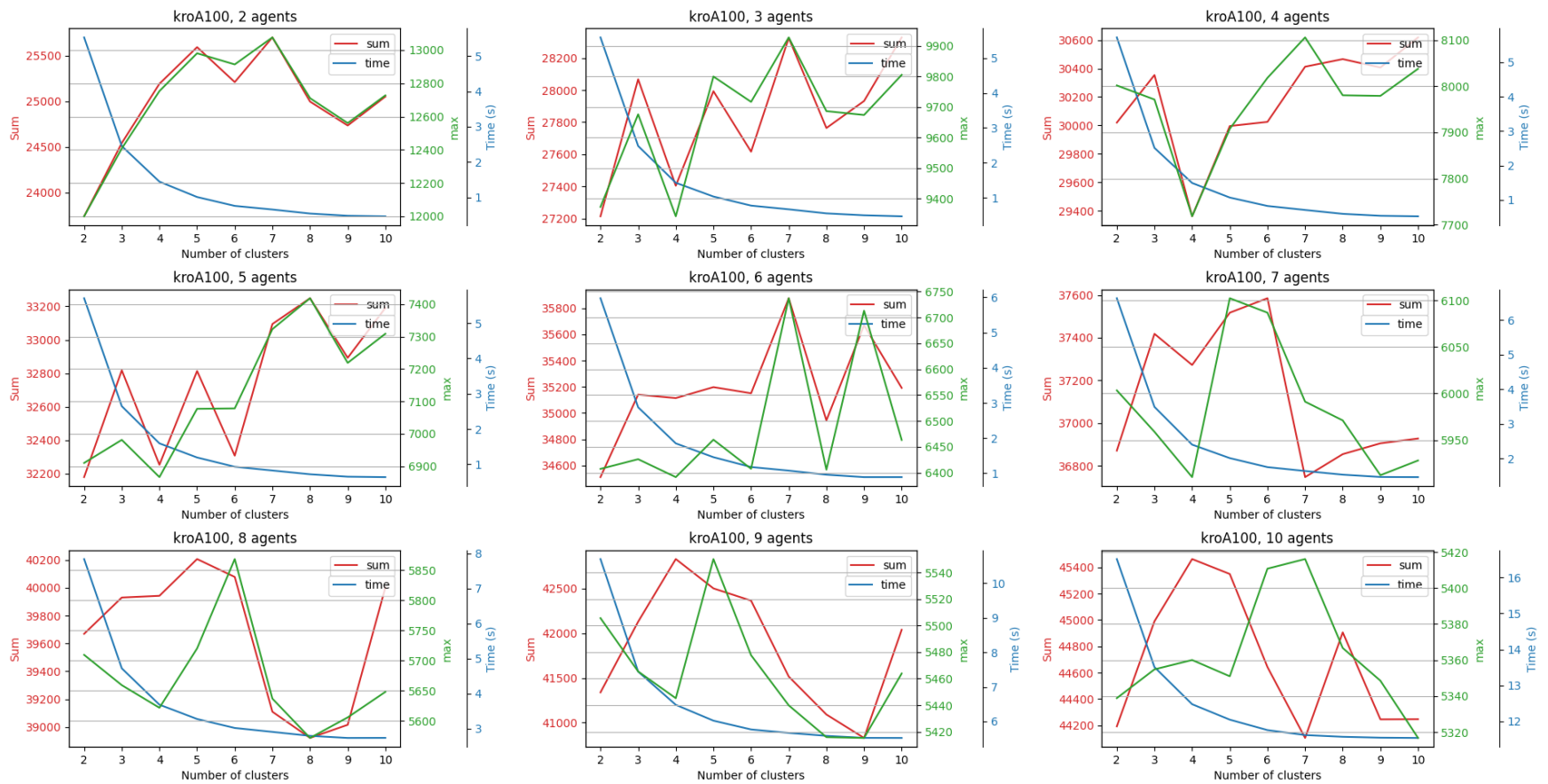


Рисунок 3.2. Влияние количества кластеров на время расчёта и качество решения при изменении количества агентов для задачи KroA100

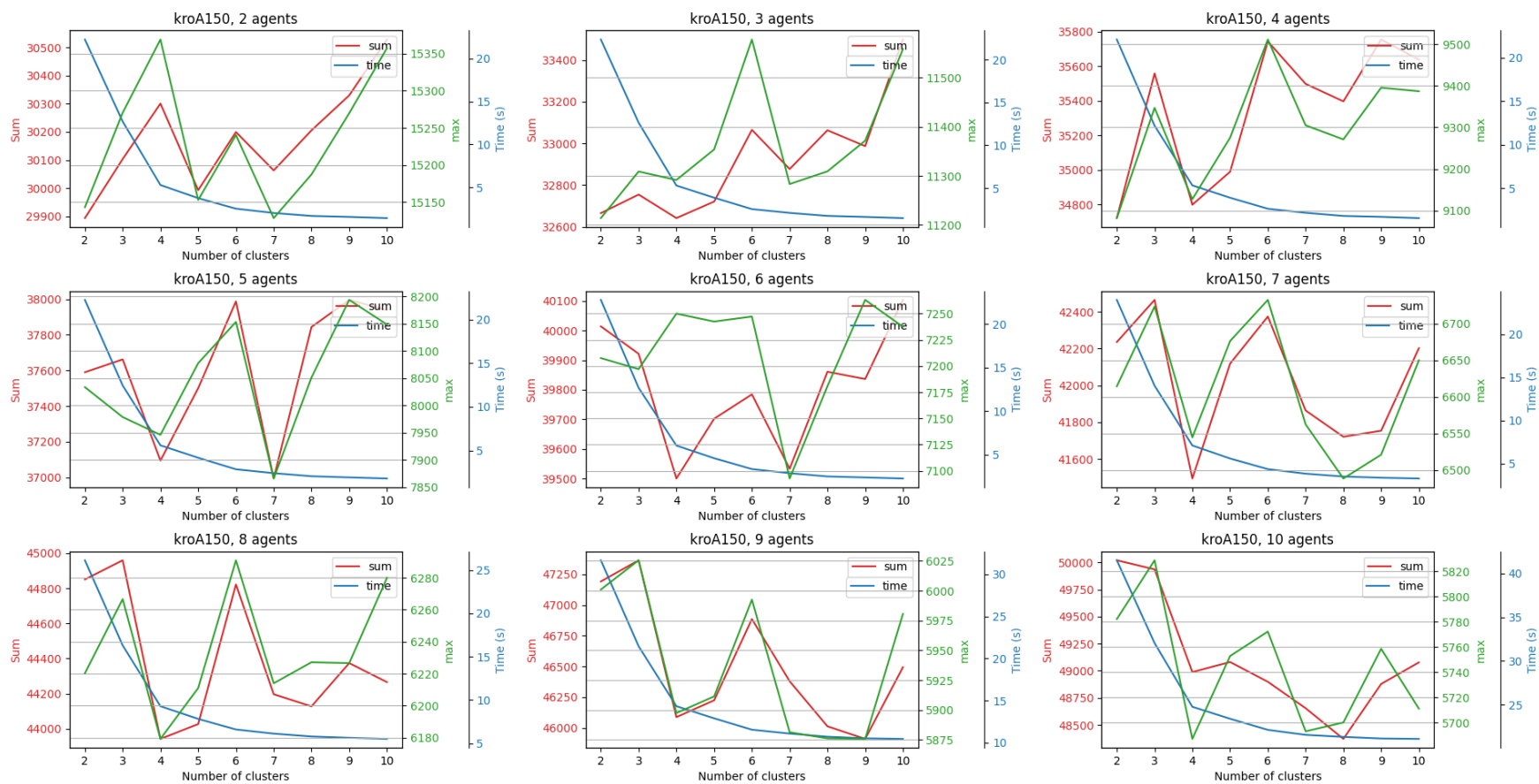


Рисунок 3.3. Влияние количества кластеров на время расчёта и качество решения при изменении количества агентов для задачи KroA150

3.1.2. Определение узлов соединения кластеров

После кластеризации и построения маршрутов необходимо объединить их в единый маршрут. Для этого требуется определить по два узла (города) в каждом кластере, которые будут служить точками соединения с другими кластерами [75].

Для определения этих узлов предлагается использовать следующую эвристику: выбирать по два узла в каждом кластере, которые находятся ближе всего к потенциальным соседним кластерам. При этом возникает вопрос, как правильно определить соседние кластеры, с которыми данный кластер должен напрямую соединиться, поскольку простое определение соседних кластеров на основе ближайших двух узлов, принадлежащим этим кластерам, может привести к некачественному решению. Рисунок 3.4 иллюстрирует подобную ситуацию.

Алгоритм соединения, основанный на выборе двух ближайших городов из разных кластеров, сначала соединяет города Γ_3 из кластера 1 и Γ_9 из кластера 3 (рисунок 3.4), так как они являются ближайшими друг к другу по сравнению с другими кластерами. Далее соединяются Γ_2 и Γ_4 , Γ_8 и Γ_{11} . Для завершения гамильтонова цикла остаётся соединить кластеры 2 и 4, при этом ближайшими узлами в этих кластерах являются Γ_5 и Γ_{10} . В итоге маршрут пересекается с самим собой, что свидетельствует о недостатках данной эвристики.

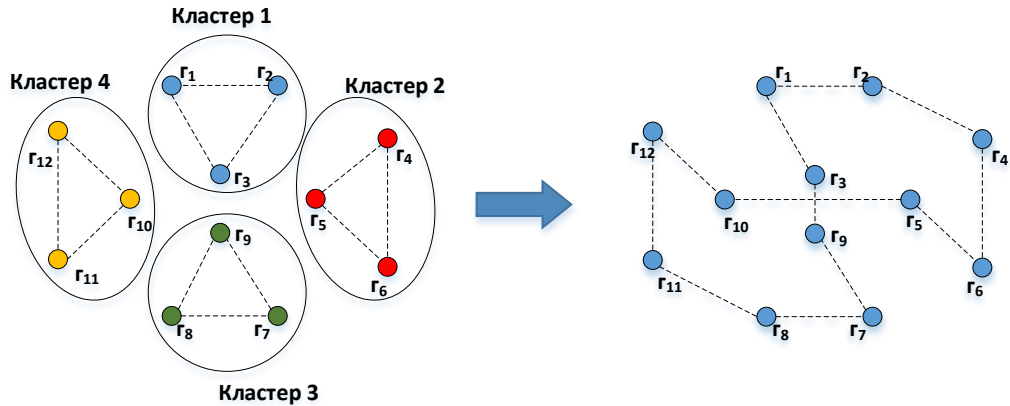


Рисунок 3.4. Пример пересечения маршрута с самим собой

Чтобы избежать подобной ситуации предлагается сначала найти оптимальный межкластерный гамильтонов цикл, проходящий через центры кластеров и соединять кластеры только с соседними кластерами по этому циклу. Таким образом, соединяются ближайшие города кластеров 1 и 2, 2 и 3, 3 и 4, 4 и 1, что позволяет избежать соединения кластеров 1 и 3, как показано на рисунке 3.5.

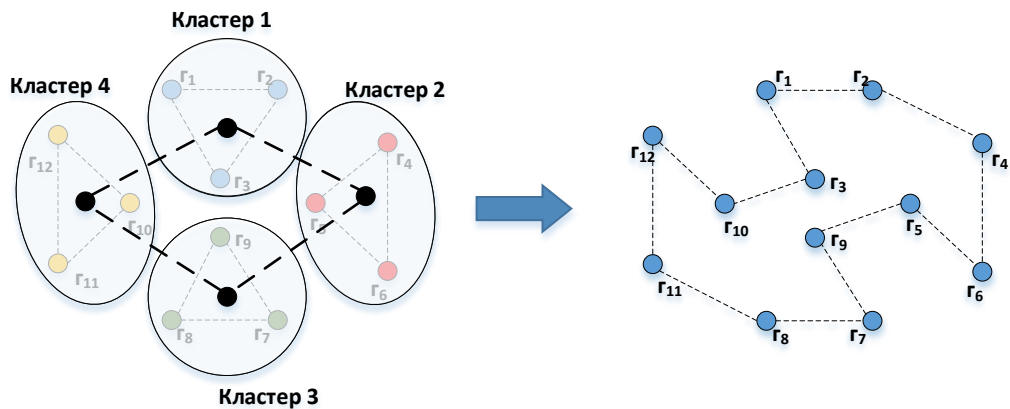


Рисунок 3.5. Пример предлагаемого решения выбора точек соединения кластеров с учётом межкластерного гамильтонова цикла

3.1.3. Определение внутрикластерных маршрутов

В подходе СКЗМ достаточно добавить город депо в каждый кластер и затем найти оптимальный гамильтонов цикл внутри каждого кластера чтобы

получить решение МКВ. Однако в предлагаемом методе такое решение оказывается неприемлемым, поскольку полученные подмаршруты необходимо заново соединять в супер-маршрут. Это приводит к существенному изменению порядка посещения городов внутри каждого кластера, часто далеко от оптимального [75].

Рисунок 3.6 демонстрирует ситуацию, в которой после построения оптимального гамильтонова цикла в каждом кластере происходит соединение соседних кластеров согласно эвристике, указанной в предыдущем параграфе: в кластерах 1 и 2 узлы Γ_5 и Γ_7 являются ближайшими, поэтому разрываем ребра $\Gamma_5\Gamma_6$ и $\Gamma_7\Gamma_{11}$ и соединяем Γ_5 и Γ_7 . В кластерах 2 и 3 разрываем ребра $\Gamma_{10}\Gamma_{11}$ и $\Gamma_{12}\Gamma_{13}$ и соединяем Γ_{10} и Γ_{13} . В кластерах 3 и 4 разрываем ребра $\Gamma_{16}\Gamma_{12}$ и $\Gamma_{17}\Gamma_{20}$ и соединяем Γ_{16} и Γ_{17} . В кластерах 4 и 1 разрываем ребра $\Gamma_{24}\Gamma_{20}$ и $\Gamma_4\Gamma_6$ и соединяем Γ_{24} и Γ_4 . В результате узлы Γ_6 , Γ_{11} , Γ_{12} и Γ_{20} выпадают из образованного цикла, что требует дополнительных вычислений для их включения в итоговый маршрут.

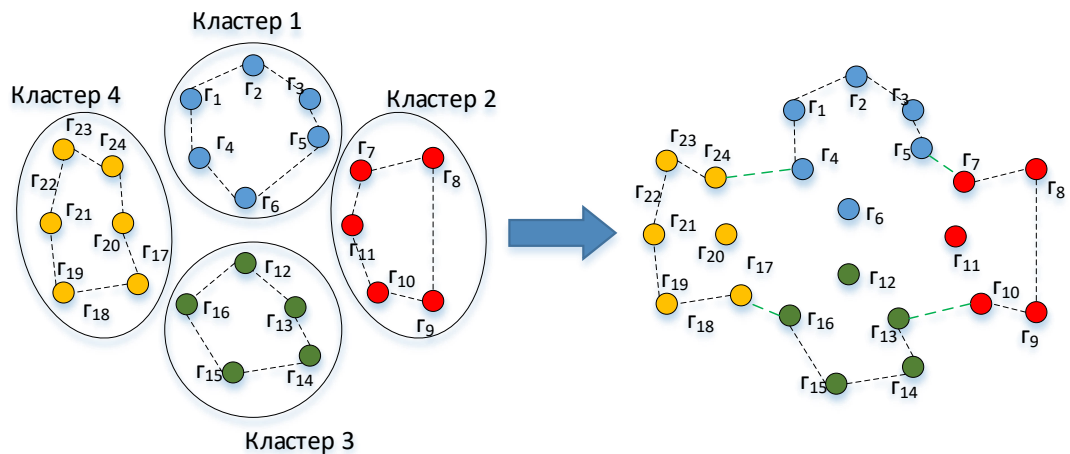


Рисунок 3.6. Пример возникновения проблемы при соединении маршрутов: после построения оптимальных гамильтоновых циклов внутри кластеров некоторые узлы остаются вне основного маршрута, что усложняет формирование цельного маршрута и требует дополнительных корректировок

Для решения этой проблемы предлагается выполнить этап определения узлов соединения перед этапом межкластерной маршрутизации и вместо поиска оптимального гамильтонова цикла в каждом кластере предлагается найти оптимальный гамильтонов путь — незамкнутый путь, проходящий через все вершины кластера. В качестве начального узла гамильтонова пути выбираем один из узлов соединения, а оставшийся узел соединения выбираем в качестве конечного узла. Такое решение позволяет избежать разрывов пути на этапе соединения кластеров и обеспечивает непрерывность супер-маршрута.

Таким образом, алгоритм предлагаемого метода будет выглядеть следующим образом:

Алгоритм 2 метод КМСП.

- 1 Инициализация гиперпараметров муравьиного алгоритма и количества кластеров.
- 2 Кластеризация с помощью алгоритма K-Means
- 3 Поиск оптимального межкластерного гамильтонова цикла
- 4 Определение узлов соединения кластеров
- 5 Поиск оптимального гамильтонова пути в каждом кластере
- 6 Соединение гамильтонова путей в супермаршрут
- 7 Разделение супермаршрута между агентами

3.2. Оценка масштабируемости предлагаемого метода

В данном параграфе анализируется масштабируемость предлагаемого метода (КМСП) в зависимости от сложности задачи, выраженной количеством городов и агентов. Цель анализа — оценить, как изменяется время вычисления решений с увеличением размера входных данных. Такой анализ позволит сформулировать ограничения на использование предлагаемого метода, а также эмпирически оценить его временную сложность.

Для проведения указанного анализа сформирована репрезентативная выборка, включающая результаты 100 запусков метода КМСР для решения задач с количеством городов (50, 75, 100, 150, 200). Для каждой задачи данные объединялись для 2, 3, 5, 7, 9 и 10 агентов, что позволяет оценить масштабируемость метода при различных постановках МКВ.

В таблице 3.1 показано среднее время расчета для 100 запусков КМСР для решения МКВ, включая каждую пару (количество городов, количество агентов).

Таблица 3.1. Результаты исследования масштабируемости метода КМСР.

Количество городов	Количество агентов	Среднее время расчета (секунд)
50	2	0.187
	3	0.193
	5	0.251
	7	0.568
	9	2.12
	10	4.443
75	2	0.384
	3	0.391
	5	0.472
	7	0.905
	9	3.032
	10	6.216
100	2	0.723
	3	0.731
	5	0.836

	7	1.404
	9	4.194
	10	8.368
150	2	1.775
	3	1.782
	5	1.929
	7	2.734
	9	6.675
	10	12.546
200	2	3.8716
	3	3.8719
	5	4.074
	7	5.204
	9	10.724
	10	18.93

На рисунке 3.7 представлен трёхмерный график влияния увеличения количества городов и агентов МКВ на время расчета решения методом КМСП. Проекция графика на плоскость XZ отображает функцию времени расчета от количества городов с учетом различного числа агентов. Проекция графика на плоскость YZ отображает функцию времени расчета от количества агентов с учетом разного числа городов.

Для эмпирической оценки временной сложности метода используется метод нормализации, основная идея которого заключается в сравнении измеренного времени расчета решения $T(x)$ с предполагаемыми моделями временной сложности $g(x)$ (например, $g(x) = \log(x)$, $g(x) = x \log(x)$, $g(x) = x^2$) путём нормализации данных следующим образом:

$$y(x) = \frac{T(x)}{g(x)}. \quad (3.1)$$

Будем считать, что модель $g(x)$ правильно отражает временную сложность метода, если нормализованные значения $y(x)$ имеют тенденцию оставаться приблизительно постоянными для разных размеров задач x , в том числе когда x обозначает количество городов n или количество агентов m .

Для нахождения временной сложности относительно количество агентов фиксируем количество городов и рассчитываем стандартное отклонения σ_y значения $y(x)$ при нескольких значений количества агентов, используя следующую формулу:

$$\sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2}, \quad (3.2)$$

где: \bar{y} – среднее значение $y(x)$ по разными количествами агентов.

Затем рассчитываем коэффициент вариации CV , который масштабирует σ_y по среднему значению, что позволяет сравнивать значения различных величин:

$$CV = \frac{\sigma_y}{\bar{y}}. \quad (3.3)$$

Чем ниже значение CV для заданной модели, тем точнее она представляет фактическую временную сложность метода.

Таблица 3.2 содержит значения CV для нескольких моделей временной сложности в зависимости от количества агентов в задаче m : $g(m) = m$, $g(m) = \log(m)$, $g(m) = m \log(m)$, $g(m) = m^2$, $g(m) = m^3$, $g(m) = 2^m$, $g(m) = m!$, модель с минимальным значением CV для заданного количества городов n выделена жирным шрифтом.

Таблица 3.2. значения CV для нескольких моделей временной сложности, относительно количество агентов задачи

n	CV m	CV $\log(m)$	CV $m \log(m)$	CV m^2	CV m^3	CV 2^m	CV $m!$
50	0.869	1.017	0.604	0.540	1.095	1.030	1.609
75	0.731	0.896	0.559	0.584	1.214	1.116	1.616
100	0.601	0.771	0.582	0.668	1.298	1.172	1.621
150	0.427	0.570	0.694	0.815	1.389	1.232	1.626
200	0.357	0.431	0.798	0.916	1.437	1.262	1.628

По определению нотации O метод, таким образом, имеет сложность:

$$T(m) = O(\max(m, m \log(m), m^2)) = O(m^2). \quad (3.4)$$

Таблица 3.3 содержит значения CV для нескольких моделей временной сложности в зависимости от количества городов в задаче n : $g(n) = n$, $g(n) = \log(n)$, $g(n) = m \log(n)$, $g(n) = n^2$, $g(n) = n^3$, $g(n) = 2^n$, $g(n) = n!$, модель с минимальным значением CV для заданного количества агентов m выделена жирным шрифтом.

Таблица 3.3. значения CV для нескольких моделей временной сложности, относительно количество городов задачи

m	CV n	CV $\log(n)$	CV $n \log(n)$	CV n^2	CV n^3	CV 2^n	CV $n!$
2	0.598	0.909	0.503	0.126	0.444	2.000	2.000
3	0.589	0.902	0.493	0.118	0.457	2.000	2.000
5	0.526	0.855	0.426	0.091	0.541	2.000	2.000
7	0.330	0.696	0.226	0.243	0.742	2.000	2.000
9	0.105	0.482	0.075	0.443	0.899	2.000	2.000
10	0.052	0.415	0.109	0.494	0.935	2.000	2.000

Метод, таким образом, имеет сложность:

$$T(n) = O(\max(n, n \log(n), n^2)) = O(n^2). \quad (3.5)$$

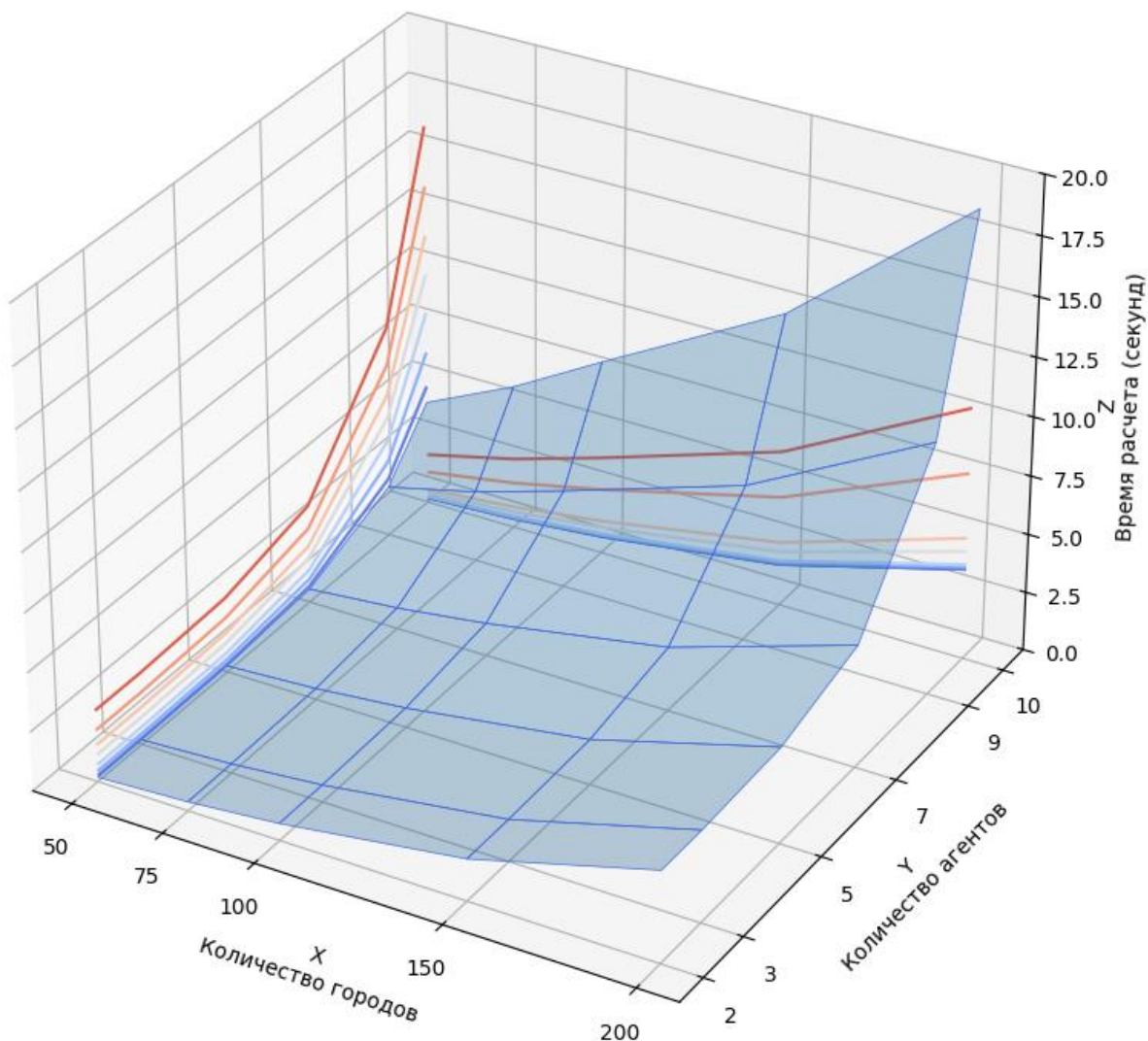


Рисунок 3.7. График влияния роста количества городов и агентов МКВ на время расчета решения методом КМСП

3.3. Моделирование и результаты

Для проведения сравнительного анализа предлагаемого метода с методами, реализующими подходы СКЗМ (метод КМ-САСО) и СМЗК (СМЗК-МА-2орт), были выбраны три задачи eil51, kroA100 и kroA150. Для каждой задачи были заданы три сценария с 3, 5 и 10 агентами. Каждый из этих сценариев решался по 100 раз каждым из рассматриваемых методов.

Критерии оценки эффективности изучаемых методов остаются теми же, что и в предыдущей главе (время расчета, MinSum, MinMax).

Для указанных выше методов, количество муравьев K составило n – количество городов МКВ, а количество итерации не указано, критерии остановки муравьиного алгоритма было повтора 20 итерации без уличения найденного решения. Методы были реализованы на языке Python 3.8. Моделирование проводилось на процессоре Intel Core i7-9700KF CPU @ 3.60GHz, с оперативной памятью 32GB, под управлением операционной системы Windows 10 Pro 64-bit.

Ниже представлены результаты статистического анализа 100 экспериментов для каждого из сценариев в виде преимущества использования метода КМСП по сравнению с методами, реализующими подходы СКЗМ и СМЗК.

На рисунках 3.8-3.16 представлены скрипичные диаграммы, отражающие процентное преимущество предлагаемого метода по критериям оценки качества решения по сравнению с методами, реализующими подходы СКЗМ (КМ-САСО) и СМЗК (СМЗК-МА-2opt). Ширина диаграммы отражает количество экспериментов с соответствующим значением преимущества. Отрицательное значение преимущества означает, что в некоторых случаях предлагаемый метод не улучшает, а, напротив, ухудшает результат по сравнению с остальными методами.

Eil51

Статистический анализ 100 экспериментов задачи eil51 с 3 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [41.53, 62.62] %, в среднем составляя 53.73%;

- суммарная длина маршрута изменяется в диапазоне [-9.92, 12.07] %, в среднем составляя 0.21%;
- максимальная длина среди маршрутов изменяется в диапазоне [-8.25, 23.17] %, в среднем составляя 8.37%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [92.59, 98.38] %, в среднем составляя 95.56%;
 - суммарная длина маршрута изменяется в диапазоне [-19.11, 8.97] %, в среднем составляя -5.89%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-16.07, 13.22] %, в среднем составляя -5.97%;

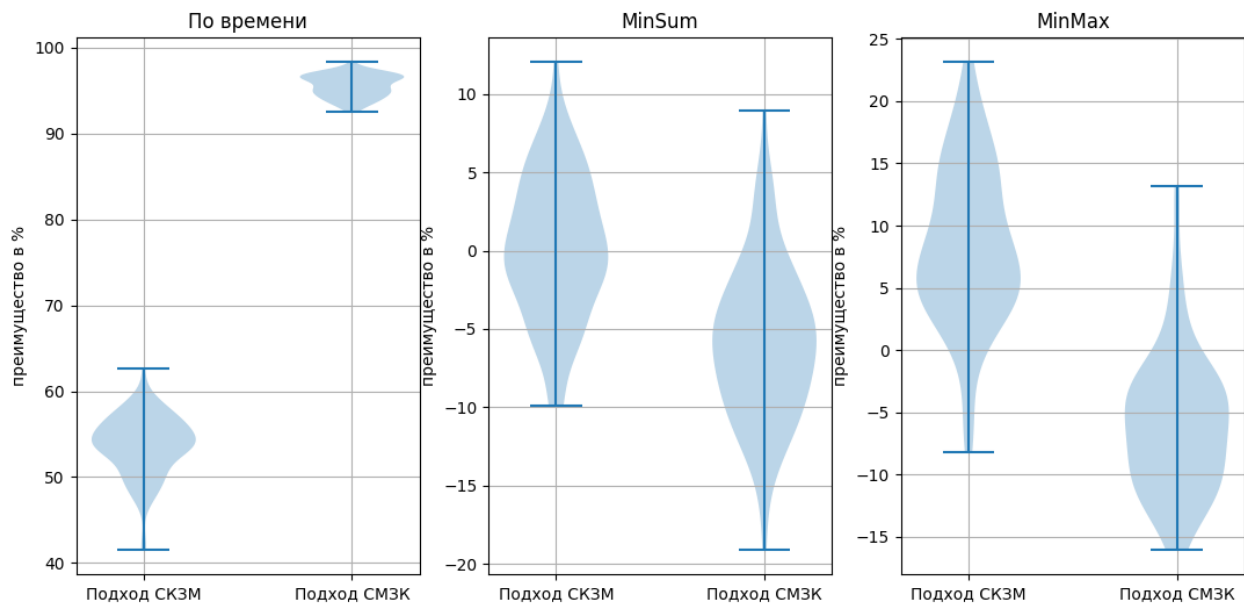


Рисунок 3.8. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задаче eil51 с 3 агентами

Статистический анализ 100 экспериментов задачи eil51 с 5 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:

- время расчёта решения изменяется в диапазоне [59.13- 81] %, в среднем составляя 67.86%;
- суммарная длина маршрута изменяется в диапазоне [-17.81, 4.01] %, в среднем составляя -5.32%;
- максимальная длина среди маршрутов изменяется в диапазоне [-15.99, 21.73] %, в среднем составляя 4.63%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [91.01, 96.91] %, в среднем составляя 93.93%;
 - суммарная длина маршрута изменяется в диапазоне [-11.41, 12.63] %, в среднем составляя 0.75%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-12.18, 28.28] %, в среднем составляя 0.00%;

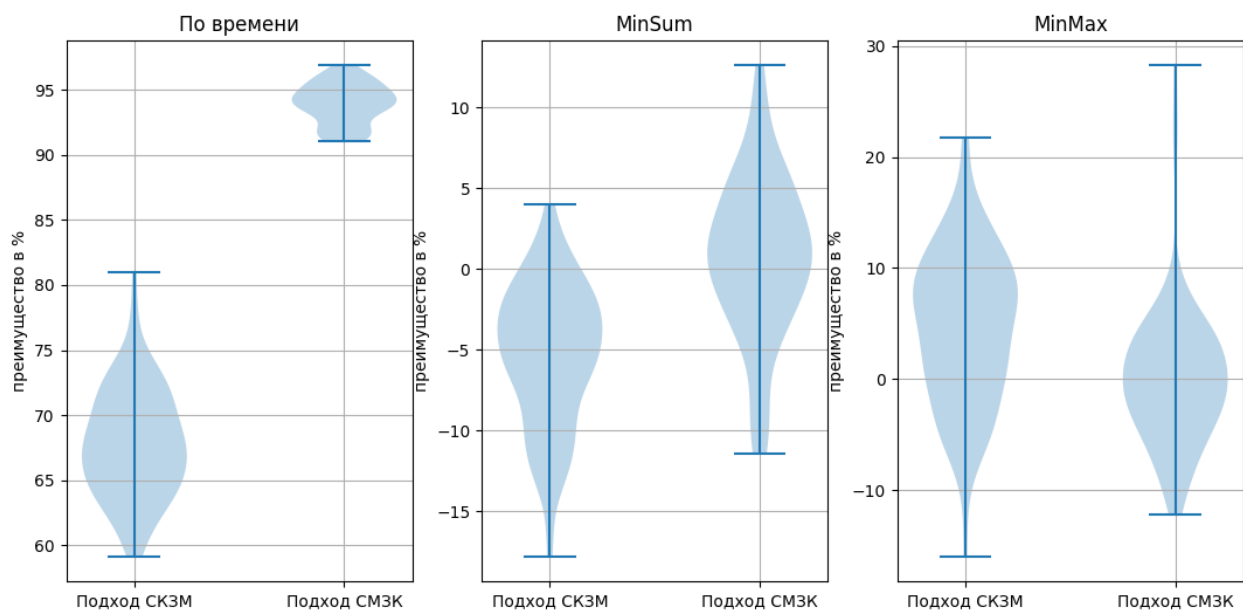


Рисунок 3.9. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задаче eil51 с 5 агентами

Статистический анализ 100 экспериментов задачи eil51 с 10 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [-269.81, -170.03] %, в среднем составляя -233.26%;
 - суммарная длина маршрута изменяется в диапазоне [-13.60, -1.68] %, в среднем составляя -7.50%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-3.42, 18.27] %, в среднем составляя 5.73%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [33.62, 60.22] %, в среднем составляя 45.86%;
 - суммарная длина маршрута изменяется в диапазоне [-5.52, 7.97] %, в среднем составляя 0.02%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-6.86, 5.26] %, в среднем составляя -0.74%;

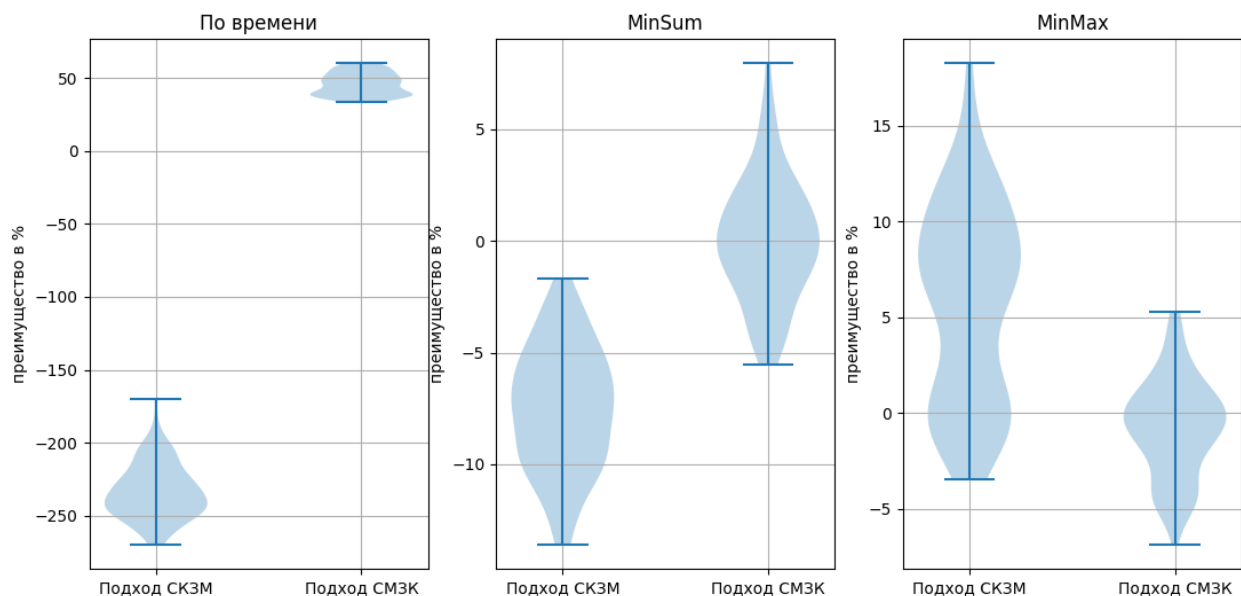


Рисунок 3.10. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи eil51 с 10 агентами

KroA100

Статистический анализ 100 экспериментов задачи KroA100 с 3 агентами показали, что в результате использования КМСР:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [-8.34, 60.26] %, в среднем составляя 31.71%;
 - суммарная длина маршрута изменяется в диапазоне [-13.12, 12.30] %, в среднем составляя 1.75%;
 - максимальная длина среди маршрутов изменяется в диапазоне [5.08, 32.12] %, в среднем составляя 21.10%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [94.25, 98.42] %, в среднем составляя 96.17%;
 - суммарная длина маршрута изменяется в диапазоне [-9.86, 5.57] %, в среднем составляя -2.49%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-13.17, 13.29] %, в среднем составляя -2.31%;

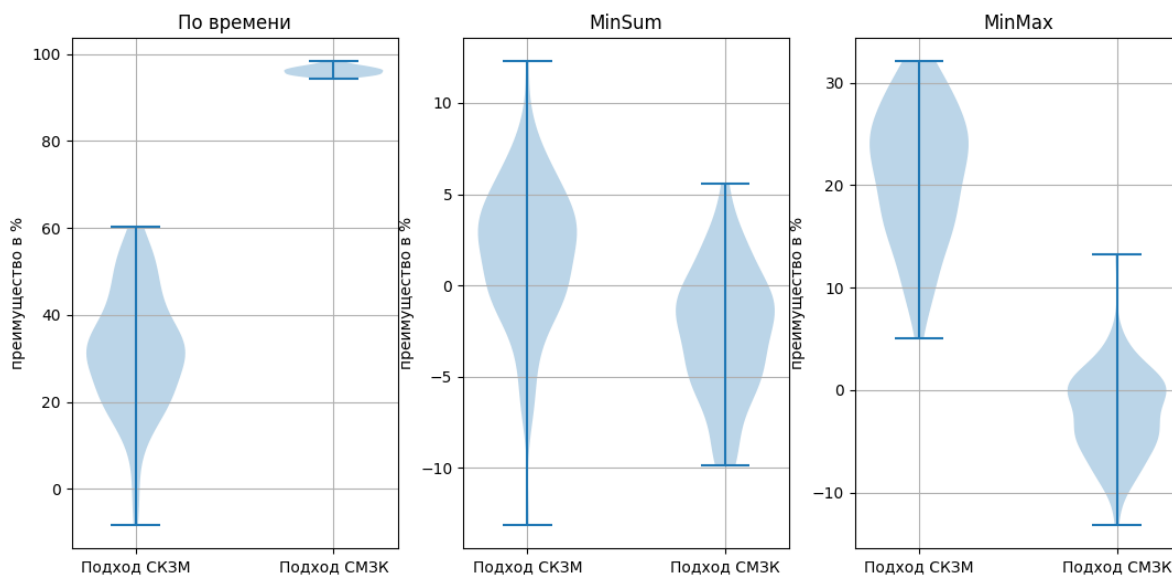


Рисунок 3.11. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи KroA100 с 3 агентами
 Статистический анализ 100 экспериментов задачи KroA100 с 5 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [23.89, 69.84] %, в среднем составляя 52.79%;
 - суммарная длина маршрута изменяется в диапазоне [-18.95, 1.34] %, в среднем составляя -5.85%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-6.73, 38.24] %, в среднем составляя 11.78%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [93.93, 98.06] %, в среднем составляя 95.61%;
 - суммарная длина маршрута изменяется в диапазоне [-13.28, 8.9] %, в среднем составляя -0.29%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-20.96, 18.03] %, в среднем составляя 2.56%;

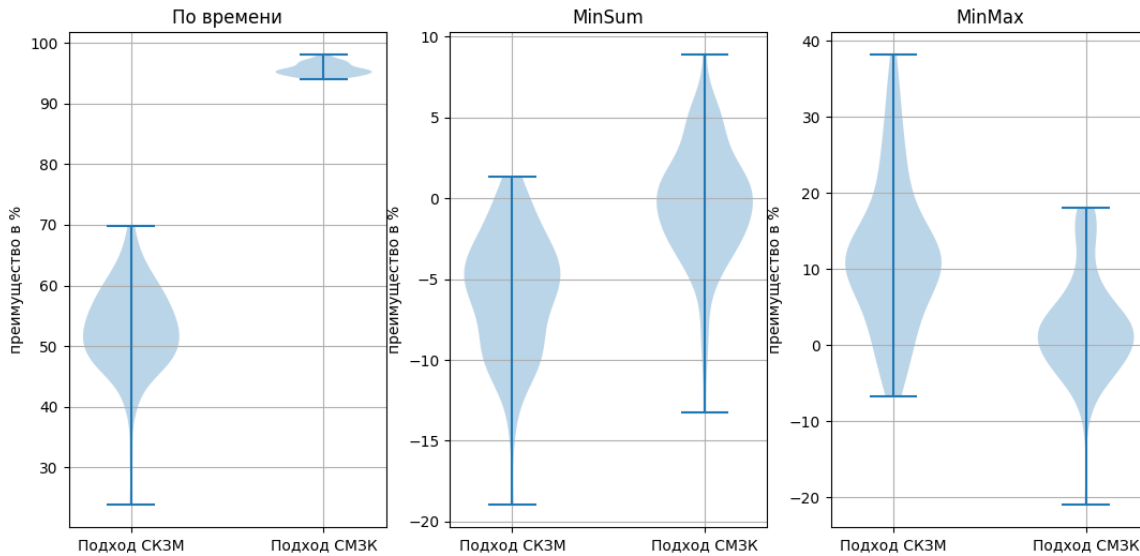


Рисунок 3.12. Скрипичные диаграммы преимуществ метода КМСР по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи KroA100 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA100 с 10 агентами показали, что в результате использования КМСР:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне $[-135.17, -54.9]$ %, в среднем составляя -103.58% ;
 - суммарная длина маршрута изменяется в диапазоне $[-24.18, -4.06]$ %, в среднем составляя -12.43% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-39.74, 22.47]$ %, в среднем составляя 2.84% ;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне $[67.82, 87.09]$ %, в среднем составляя 73.94% ;
 - суммарная длина маршрута изменяется в диапазоне $[-14.24, 4.65]$ %, в среднем составляя -2.33% ;
 - максимальная длина среди маршрутов изменяется в диапазоне $[-37.78, 24.54]$ %, в среднем составляя 0.63% ;

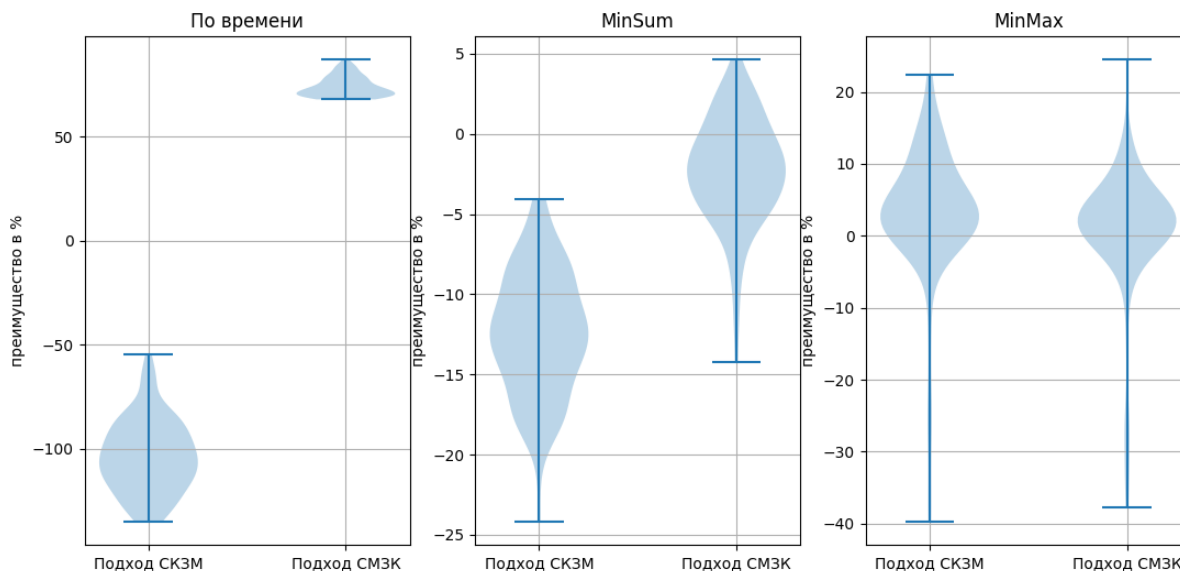


Рисунок 3.13. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи KroA100 с 10 агентами

KroA150

Статистический анализ 100 экспериментов задачи KroA150 с 3 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [-44.73, 52.63] %, в среднем составляя 7.40%;
 - суммарная длина маршрута изменяется в диапазоне [-4.57, 14.63] %, в среднем составляя 5.52%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-1.18, 38.33] %, в среднем составляя 25.46%;
- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [94.72, 97.98] %, в среднем составляя 96.49%;

- суммарная длина маршрута изменяется в диапазоне [-15.02, 8.62] %, в среднем составляя -1.64%;
- максимальная длина среди маршрутов изменяется в диапазоне [-35.01, 20.11] %, в среднем составляя -2.53%;

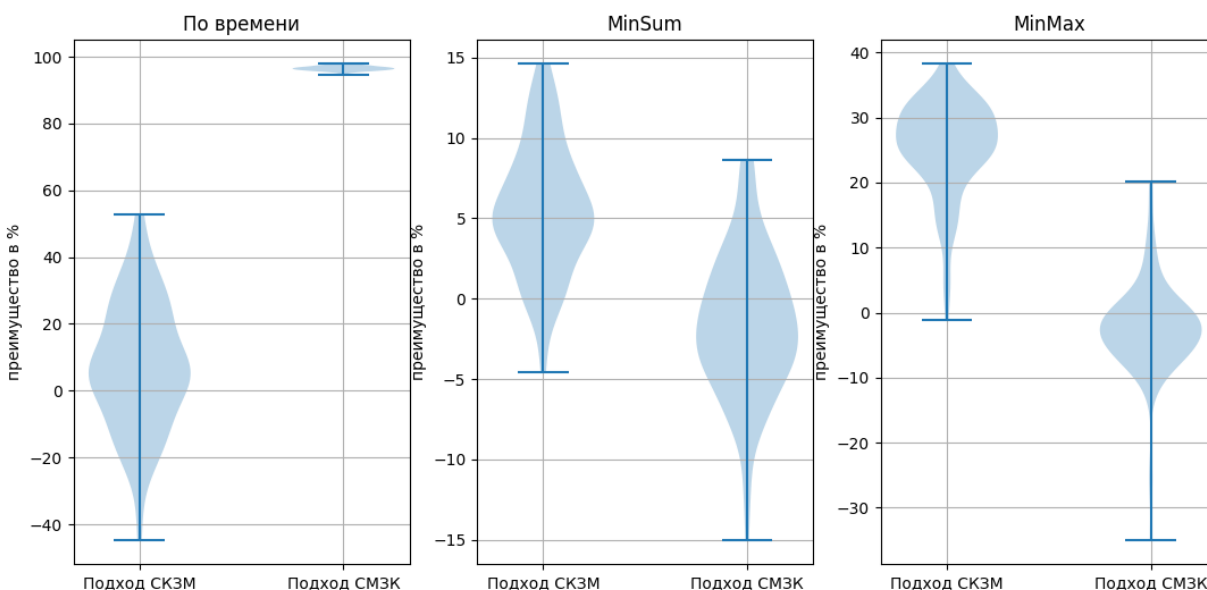


Рисунок 3.14. Скрипичные диаграммы преимуществ КМСР по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи KroA150 с 3 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 5 агентами показали, что в результате использования КМСР:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [8.11, 58.99] %, в среднем составляя 36.98%;
 - суммарная длина маршрута изменяется в диапазоне [-11.44, 7.81] %, в среднем составляя -2.52%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-12, 38.28] %, в среднем составляя 11.21%;
- по сравнению с СМЗК-МА-2opt:

- время расчёта решения изменяется в диапазоне [94.82, 98.43] %, в среднем составляя 96.37%;
- суммарная длина маршрута изменяется в диапазоне [-12.89, 12] %, в среднем составляя 0.05%;
- максимальная длина среди маршрутов изменяется в диапазоне [-21.34, 27.01] %, в среднем составляя 0.32%;

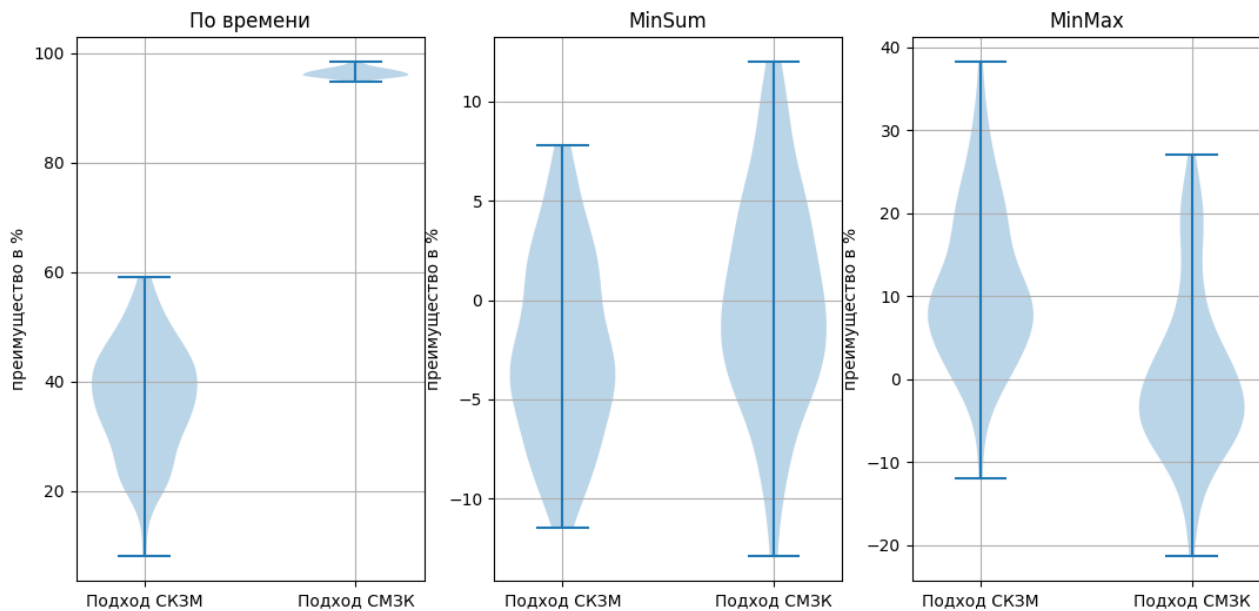


Рисунок 3.15. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задаче KroA150 с 5 агентами

Статистический анализ 100 экспериментов задачи KroA150 с 10 агентами показали, что в результате использования КМСП:

- по сравнению с КМ-САСО:
 - время расчёта решения изменяется в диапазоне [-91.17, -27.23] %, в среднем составляя -62%;
 - суммарная длина маршрута изменяется в диапазоне [-15.72, 0.18] %, в среднем составляя -7.66%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-26.79, 25.11] %, в среднем составляя 7.18%;

- по сравнению с СМЗК-МА-2opt:
 - время расчёта решения изменяется в диапазоне [79.08, 93.16] %, в среднем составляя 84.46%;
 - суммарная длина маршрута изменяется в диапазоне [-10.88, 9.14] %, в среднем составляя 0.67%;
 - максимальная длина среди маршрутов изменяется в диапазоне [-30.44, 27.98] %, в среднем составляя 0.71%;

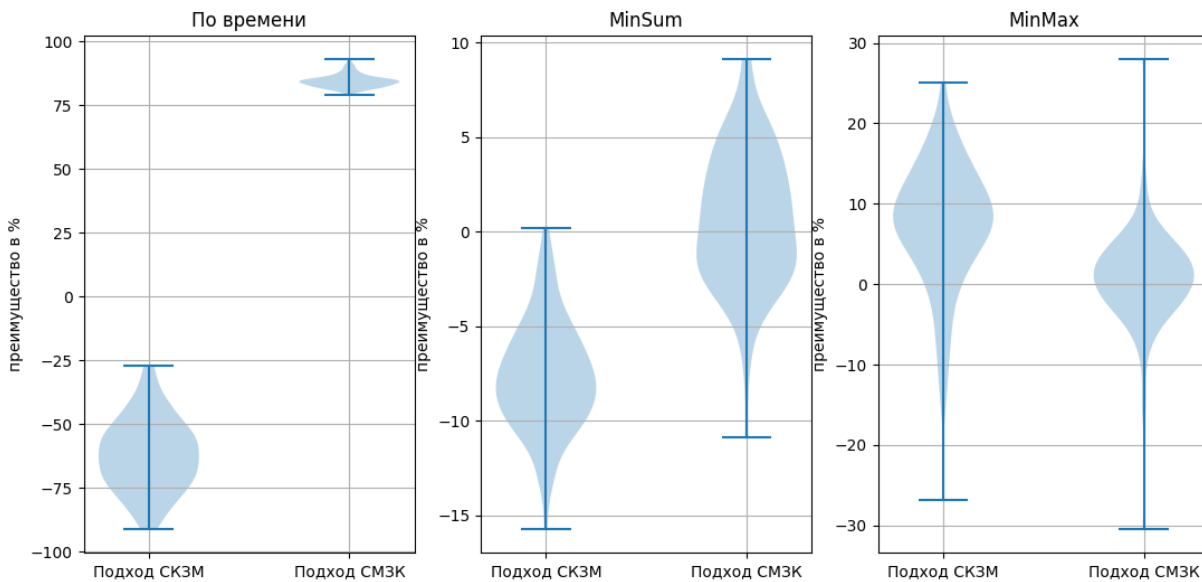


Рисунок 3.16. Скрипичные диаграммы преимуществ метода КМСП по сравнению с методами КМ-САСО и СМЗК-МА-2opt в задачи KroA150 с 10 агентами

Таблицы 3.4, 3.5 и 3.6 последовательно представляют результаты решения задач eil51, kroA100 и kroA150 с 3, 5 и 10 агентами (m). В таблицах приведены среднее значения по 100 запусков: время расчета решения в секундах (μ_t), сумма длин маршрутов в метрах (μ_{sum}) и максимальная длина маршрута среди коммивояжёров в метрах (μ_{max}). Соответствующие наилучшие значения выделены жирным шрифтом.

Таблица 3.4. Результаты решения задачи eil51.

m	3			5			10		
ме- тод	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП
μ_t	0.519	4.5	0.18	0.76	4.27	0.24	1.31	8.24	4.37
μ_{sum}	514.18	513.43	544.07	560.6	595.4	590.15	769.5	827.5	826.98
μ_{max}	190.87	177.43	187.1	133.33	127.08	126.61	99.78	93.16	93.8

Таблица 3.5. Результаты решения задачи kroA100.

m	3			5			10		
ме- тод	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП
μ_t	2.11	37.52	1.43	3.41	38.47	1.58	6.1	49.38	12.33
μ_{sum}	28054	26855	27491.53	30658	32360	32433	40412	44391	45412
μ_{max}	11775	9233	9436.69	7916	7111	6903	5550	5417	5374

Таблица 3.6. Результаты решения задачи kroA150.

m	3			5			10		
ме- тод	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП	КМ- САСО	СМЗК	КМСП
μ_t	5.87	152.61	5.15	8.41	151.4	5.19	15.21	162.8	24.45
μ_{sum}	34671	32006	32728	36192	37146	37081	45507	49344	48983
μ_{max}	15355	11058	11309	8998	8002	7920	6169	5749	5701

Заметим, что метод КМСП демонстрирует преимущество по времени расчёта для сценариев с 3 и 5 агентами, а также по критерию MinMax — для сценариев с 5 и 10 агентами.

По сравнению с методом СМЗК-МА-2орт предлагаемый метод снижает время расчёта на 33–98%, а максимальная длина маршрута уменьшается в среднем на 0.1–10% для задач с 5 и 10 агентами. При этом для задач с 3 агентами наблюдается незначительное увеличение максимальной длины маршрута в среднем на 2–5%.

Предлагаемый метод по сравнению с методом КМ-САСО снижает время расчёта на 12–68% в задачах с 3 и 5 агентами, но увеличивает его в среднем на 60–233% в задачах с 10 агентами. При этом максимальная длина маршрута уменьшается в среднем на 10.33% [76].

Причина увеличения время расчета при использовании предлагаемого метода для решения задачи с десятью агентами по сравнению с методом КМ-САСО заключается в том, что количество кластеров в методе КМ-САСО равно количеству агентов задачи — 10 кластеров, что больше, чем количество кластеров в предлагаемом методе — 4 кластера, следовательно, сложность проблемы и время расчета больше чем у метода КМ-САСО.

3.4. Выводы по главе 3

- Разработан гибридный метод решения многоагентной задачи коммивояжёра, основанный на объединении элементов подходов «Сначала кластеризация, затем маршрутизация» и «Сначала маршрутизация, затем кластеризация». В отличие от первого подхода, предлагаемый метод учитывает балансировку маршрутов по критерию MinMax, а в отличие от второго — обеспечивает сокращение времени вычислений за счёт предварительного разбиения задачи.

- Выявлено, что при использовании метода КМСР увеличение количества агентов МКВ сильнее влияет на время расчета, чем увеличение количества городов.
- Проведено сравнение эффективности предлагаемого метода решения МКВ с методами, основанными на подходах «Сначала маршрутизация, затем кластеризация» и «Сначала кластеризация, затем маршрутизация». Установлено, что предлагаемый метод обладает гибким механизмом достижения компромисса между качеством решения и временем расчёта.

Глава 4. Экспериментальные исследования разработанных методов

Целью исследования, проведенного в данной главе, является подтверждение достоверности количественных характеристик результатов сравнительных анализов, полученных в третьей главе, с помощью симулятора. В качестве объекта исследования рассматривается система управления группой подвижных объектов, состоящих из трёх однородных робототехнических платформ (РТП).

Симулятор – это программное обеспечение предназначенного для виртуального моделирования практического использования робототехнических комплексов. Программа функционирует на ПЭВМ типа IBM PC AT в операционной системе Windows «10». Для эксплуатации программного изделия необходим ПЭВМ со следующим минимальным набором технических средств: процессор – Intel Core I7 (8700К); ОЗУ – 32 Гб; видеоадаптер – SVGA (2шт. с ОЗУ не менее 8 Гб каждый); жесткий диск SSD – 500 Гб свободного места; ОС – Windows 10; интерфейсы: Ethernet (1 шт.).

Симулятор обеспечивает:

- имитацию функциональных систем и составных частей РТК с учётом их характеристик;
- моделирование физических свойств и формирование 3D-модели окружающего пространства в том числе:
 - имитацию городской застройки;
 - имитацию лесисто-холмистой местности с непреодолимыми препятствиями выше уровня поверхности (деревья, холмы);
 - имитацию лесисто-холмистой местности с непреодолимыми препятствиями ниже уровня поверхности (овраг, яма, водоем).

- настройку свойств и параметров, моделируемых в Симуляторе объектов окружающего пространства и природных явлений, а также параметров составных частей РТК;

- отображение результатов моделирования;
- отображение диагностической информации составных частей РТК, ввод исходных данных и маршрутного задания.

Основные технические характеристики системы группового управления приведены в таблице 4.1.

Таблица 4.1. Основные технические характеристики системы группового управления

Техническая характеристика	Значение
Количество РТП в группе, шт.	не более 5
Максимальная скорость движения при выполнении маршрутного задания группой РТП, км/ч	не более 30
Погрешность поддержания скорости, %	не более 10 от маршрутного задания
Сохранение расстояния заданной конфигурации группы при движении, %	не более 10 От маршрутного задания
Скорость движения группы РТП с автоматическим обходом препятствий, км/ч	не более 20

Система группового управления РТП служит для:

- реализации алгоритмов целеполагания с возможностью прогнозирования состояния группы РТП при скоординированном движении;

- реализации алгоритмов автоматического целераспределения с возможностью реконфигурации группы РТП в зависимости от условий окружающего пространства и по командам со стационарного пульта управления (СПУ) и носимого пульта управления (НПУ);
- реализации алгоритмов генерации/коррекции поведений РТП с возможностью адаптации поведения группы РТП;
- реализации алгоритмов группового скоординированного движения РТП.

Среда виртуального моделирования (Симулятор) предназначена для:

- численного моделирования и исследования математических моделей РТП, группы РТП;
- имитации и визуального представления процесса управления РТК в условиях окружающего пространства;
- разработки и исследования методов автономного управления движением РТП и обхода препятствий в режиме численного моделирования, включая режим моделирования реального времени;
- разработки и исследования методов и проверки алгоритмов управления скоординированным движением группы РТП;
- отработки маршрутного задания группы РТП;
- настройки параметров математических моделей и оптимизации параметров алгоритмов автономного управления движением группы РТП;
- численного моделирования и визуализации автономного управления движением группы РТП;

4.1. Моделирование

В главе 3 было отмечено, что метод КМСР позволяет снизить время расчёта решения при одновременном повышении эффективности процесса распределения задач. В связи с этим в данном разделе проводится проверка работоспособности методов, представленных в главах 2 и 3, а также численная оценка их преимуществ.

Исходное положение подвижных объектов: однородная группа РТП размещается в точке выезда в произвольном порядке. В настоящих экспериментах не учитываются погрешности позиционирования — координаты всех РТП считаются известными с высокой точностью. Системы технического зрения и связи не моделируются: среда является детерминированной, препятствия отсутствуют, каждый РТП обладает полной информацией о своём текущем положении.

На рисунке 4.1 показана среда моделирования. Сцена включает три автономных РТП, выполняющих запланированные маршруты в условиях виртуальной местности.

Интерфейс включает мини-карту (в левом верхнем углу) для глобального отслеживания положения РТП, панель выбора РТП (в нижней части экрана), а также отображение системных метрик в реальном времени (в правом верхнем углу).

Данная среда применялась для проверки осуществимости построенных маршрутов, выявления возможных аномалий в процессе выполнения заданий и наблюдения за координацией действий между агентами в управляемом трёхмерном пространстве.



Рисунок 4.1. Виртуальная среда моделирования

На рисунке 4.2 представлена визуализация вычисленных маршрутов для нескольких агентов с использованием ROS2 и RViz. Каждый цветной маршрут соответствует отдельному агенту, выполняющему назначенный ему сегмент решения МКВ. Узлы обозначают города задачи, а рёбра отражают запланированные пути обхода. Красные прямоугольники в центральной области отображают начальные позиции РТП, а наложение маршрутов подтверждает корректную сегментацию и распределение задач. Данная визуализация использовалась как для отладки, так и для проверки корректности алгоритмической работы во время моделирования.

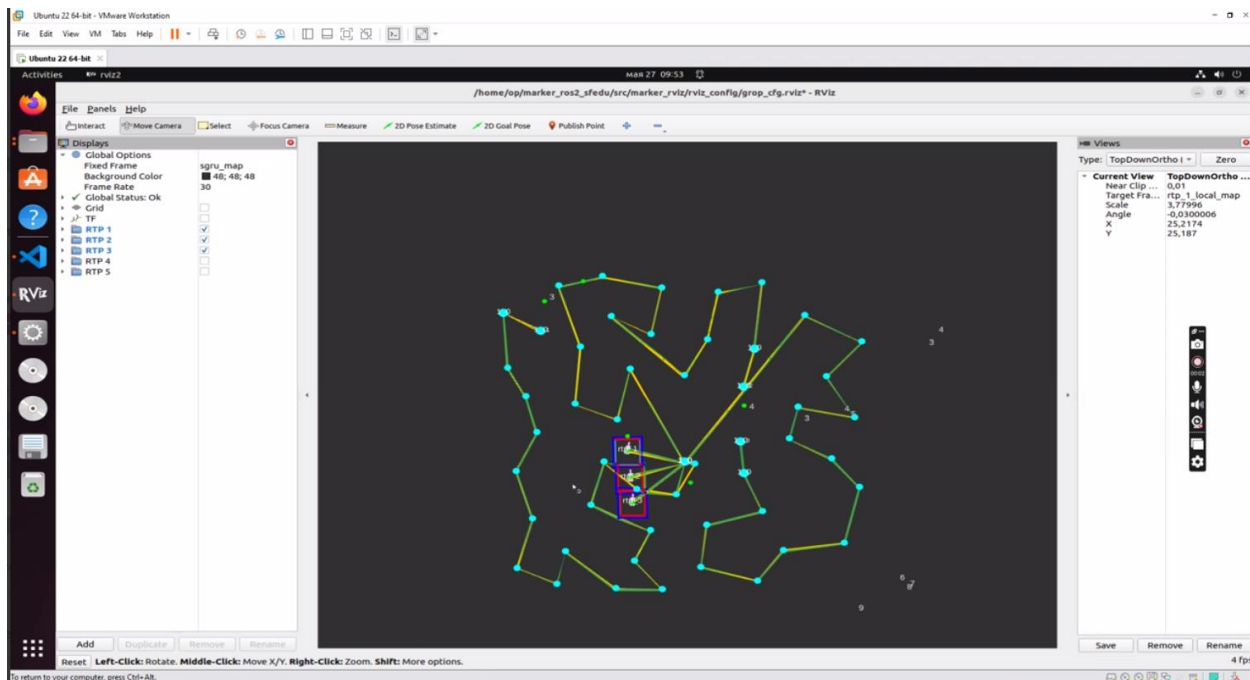


Рисунок 4.2. Визуализация выполнения траектории несколькими РТП в RViz

Для проведения сравнительного анализа производительности метода КМСП с методами КМ-САСО и СМЗК-МА-2opt были взяты в основу 100 задач, каждая из которых содержала 51 случайно сгенерированный город. Для каждой из этих задач был рассмотрен сценарий с 3 агентами.

Критерии оценки эффективности изучаемых методов остаются теми же, что и в предыдущей главе (время расчета, MinSum, MinMax).

Для методов КМ-САСО, СМЗК-МА-2opt и КМСП, количество муравьев муравьиного алгоритма K составило n – количество городов МКВ, а количество итерации не указано, критерии останова муравьиного алгоритма было повтора 20 итерации без уличения найденного решения. Методы были реализованы на языке Python 3.8. Моделирование проводилось на процессоре Intel Core i7-9700KF CPU @ 3.60GHz, с оперативной памятью 32GB, под управлением операционной системы Linux Ubuntu 14.

4.2. Результаты и сравнительный анализ

Таблица 4.2 представляет результаты решения каждой из сгенерированных задач. В таблице приведены: время расчёта решения в секундах (t), суммарная длина маршрута в метрах (sum) и максимальная длина маршрута среди агентов в метрах (max). Соответствующие оптимальные значения выделены жирным шрифтом.

Таблица 4.2. Результаты моделирования.

Номер эксперимента	Метод	KM-CACO	CMЗК-МА-2opt	KMCP
	Критерии			
1.	t	0.736	7.355	0.24
	sum	1030.955	974.201	950.725
	max	425.640	355.336	335.160
2.	t	0.984	5.912	0.278
	sum	950.547	971.444	935.466
	max	320.670	343.331	354.348
3.	t	0.745	3.911	0.372
	sum	1098.952	1012.857	1063.352
	max	417.544	374.363	373.724
4.	t	0.7	3.91	0.326
	sum	999.227	941.343	947.728
	max	376.388	336.346	340.938
5.	t	0.642	5.814	0.363
	sum	943.567	916.132	930.906
	max	318.173	334.693	326.882
6.	t	0.691	4.242	0.317

	<i>sum</i>	1066.470	955.732	987.249
	<i>max</i>	377.383	328.518	348.011
7.	<i>t</i>	0.64	6.931	0.475
	<i>sum</i>	1005.757	1024.672	930.513
	<i>max</i>	357.245	381.988	329.905
8.	<i>t</i>	0.644	4.152	0.35
	<i>sum</i>	980.962	988.198	945.657
	<i>max</i>	405.211	342.489	338.351
9.	<i>t</i>	0.823	5.778	0.302
	<i>sum</i>	949.038	868.807	935.900
	<i>max</i>	372.368	321.695	346.889
10.	<i>t</i>	0.599	5.229	0.292
	<i>sum</i>	994.192	908.615	954.770
	<i>max</i>	390.973	330.987	343.500
11.	<i>t</i>	0.588	6.211	0.414
	<i>sum</i>	985.154	987.357	1022.958
	<i>max</i>	392.294	356.338	366.623
12.	<i>t</i>	0.552	8.897	0.34
	<i>sum</i>	1034.001	937.127	978.473
	<i>max</i>	458.776	313.726	341.233
13.	<i>t</i>	0.588	5.27	0.35
	<i>sum</i>	1011.784	996.421	979.202
	<i>max</i>	429.057	356.292	338.506
14.	<i>t</i>	0.644	6.837	0.439
	<i>sum</i>	984.468	906.873	964.793

	<i>max</i>	363.855	320.036	336.642
15.	<i>t</i>	0.616	4.665	0.445
	<i>sum</i>	1062.410	1021.915	938.963
	<i>max</i>	497.310	343.914	336.372
16.	<i>t</i>	0.606	8.489	0.456
	<i>sum</i>	1084.748	988.459	988.160
	<i>max</i>	399.578	349.144	358.904
17.	<i>t</i>	0.576	6.511	0.345
	<i>sum</i>	1081.111	935.051	960.468
	<i>max</i>	387.128	339.373	327.722
18.	<i>t</i>	0.635	4.566	0.241
	<i>sum</i>	1063.983	913.278	928.491
	<i>max</i>	421.853	341.447	325.331
19.	<i>t</i>	0.572	5.611	0.262
	<i>sum</i>	1007.402	895.656	948.924
	<i>max</i>	357.967	312.551	324.433
20.	<i>t</i>	0.577	5.609	0.28
	<i>sum</i>	1103.625	1106.993	1038.672
	<i>max</i>	443.904	431.621	400.141
21.	<i>t</i>	0.647	7.612	0.287
	<i>sum</i>	971.992	894.442	915.807
	<i>max</i>	370.545	320.151	344.380
22.	<i>t</i>	0.56	5.67	0.289
	<i>sum</i>	984.497	887.225	941.160
	<i>max</i>	379.667	313.198	340.616

23.	<i>t</i>	0.624	3.895	0.286
	<i>sum</i>	1079.594	920.129	968.545
	<i>max</i>	422.129	326.876	368.037
24.	<i>t</i>	0.618	6.27	0.293
	<i>sum</i>	1045.377	1013.168	932.352
	<i>max</i>	435.580	351.478	324.367
25.	<i>t</i>	0.589	5.745	0.287
	<i>sum</i>	981.396	991.592	968.877
	<i>max</i>	370.454	342.911	379.840
26.	<i>t</i>	0.573	4.608	0.254
	<i>sum</i>	993.339	953.480	952.902
	<i>max</i>	413.259	348.334	347.135
27.	<i>t</i>	0.778	6.119	0.305
	<i>sum</i>	991.038	967.773	971.510
	<i>max</i>	379.441	356.198	328.172
28.	<i>t</i>	0.589	4.339	0.237
	<i>sum</i>	973.417	982.419	973.018
	<i>max</i>	419.676	338.083	347.322
29.	<i>t</i>	0.58	4.199	0.238
	<i>sum</i>	1090.186	962.938	961.686
	<i>max</i>	394.444	362.334	334.286
30.	<i>t</i>	0.69	6.015	0.261
	<i>sum</i>	1024.366	964.529	967.179
	<i>max</i>	403.114	327.818	330.065
31.	<i>t</i>	0.608	5.186	0.237

	<i>sum</i>	1019.354	972.356	973.629
	<i>max</i>	396.102	371.925	349.945
32.	<i>t</i>	0.604	6.534	0.282
	<i>sum</i>	1018.057	968.621	963.780
	<i>max</i>	348.108	373.036	334.404
33.	<i>t</i>	0.717	4.059	0.223
	<i>sum</i>	1034.170	946.629	966.824
	<i>max</i>	394.966	338.034	347.578
34.	<i>t</i>	0.615	6.987	0.24
	<i>sum</i>	998.999	998.263	962.147
	<i>max</i>	424.734	367.391	337.656
35.	<i>t</i>	0.686	5.613	0.253
	<i>sum</i>	1070.667	956.168	959.525
	<i>max</i>	387.328	322.179	361.939
36.	<i>t</i>	0.772	5.428	0.281
	<i>sum</i>	978.295	956.023	997.489
	<i>max</i>	351.999	357.878	375.741
37.	<i>t</i>	0.628	7.532	0.284
	<i>sum</i>	1010.137	1035.008	945.952
	<i>max</i>	363.763	355.181	338.211
38.	<i>t</i>	0.691	4.821	0.21
	<i>sum</i>	1047.645	961.180	946.830
	<i>max</i>	385.259	342.779	323.643
39.	<i>t</i>	0.613	4.581	0.276
	<i>sum</i>	983.164	984.868	954.930

	<i>max</i>	358.325	342.774	327.846
40.	<i>t</i>	0.662	6.949	0.203
	<i>sum</i>	1029.810	952.271	978.349
	<i>max</i>	376.782	336.570	374.381
41.	<i>t</i>	0.62	6.229	0.264
	<i>sum</i>	1000.526	919.321	957.398
	<i>max</i>	428.865	338.269	356.481
42.	<i>t</i>	0.604	5.23	0.215
	<i>sum</i>	1053.055	1006.056	957.683
	<i>max</i>	400.652	371.456	329.699
43.	<i>t</i>	0.599	7.911	0.248
	<i>sum</i>	1099.160	905.114	996.365
	<i>max</i>	389.081	313.298	367.618
44.	<i>t</i>	0.69	7.505	0.265
	<i>sum</i>	991.109	948.013	955.420
	<i>max</i>	371.793	343.037	363.679
45.	<i>t</i>	0.637	5.77	0.246
	<i>sum</i>	1108.656	942.096	961.730
	<i>max</i>	412.000	324.041	332.221
46.	<i>t</i>	0.794	5.334	0.25
	<i>sum</i>	1008.944	963.891	981.419
	<i>max</i>	382.245	330.465	354.293
47.	<i>t</i>	0.616	5.302	0.231
	<i>sum</i>	1036.402	963.271	984.187
	<i>max</i>	399.156	339.676	338.314

48.	<i>t</i>	0.551	4.868	0.249
	<i>sum</i>	1034.133	949.087	937.319
	<i>max</i>	375.552	331.927	344.921
49.	<i>t</i>	0.653	9.237	0.318
	<i>sum</i>	999.360	985.483	963.344
	<i>max</i>	337.196	338.677	354.420
50.	<i>t</i>	0.592	5.448	0.267
	<i>sum</i>	964.802	992.791	1013.302
	<i>max</i>	340.038	336.723	352.215
51.	<i>t</i>	0.595	8.028	0.239
	<i>sum</i>	1013.170	930.451	959.115
	<i>max</i>	418.614	323.460	346.363
52.	<i>t</i>	0.687	7.28	0.248
	<i>sum</i>	1026.529	951.950	939.741
	<i>max</i>	367.276	345.363	339.138
53.	<i>t</i>	0.64	6.259	0.25
	<i>sum</i>	1031.426	932.145	925.845
	<i>max</i>	424.097	338.909	348.017
54.	<i>t</i>	0.661	5.606	0.22
	<i>sum</i>	949.147	937.755	960.328
	<i>max</i>	369.765	337.721	331.884
55.	<i>t</i>	0.614	4.772	0.248
	<i>sum</i>	986.720	967.847	940.616
	<i>max</i>	447.614	328.697	331.867
56.	<i>t</i>	0.67	7.39	0.22

	<i>sum</i>	1080.359	981.899	925.337
	<i>max</i>	398.358	331.813	328.914
57.	<i>t</i>	0.733	5.718	0.223
	<i>sum</i>	1004.256	1009.964	955.231
	<i>max</i>	358.986	362.699	329.931
58.	<i>t</i>	0.596	5.162	0.269
	<i>sum</i>	1033.450	967.469	925.622
	<i>max</i>	386.651	344.956	326.575
59.	<i>t</i>	0.667	4.667	0.255
	<i>sum</i>	986.152	942.334	918.335
	<i>max</i>	356.044	317.394	319.206
60.	<i>t</i>	0.713	5.226	0.218
	<i>sum</i>	942.329	909.689	963.309
	<i>max</i>	388.999	330.451	346.547
61.	<i>t</i>	0.641	9.025	0.215
	<i>sum</i>	1024.109	889.138	957.106
	<i>max</i>	349.126	314.883	334.818
62.	<i>t</i>	0.646	8.612	0.212
	<i>sum</i>	925.236	976.142	962.394
	<i>max</i>	417.047	327.783	341.734
63.	<i>t</i>	0.649	4.241	0.236
	<i>sum</i>	998.738	1000.561	980.360
	<i>max</i>	423.206	351.054	343.628
64.	<i>t</i>	0.599	5.373	0.254
	<i>sum</i>	968.875	969.599	929.276

	<i>max</i>	357.838	359.439	330.467
65.	<i>t</i>	0.632	3.512	0.233
	<i>sum</i>	955.896	932.837	957.834
	<i>max</i>	335.306	352.489	326.041
66.	<i>t</i>	0.596	4.263	0.247
	<i>sum</i>	1032.975	947.685	928.004
	<i>max</i>	414.534	334.282	333.603
67.	<i>t</i>	0.606	6.387	0.261
	<i>sum</i>	1025.742	963.126	977.045
	<i>max</i>	362.976	367.217	335.363
68.	<i>t</i>	0.625	3.844	0.251
	<i>sum</i>	1060.979	963.691	1038.656
	<i>max</i>	428.200	355.814	388.871
69.	<i>t</i>	0.612	5.316	0.279
	<i>sum</i>	1051.486	968.229	990.518
	<i>max</i>	380.520	345.572	342.587
70.	<i>t</i>	0.599	5.179	0.281
	<i>sum</i>	1022.047	926.628	962.058
	<i>max</i>	368.817	315.733	332.236
71.	<i>t</i>	0.699	6.174	0.232
	<i>sum</i>	1052.167	975.340	915.557
	<i>max</i>	394.084	339.244	348.805
72.	<i>t</i>	0.597	5.759	0.279
	<i>sum</i>	1082.358	956.304	916.480
	<i>max</i>	416.696	328.353	332.109

73.	<i>t</i>	0.591	3.589	0.235
	<i>sum</i>	996.645	951.812	959.645
	<i>max</i>	337.679	336.086	326.222
74.	<i>t</i>	0.81	5.773	0.261
	<i>sum</i>	982.098	922.132	984.840
	<i>max</i>	442.139	358.001	333.514
75.	<i>t</i>	0.563	5.682	0.23
	<i>sum</i>	959.958	961.584	925.992
	<i>max</i>	360.006	359.962	331.758
76.	<i>t</i>	0.556	8.022	0.284
	<i>sum</i>	977.897	732.647	948.828
	<i>max</i>	362.614	304.842	340.446
77.	<i>t</i>	0.571	5.009	0.243
	<i>sum</i>	1049.576	910.110	941.103
	<i>max</i>	451.000	316.286	371.196
78.	<i>t</i>	0.604	5.896	0.245
	<i>sum</i>	1019.343	1226.839	924.025
	<i>max</i>	367.775	460.336	328.265
79.	<i>t</i>	0.679	4.79	0.279
	<i>sum</i>	947.720	972.312	969.921
	<i>max</i>	346.597	355.636	348.537
80.	<i>t</i>	0.556	3.935	0.226
	<i>sum</i>	1063.195	981.975	973.636
	<i>max</i>	382.479	333.327	336.207
81.	<i>t</i>	0.641	3.3	0.265

	<i>sum</i>	948.436	963.547	947.179
	<i>max</i>	328.036	361.402	328.667
82.	<i>t</i>	0.574	5.497	0.243
	<i>sum</i>	977.128	941.642	953.393
	<i>max</i>	435.692	343.939	333.049
83.	<i>t</i>	0.63	5.811	0.29
	<i>sum</i>	995.409	931.506	880.152
	<i>max</i>	383.232	315.185	299.145
84.	<i>t</i>	0.568	8.482	0.247
	<i>sum</i>	1059.498	951.261	932.412
	<i>max</i>	388.263	327.085	345.323
85.	<i>t</i>	0.544	5.041	0.26
	<i>sum</i>	993.555	944.481	1037.055
	<i>max</i>	409.335	348.662	360.034
86.	<i>t</i>	0.576	5.786	0.247
	<i>sum</i>	944.047	951.550	913.975
	<i>max</i>	335.039	321.916	317.634
87.	<i>t</i>	0.589	4.835	0.263
	<i>sum</i>	1018.193	972.185	977.739
	<i>max</i>	367.308	351.283	332.115
88.	<i>t</i>	0.58	5.015	0.272
	<i>sum</i>	1000.954	911.368	964.871
	<i>max</i>	382.350	332.959	335.337
89.	<i>t</i>	0.555	5.577	0.269
	<i>sum</i>	1056.292	1027.880	979.809

	<i>max</i>	386.136	363.560	330.582
90.	<i>t</i>	0.628	4.622	0.282
	<i>sum</i>	967.968	978.341	991.046
	<i>max</i>	418.049	338.786	351.866
91.	<i>t</i>	0.594	4.651	0.265
	<i>sum</i>	1044.537	911.885	1062.545
	<i>max</i>	368.146	331.117	376.460
92.	<i>t</i>	0.656	5.233	0.249
	<i>sum</i>	967.844	908.239	915.359
	<i>max</i>	363.613	317.859	328.641
93.	<i>t</i>	0.589	4.487	0.249
	<i>sum</i>	1085.598	864.708	973.426
	<i>max</i>	447.109	305.489	366.348
94.	<i>t</i>	0.576	7.28	0.259
	<i>sum</i>	1162.534	958.653	1060.834
	<i>max</i>	394.003	356.686	384.713
95.	<i>t</i>	0.609	6.551	0.246
	<i>sum</i>	1027.153	949.779	978.424
	<i>max</i>	413.950	353.647	335.417
96.	<i>t</i>	0.541	5.348	0.227
	<i>sum</i>	1097.659	1011.007	1013.676
	<i>max</i>	387.341	360.463	363.925
97.	<i>t</i>	0.611	4.113	0.278
	<i>sum</i>	1015.568	1009.287	992.354
	<i>max</i>	426.031	346.857	346.283

98.	<i>t</i>	0.592	6.267	0.197
	<i>sum</i>	1027.910	922.416	965.525
	<i>max</i>	406.382	358.647	331.615
99.	<i>t</i>	0.631	5.682	0.219
	<i>sum</i>	1026.774	974.768	998.603
	<i>max</i>	401.379	327.868	358.947
100.	<i>t</i>	0.546	9.755	0.269
	<i>sum</i>	1130.595	950.948	1034.691
	<i>max</i>	391.357	333.664	391.199

В таблице 4.3 приведены средние значения по 100 экспериментам: время расчёта решения в секундах (μ_t), суммарная длина маршрутов в метрах (μ_{sum}) и максимальная длина маршрута в метрах (μ_{max}).

Таблица 4.3. Средние значения результатов моделирования

Метод	μ_t	μ_{sum}	μ_{max}
КМ-САСО	0.631	1018.58	389.733
СМЗК-МА-2opt	5.747	957.909	342.517
КМСР	0.271	963.718	343.751

Метод СМЗК-МА-2opt показал наилучшую общую производительность с точки зрения качества решения, обеспечив наименьшую общую и максимальную длину маршрута. Однако это было достигнуто за счёт значительно большего времени вычислений — 5.747 секунд.

КМСР продемонстрировал конкурентоспособное качество решения — суммарная длина маршрута была выше всего на 0.6 %, а максимальная длина маршрута оказалась выше всего на 0.3 % по сравнению с СМЗК-МА-2opt. При

этом КМСП достиг самого быстрого времени вычислений (0.271 с), превосходя как СКЗМ, так и СМЗК по этому показателю.

На рисунке 4.3 показаны скрипичные диаграммы преимуществ в процентах по времени, суммарной длине маршрутов и максимальной длине маршрута соответственно (слева направо) при сравнении предлагаемого метода с методами КМ-САСО и СМЗК-МА-2opt. Отрицательное значение преимущества означает, что предлагаемый метод показывает худший результат по заданному критерию по сравнению с соответствующим методом. Ширина диаграммы отражает количество экспериментов с заданным значением преимущества.

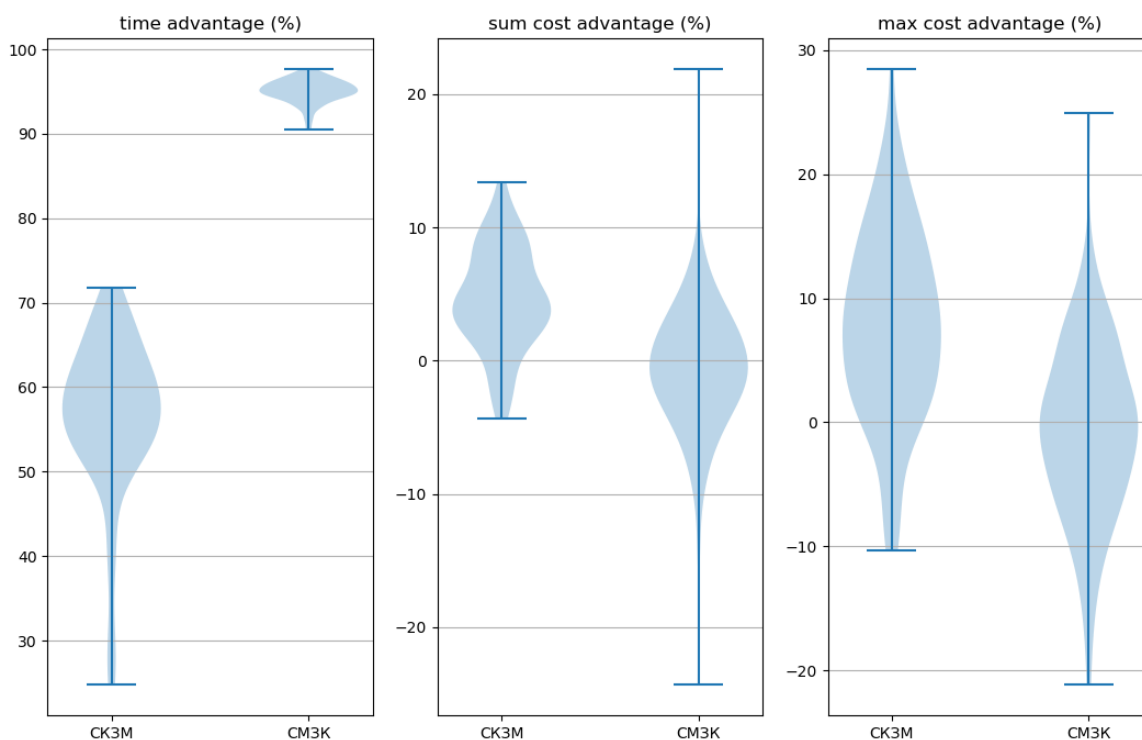


Рисунок 4.3. Сравнительное преимущество КМСП над КМ-САСО и СМЗК-МА-2opt

КМСП значительно превосходит оба метода по времени вычислений. Преимущество перед СМЗК-МА-2opt особенно заметно и стабильно: значения

плотно сгруппированы в диапазоне 95–98%, что указывает на то, что КМСП в среднем работает почти в 2 раза быстрее.

По сравнению с методом КМ-САСО экономия времени более вариативна и колеблется примерно от 25% до 70%, что подтверждает постоянно превосходящую эффективность КМСП.

С точки зрения суммарной длины маршрута КМСП превосходит КМ-САСО почти во всех запусках, обычно демонстрируя преимущество в диапазоне 5–10%. По сравнению с СМЗК-МА-2opt, КМСП иногда достигает более коротких маршрутов. Распределение результатов смещено немного ниже нуля.

КМСП последовательно снижает максимальную длину индивидуального маршрута по сравнению с КМ-САСО, показывая среднее преимущество около 15–20%. По сравнению с СМЗК-МА-2opt преимущество КМСП более сбалансировано, с более широким разбросом значений и медианой немного ниже нуля, что говорит о том, что СМЗК-МА-2opt иногда может превосходить КМСП, однако КМСП по-прежнему остается конкурентоспособным.

Эти распределения выделяют КМСП как высокоэффективный метод, обеспечивающий значительное ускорение процесса расчета решения МКВ при сохранении конкурентоспособного качества маршрутизации. В то время как СМЗК-МА-2opt лидирует в оптимизации суммарной длины маршрутов, КМСП демонстрирует лучшую масштабируемость и оперативность, что делает его особенно подходящим для приложений, где критически важны вычисления в реальном времени или ограниченные вычислительные ресурсы.

4.3. Статистическая значимость полученных результатов

Для оценки статистической значимости результатов, полученных в предыдущем подразделе, проведены выборочные Т-тесты распределений значений преимуществ по результатам 100 экспериментальных запусков.

T-тест — это статистический тест для проверки гипотез, используемый для сравнения средних значений двух групп [77]. Он определяет, есть ли статистически значимая разница между средними значениями, учитывая вариативность данных. Он особенно полезен, когда стандартное отклонение генеральной совокупности неизвестно, а размеры выборок относительно невелики.

Цель данного теста является подтверждения репрезентативности выборки, полученной в параграфе 4.2 — способность выборок отражать свойства генеральной совокупности.

Типичный алгоритм проведения статистического теста описан ниже.

Первым шагом статистического теста является сбор данных, то есть получить выборку. Это было сделано в предыдущем параграфе.

Далее необходимо сформулировать нулевую H_0 и альтернативную H_1 гипотезы, исходя из исследовательского вопроса. Нулевые гипотезы будут предполагать, что средние значения выигрыша при использовании метода КМРС равны средним значениям выигрыша, полученных при исследовании задачи eil51 с тремя агентами в третьей главе. Альтернативные гипотезы будут предполагать, что средние значения выигрыша при использовании метода КМРС меньше средних значений выигрыша, полученных при исследовании задачи eil51 с тремя агентами в третьей главе (левосторонние). Таким образом, в случае невозможности отвергнуть нулевые гипотезы можно считать, что выборки из исследования в главе 3 являются репрезентативными и достаточными для представления генеральной выборки.

Стоит отметить, что подтвердить гипотезу на основе экспериментальных данных нельзя — это фундаментальное ограничение. Всё, что можно сделать по итогам проверки — это отвергнуть гипотезу или нет.

Следующим шагом в общей схеме статистического анализа является построение выборочного распределения. Если H_0 верна, истинное среднее

значение генеральной совокупности равно числу, предположенному в нулевой гипотезе. Получается, что центр выборочного распределения — предполагаемое в H_0 значение. Выборочная дисперсия равна оценке стандартной ошибки выборочного среднего арифметического:

$$SE = \frac{\sigma}{\sqrt{N}}, \quad (4.1)$$

где N — объем выборки, а стандартное отклонение выборки σ рассчитывается по формуле:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{N - 1}}, \quad (4.2)$$

где x_i — элементы выборки, μ — среднее арифметическое.

С помощью построенного распределения нужно найти p-value — вероятность получить наблюдаемое или ещё более далёкое от предполагаемого в H_0 значения при условии, что H_0 верна. Другими словами, p-value представляет собой вероятность наступления данного события, указанного в нулевой гипотезе.

Что имеется в виду под «ещё более далёким», зависит от выбора H_1 : например, если H_1 правосторонняя, более далёкими от предполагаемого в H_0 значениями будут считаться те, что больше наблюдаемого, то есть справа от него.

Нулевая гипотеза отвергается, если вероятность получить среднее, рассчитанное по выборочным данным, слишком мала. Уровень статистической значимости α численно определяет, где проходит граница.

Уровень статистической значимости выбирается перед проведением статистического теста. Если p-value меньше выбранного уровня статистической значимости, то нулевая гипотеза отвергается в пользу альтернативной,

это означает, что наблюдаемое значение очень маловероятно при такой нулевой гипотезе, и мы отвергаем её в пользу альтернативной. А если больше, то не отвергается.

Далее проведен Т-тест на полученные с помощью симулятора выборки при α (уровень статистической значимости) = 10%

Выигрыш по времени расчета

H_0 : средний выигрыш по времени против метода КМ-САСО составляет 53.7%

H_1 : средний выигрыш по времени против метода КМ-САСО меньше, чем 53.7%

p-value = 0.99 > α

Результат: не получилось отвергнуть нулевую гипотезу

H_0 : средний выигрыш по времени против метода СМЗК-МА-2opt составляет 95.56%

H_1 : средний выигрыш по времени против метода СМЗК-МА-2opt меньше чем 95.56%

p-value = 0.609 > α

Результат: не получилось отвергнуть нулевую гипотезу

выигрыш по MinSum

H_0 : средний выигрыш по критерию MinSum MinSum против метода КМ-САСО составляет 0.21%

H_1 : средний выигрыш по критерию MinSum против метода КМ-САСО меньше чем 0.21%

p-value = 1 > α

Результат: не получилось отвергнуть нулевую гипотезу

H_0 : средний выигрыш по критерию MinSum против метода СМЗК-МА-2opt составляет -5.89%

H_1 : средний выигрыш по критерию MinSum против метода СМЗК-МА-2opt меньше чем -5.89%

$$p\text{-value} = 1 > \alpha$$

Результат: не получилось отвергнуть нулевую гипотезу

выигрыш по MinMax

H_0 : средний выигрыш по критерию MinMax против метода КМ-САСО составляет 8.37%

H_1 : средний выигрыш по критерию MinMax против метода КМ-САСО меньше чем 8.37%

$$p\text{-value} = 0.37 > \alpha$$

Результат: не получилось отвергнуть нулевую гипотезу

H_0 : средний выигрыш по критерию MinMax против метода СМЗК-МА-2opt составляет -5.97%

H_1 : средний выигрыш по критерию MinMax против метода СМЗК-МА-2opt меньше чем -5.97%

$$p\text{-value} = 0.99 > 0.1$$

Результат: не получилось отвергнуть нулевую гипотезу

Таким образом, можно сделать предположение, что в случае решения экземпляра МКВ с 50 городами и 3 агентами с помощью методов КМСР, КМ-САСО и СМЗК-МА-2opt существует высокая вероятность того, что выигрыш по критериям: время расчета, MinSum и MinMax при использовании КМСР будет равен соответствующим значениям, указанным в нулевых гипотезах.

4.4. Выводы по главе 4

- Предлагаемый метод КМСР, основанный на объединении элементов подходов «Сначала кластеризация, затем маршрутизация» и «Сначала маршрутизация, затем кластеризация» обеспечивает компромисс между качеством решения и временем вычислений решения МКВ. Несмотря на то, что СМЗК немного превосходит его по показателям MinSum и MinMax, КМСР значительно быстрее находит решения чем СКЗМ. Полученные результаты подтверждают, что КМСР является эффективной и практичной альтернативой для решения МКВ, особенно в случаях, когда вычислительная эффективность играет ключевую роль.

- Для оценки достоверности полученных результатов были проведены статистические проверки с использованием t-теста при уровне значимости 10%. По всем ключевым критериям оценки качества решений (время вычислений, MinSum, MinMax) p-value оказалось выше выбранного порога, что не позволило отвергнуть нулевую гипотезу об отсутствии статистически значимых различий.

Заключение

Основным результатом работы является решение актуальной научной задачи, обладающей практической значимостью: разработка, теоретическое обоснование и исследование методов и алгоритмов решения многоагентной задачи коммивояжёра. В качестве критериев оценки использовались время расчёта решения, суммарная длина маршрутов, и максимальная длина маршрута среди маршрутов агентов.

В диссертационной работе получены следующие результаты:

1. Улучшенный метод решения многоагентной задачи коммивояжёра на основе подхода «сначала маршрутизация, затем кластеризация», который позволяет повысить качество решения. Для сравнения предложенного метода с методами, реализующими подходы одновременной оптимизации и СКЗМ были проведены эксперименты на 3 эталонных задачах коммивояжёра из библиотеки TSPLIB. Анализ результатов показал, что предложенный метод превосходит метод ACO-BMTSP, реализующий подход одновременной оптимизации по всем рассматриваемым критериям оценки. В частности, время расчёта при использовании предложенного метода в среднем уменьшается на 9–54.5 % по сравнению с ACO-BMTSP, при этом суммарная длина маршрутов сокращается в среднем на 8.9–19.2 %, а максимальная длина маршрута среди коммивояжёров уменьшается в среднем на 12.4–20.2 %.

Предложенный метод превосходит KM-CACO, реализующий подход СКЗМ с точки зрения минимизации максимальной длины маршрута среди коммивояжёров. Однако время расчёта при использовании предлагаемого метода увеличивается в среднем на 1253 %. При этом суммарная длина маршрутов увеличивается в среднем на 2–9 % для сценариев с 5 и 10 коммивояжёрами,

и уменьшается в среднем на 0.1–7 % для сценариев с 3 коммивояжёрами. Максимальная длина маршрута среди коммивояжёров сокращается в среднем на 2–28.4 %.

2. Гибридный метод решения многоагентной задачи коммивояжёра, объединяющий подходы «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация», что позволяет снизить время расчёта решения и дисбаланс нагрузки. По сравнению с методом СМЗК-МА-2орт предложенный метод снижает время расчёта на 33–98%, а максимальная длина маршрута уменьшается в среднем на 0.1–10% для задач с 5 и 10 агентами. При этом для задач с 3 агентами наблюдается незначительное увеличение максимальной длины маршрута в среднем на 2–5%. Предложенный метод по сравнению с методом СМЗК-МА-2орт снижает время расчёта на 12–68 % в задачах с 3 и 5 агентами, но увеличивает его в среднем на 60–233 % в задачах с 10 агентами. При этом максимальная длина маршрута уменьшается в среднем на 1–26 %. Предложенный метод был интегрирован в симулятор РТП. По результатам вычислительных экспериментов можно сделать вывод, что метод КМСР демонстрирует хороший компромисс между качеством решения и временем вычислений. Хотя метод СМЗК-МА-2орт немного превосходит КМСР по показателям качества решения, КМСР значительно быстрее находить решения, чем метод КМ-САСО. Эти результаты подтверждают, что КМСР является эффективной и практичной альтернативой для решения МКВ, особенно в сценариях, где время расчёта играет решающую роль.

Полученные результаты были применены в работах, выполненных в Научно-исследовательском институте робототехники и процессов управления Южного федерального университета. Эти результаты могут быть использованы при разработке систем группового управления автономными подвиж-

ными объектами, что позволяет повысить эффективность их функционирования и сократить время расчёта решений. В частности, разработанные методы применимы для решения задач мониторинга в двумерных и трёхмерных средах. Также возможно масштабирование результатов для сценариев с большим количеством подвижных объектов в группе.

Все поставленные во введении задачи решены: проведён анализ существующих методов решения МКВ и выделены критерии качества, разработаны модифицированный метод на основе подхода СМЗК и гибридный метод КМСР, обеспечивающие компромисс между качеством решений и временем вычислений, а также выполнены их экспериментальные исследования в симуляторе. Полученные результаты подтверждают достижение цели диссертационной работы — повышение качества решений многоагентной задачи коммивояжёра при одновременном сокращении времени расчёта.

СПИСОК ЛИТЕРАТУРЫ

1. H. -L. Choi, L. Brunet and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," in IEEE Transactions on Robotics, vol. 25, no. 4, pp. 912-926, Aug. 2009
2. Германчук М. С. ПРИКЛАДНЫЕ ЗАДАЧИ МНОГОАГЕНТНОЙ МАРШРУТИЗАЦИИ // ТВИМ. 2021. №4 (53).
3. Козлова М.Г., Лемтюжникова Д.В., Лукьяненко В.А., Макаров О.О. МОДЕЛИ И АЛГОРИТМЫ МНОГОАГЕНТНОЙ ИЕРАРХИЧЕСКОЙ МАРШРУТИЗАЦИИ С ВРЕМЕННЫМИ ОКНАМИ // Известия Российской академии наук. Теория и системы управления. - 2023. - №5. - С. 103-126.
4. Е.Е. Полупанова, А.А. Рыбалко АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОЙ ГИБРИДИЗАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА, Известия ЮФУ. Технические науки, DOI 10.18522/2311-3103-2023-3-108-118, pp 108-117, 2023.
5. Тарков М.С., Дугаров Г.А. Параллельный алгоритм решения задачи коммивояжера с использованием рекуррентной нейронной сети // Проблемы информатики. 2010. №2.
6. В.В. Бурховецкий, Б.Я. Штейнберг, Точное и приближенное решения задачи коммивояжера большого размера, ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ И ПРОГРАММИРОВАНИЕ / NUMERICAL METHODS AND PROGRAMMING2024, 25 (4), 476–482.
7. Pengfei He, HaoJin-Kao Hao, Hybrid search with neighborhood reduction for the multiple traveling salesman problem, Computers & Operations Research 142(2), February 2022.
8. Yifan Guo, Zhongqiang Ren, Chen Wang, iMTSP: Solving Min-Max Multiple Traveling Salesman Problem with Imperative Learning, arXiv:2405.00285v4 [cs.AI] 23 Aug 2024.

9. Abhay Singh Bhadoriya, Deepjyoti Deka, and Kaarthik Sundar, Equitable Routing - Rethinking the Multiple Traveling Salesman Problem, Los Alamos National Laboratory, Los Alamos, New Mexico, USA. 2024.
10. Omar Cheikhrouhou, Ines Khoufi, A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy, *Computer Science Review*, Volume 40, 2021
11. K. Sundar, S. Rathinam, Algorithms for heterogeneous, multiple depot, multiple unmanned vehicle path planning problems, *J. Intell. Robot. Syst., Theory Appl.* 88 (2–4) (2017) 513–526, <http://dx.doi.org/10.1007/s10846-016-0458-5>.
12. M. Vali, K. Salimifard, A constraint programming approach for solving multiple traveling salesman problem, in: *The Sixteenth International Workshop on Constraint Modelling and Reformulation*, 2017.
13. Y. Shuai, S. Bradley, H. Shoudong, L. Dikai, A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms, *European J. Oper. Res.* (2013) 72–82.
14. M.A. Al-Omeer, Z.H. Ahmed, Comparative study of crossover operators for the mtsp, in: *2019 International Conference on Computer and Information Sciences (ICCIS)*, IEEE, 2019, pp. 1–6.
15. TSPLIB95, <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>, accessed: 2025-7-6.
16. Z. Wang, X. Fang, H. Li, H. Jin, An improved partheno-genetic algorithm with reproduction mechanism for the multiple traveling salesperson problem, *IEEE Access*.
17. H. Zhou, M. Song, W. Pedrycz, A comparative study of improved ga and pso in solving multiple traveling salesmen problem, *Appl. Soft Comput.* 64 (2018) 564–580.

18. R. Bolaños, M. Echeverry, J. Escobar, A multiobjective non-dominated sorting genetic algorithm (nsga-ii) for the multiple traveling salesman problem, *Decis. Sci. Lett.* 4 (4) (2015) 559–568.
19. C. Wei, Z. Ji, B. Cai, Particle swarm optimization for cooperative multi-robot task allocation: A multi-objective approach, *IEEE Robot. Autom. Lett.* 5 (2) (2020) 2530–2537.
20. M.R. Sierra, C.A.C. Coello, Improving pso-based multi-objective optimization using crowding, mutation and ϵ -dominance, in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2005, pp. 505–519.
21. E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm, TIK-report 103.
22. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: Nsga-ii, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
23. A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.C. Coello, F. Luna, E. Alba, Smpso: A new pso-based metaheuristic for multi-objective optimization, in: *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, IEEE, 2009, pp. 66–73.
24. Dorigo, M.; Birattari, M.; Stutzle, T. (2006). Ant colony optimization, 1(4), 28–39. doi:10.1109/mci.2006.329691
25. Z. Xu, Y. Li, X. Feng, Constrained multi-objective task assignment for UUVS using multiple ant colonies system, in: *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, Vol. 1, 2008, pp. 462–466, <http://dx.doi.org/10.1109/CCCM.2008.318>.
26. R. Necula, M. Breaban, M. Raschip, Tackling the bi-criteria facet of multiple traveling salesman problem with ant colony systems, in: *2015 IEEE 27th*

International Conference on Tools with Artificial Intelligence (ICTAI), 2015, pp. 873–880, <http://dx.doi.org/10.1109/ICTAI.2015.127>.

27. L.-C. Lu, T.-W. Yue, Mission-oriented ant-team aco for min–max mtsp, *Appl. Soft Comput.* 76 (2019) 436–444.

28. X. Chen, P. Zhang, G. Du, F. Li, Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems, *IEEE Access* 6 (2018) 21745–21757.

29. L. Ke, Q. Zhang, R. Battiti, MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony, *IEEE Trans. Cybern.* 43 (6) (2013) 1845–1859, <http://dx.doi.org/10.1109/TSMCB.2012.2231860>.

30. S. Trigui, O. Cheikhrouhou, A. Koubaa, U. Baroudi, H. Youssef, Fl-mtsp: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems, *Soft Comput.* 21 (24) (2017) 7351–7362.

31. P. Venkatesh, A. Singh, Two metaheuristic approaches for the multiple traveling salesperson problem, *Appl. Soft Comput.* 26 (2015) 74–89, <http://dx.doi.org/10.1016/j.asoc.2014.09.029>.

32. V. Pandiri, A. Singh, A swarm intelligence approach for the colored traveling salesman problem, *Appl. Intell.* 48 (11) (2018) 4412–4428.

33. S. Sariel, N. Erdogan, T. Balch, An integrated approach to solving the realworld multiple traveling robot problem, in: *5th International Conference on Electrical and Electronics Engineering*, 2007.

34. M. Elango, S. Nachiappan, M.K. Tiwari, Balancing task allocation in multirobot systems using k-means clustering and auction based mechanisms, *Expert Syst. Appl.* 38 (6) (2011) 6486–6491, <http://dx.doi.org/10.1016/j.eswa.2010.11.097>.

35. H.-L. Choi, L. Brunet, J.P. How, Consensus-based decentralized auctions for robust task allocation, *Robot., IEEE Trans.* 25 (4) (2009) 912–926.

36. R.K. Karmani, T. Latvala, G. Agha, On scaling multi-agent task reallocation using market-based approach, in: First International Conference on Self-Adaptive and Self-Organizing Systems, 2007. SASO'07, IEEE, 2007, pp. 173–182.
37. E. Kivelevitch, K. Cohen, M. Kumar, A market-based solution to the multiple traveling salesmen problem, *J. Intell. Robot. Syst.* 72 (1) (2013) 21–40, <http://dx.doi.org/10.1007/s10846-012-9805-3>.
38. O. Cheikhrouhou, A. Koubaa, H. Bennaceur, Move and improve: A distributed multi-robot coordination approach for multiple depots multiple travelling salesmen problem, in: 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2014, pp. 28–35, <http://dx.doi.org/10.1109/ICARSC.2014.6849758>.
39. A. Koubaa, O. Cheikhrouhou, H. Bennaceur, M.-F. Sriti, Y. Javed, A. Ammar, Move and improve: a market-based mechanism for the multiple depot multiple travelling salesmen problem, *J. Intell. Robot. Syst.* 85 (2) (2017) 307–330.
40. I. Khoufi, P. Minet, M. Koulali, A. Kobbane, Path planning of mobile sinks in charge of data gathering: A coalitional game theory approach, in: 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC), 2016, pp. 1–8.
41. S. Trigui, O. Cheikhrouhou, A. Koubaa, U. Baroudi, H. Youssef, Flmtsp: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems, *Soft Comput.* 21 (24) (2017) 7351–7362.
42. E. Kivelevitch, Mdmtpv_ga - multiple depot multiple traveling salesmen problem solved by genetic algorithm, 2011, <http://www.mathworks.com/matlabcentral/fileexchange/31814-mdmtpv-ga-multiple-depot-multiple-traveling-salesmen-problem-solved-by-genetic-algorithm>.

43. S. Trigui, O. Cheikhrouhou, A. Koubaa, A. Zarrad, H. Youssef, An analytical hierarchy process-based approach to solve the multi-objective multiple traveling salesman problem, *Intell. Serv. Robot.* 11 (4) (2018) 355–369.
44. O. Cheikhrouhou, A. Koubaa, A. Zaard, Analytical hierarchy process based multi-objective multiple traveling salesman problem, in: 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2016, pp. 130–136, <http://dx.doi.org/10.1109/ICARSC.2016.26>.
45. R. Saaty, The analytic hierarchy process-what it is and how it is used, *Math. Model.* 9 (3–5) (1987) 161–176, [http://dx.doi.org/10.1016/0270-0255\(87\)90473-8](http://dx.doi.org/10.1016/0270-0255(87)90473-8), URL <http://www.sciencedirect.com/science/article/pii/0270025587904738>.
46. A. Asma, B. Sadok, Pso-based dynamic distributed algorithm for automatic task clustering in a robotic swarm, *Procedia Comput. Sci.* 159 (2019) 1103–1112.
47. Lah I. A new kind of numbers and its application in the actuarial mathematics, *Boletim do Instituto dos Actuários Portugueses*, 1954, pp. 7–15.
48. Hamdan, Basma & Bashir, Hamdi & Cheaitou, Ali. A novel clustering method for breaking down the symmetric multiple traveling salesman problem. *Journal of Industrial Engineering and Management.* 14. 199. 2021.
49. Shabanpour M., Yadollahi M., Hasani M.M. A New Method to Solve the Multi Traveling Salesman Problem with the Combination of Genetic Algorithm and Clustering Technique, *IJCSNS International Journal of Computer Science and Network Security*, VOL.17 No.5, May 2017.
50. Basma H., Bashir H., Cheaitou A. A Novel Clustering Method for Breaking Down the Symmetric Multiple Traveling Salesman Problem, *Journal of Industrial Engineering and Management JIEM*, 2021, vol.14, no. 2.

51. Arthur D., Vassilvitskii S. K-Means++: The Advantages of careful seeding. Proceedings of the 8th annual ACM-SIAM symposium on Discrete algorithms, 2007.
52. Sofge, D., Schultz, A., De Jong, K. (2002). Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema. Proceedings of the Applications of Evolutionary Computing on EvoWorkshops (153-162).
53. Chandran, N., Narendrananesh, K., & Ganesh, K. (2006). A clustering approach to solve the multiple travelling salesmen problem. International Journal of Industrial and Systems Engineering, 1(3), 372-387. <https://doi.org/10.1504/IJISE.2006.009794>
54. Ma, J., Shi, S., & Gu, X., (2018). An optimization-based mTSP clustering algorithm for wireless sensor networks. Proceedings of 14th International Wireless Communications & Mobile Computing Conference (IWCMC) (334-338). IEEE. <https://doi.org/10.1109/IWCMC.2018.8450272>
55. A. Asma, B. Sadok, Pso-based dynamic distributed algorithm for automatic task clustering in a robotic swarm, Procedia Comput. Sci. 159 (2019) 1103–1112.
56. Kusumahardhini N., Hertono G. F., Handari B.D. Implementation of K-Means and crossover ant colony optimization algorithm on multiple traveling salesman problem, Basic and Applied Sciences Interdisciplinary Conference, 2017.
57. Latah M. Solving multiple TSP problem by K-means and crossover based modified ACO algorithm, International Journal of Engineering Research and Technology, vol. 5, no. 02, 2016.
58. S. Trigui, A. Koubaa, O. Cheikhrouhou, B. Qureshi, H. Youssef, A clustering market-based approach for multi-robot emergency response applications,

in: 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2016, pp. 137–143, <http://dx.doi.org/10.1109/>

59. Graham R.L., Knuth D.E., Patashnik O. Concrete Mathematics, Addison–Wesley, Reading MA. ISBN 0-201-14236-8, 1988, p. 244.

60. Beasley J.E. Route First - Cluster Second Methods for Vehicle Routing // Omega. 1983. Vol. 11, Issue 4, - P. 403-408, Bozdemir M. K., Bozdemir M., Burcu Ö. Route First- Cluster Second Method for Personal Service Routing Problem // Journal of Engineering Studies and Research. 2019. Vol. 25. No. 2. - P. 18-24.

61. Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. 31, 1985–2002.

62. Haimovich, M., Rinnooy Kan, A.H.G., 1985. Bounds and heuristics for capacitated routing problems. Math. Oper. Res. 10, 527–542.

63. Liang, H., Ma, Y., Cao, Z., Liu, T., Ni, F., Li, Z., & Hao, J. (2023). SplitNet: A Reinforcement Learning Based Sequence Splitting Method for the MinMax Multiple Travelling Salesman Problem. Proceedings of the AAAI Conference on Artificial Intelligence, 37(7), 8720-8727. <https://doi.org/10.1609/aaai.v37i7.26049>

64. Kool, Wouter et al. “Attention, Learn to Solve Routing Problems!” International Conference on Learning Representations (2018).

65. Lin and B. W. Kernighan “An effective heuristic algorithm for the traveling-salesman problem”, Operations Research 21, cc. 498-516. 1973.

66. Gambardella L.M., Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In Machine Learning Proceedings 1995; Elsevier: Amsterdam, The Netherlands, 1995. – С. 252–260.

67. Хуссейн Ф. А., Финаев В. И. Исследование эффективности алгоритма искусственных потенциалов, муравьиного алгоритма и их комбинации при планировании траектории движения мобильного робота. Компьютерные и

информационные технологии в науке, инженерии и управлении (КомТех-2020), 2020. – С. 39-48

68. Ф. А. Хуссейн, В. А. Костюков, Сравнительный анализ методов решения задачи целераспределения и маршрутизации для мультиагентных робототехнических систем, журнал «Мехатроника, автоматизация, управление», Том 26, № 3 (2025) СС. 139-146.

69. Dorigo, M.; Birattari, M.; Stutzle, T.. (2006). Ant colony optimization. , 1(4), 28–39. doi:10.1109/mci.2006.329691

70. de Castro Pereira S., Solteiro Pires E.J., de Moura Oliveira P.B. Ant-Balanced Multiple Traveling Salesmen: ACO-BMTSP. Algorithms 2023, P. 37.

71. Ф.А. Хуссейн, Разработка и исследование метода централизованного распределения задач в мультиагентных системах. Известия ЮФУ. Технические науки. – 2024. – № 4(240). – С.40-49. – DOI 10.18522/2311-3103-2024-4-40-49

72. Ф.А. Хуссейн, В.А. Костюков, И.Д. Евдокимов. метод решения проблемы мульти-коммивояжера в среде без препятствий на основе уменьшения размера пространства решений, Известия ЮФУ. Технические науки. – 2024. – № 1(237). – С. 181-192. – DOI 10.18522/2311-3103-2024-1-181-192.

73. Firas A. Houssein, Vladimir F. Kostyukov, and Igor D. Evdokimov, A method for solving the multi-traveling salesman problem based on reducing the size of the solution space, CODIT 2024. СС. 1729-1733

74. Ф.А. Хуссейн, В.А. Костюков. Гибридный метод решения многоагентной задачи коммивояжера, Известия ЮФУ. Технические науки. – 2025. – № 2 (244). – С. 193-201. – DOI 10.18522/2311-3103-2025 -2-193-201.

75. Ф.А. Хуссейн, В.А. Костюков, Разработка и исследование метода решения задачи целераспределения в многоагентной системе, Известия ЮФУ.

Технические науки. – 2025. – № 4 (246). – С. 144-155. – DOI 10.18522/2311-3103-2025-4-144-155.

76. Пшихопов В. Х., Медведев М. Ю., Костюков В. А., Хуссейн Ф., Кадим А., Алгоритмы планирования траекторий в двумерной среде с препятствиями. Информатика и автоматизация, 21(3), СС. 459-492. 2022.

77. Yim KH, Nahm FS, Han KA, Park SY. Analysis of statistical methods and errors in the articles published in the korean journal of pain. Korean J Pain 2010; 23: 35-41.

Приложения

Приложение 1

УТВЕРЖДАЮ

Генеральный директор АО НПО
«Андрондная техника»
Пермяков А.Ф.
«8» декабря 2025 г.



АКТ

внедрения результатов диссертации младшего научного сотрудника научно-исследовательского института робототехники и процессов управления Южного федерального университета Хуссейна Фираса Аймановича «Методы решения многоагентной задачи коммивояжера на основе сокращения поискового пространства» на соискание ученой степени кандидата технических наук при разработке методов и алгоритмов распределения задач в многоагентных системах

«8» декабря 2025 г.

Комиссия в составе:

председатель:

Дудоров Евгений Александрович, Главный конструктор

члены:

Чеха Владислав Владимирович, Директор по перспективным проектам

Рубан Мария Сергеевна, Исполнительный директор

подготовила настоящий акт о том, что при разработке и исследовании сценариев применения групп РТП МАРКЕР при решении типовых задач в рамках проекта «Экспериментально-теоретические исследования по отработке технологии автономного управления движением группы наземных робототехнических платформ» (шифр – Маркер-Группа) использованы следующие научные результаты диссертационной работы младшего научного сотрудника научно-исследовательского института робототехники и процессов управления Южного федерального университета Хуссейна Фираса Аймановича:

– метод решения многоагентной задачи коммивояжера, основанный на подходе «сначала маршрутизация, затем кластеризация», отличающийся использованием алгоритма локального поиска для повышения качества решения;

– гибридный метод решения многоагентной задачи коммивояжера, сочетающий подходы «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация», объединяющий их сильные стороны и устраняющий их недостатки.

Использование предложенных решений при разработке и исследовании сценариев группового применения робототехнических платформ (РТП) позволило повысить качество выполнения задачи целераспределения при решении типовых задач.

Эффективность предложенных Хуссейном Ф.А. решений подтверждена результатами моделирования применения РТП при решении типовых задач на моделирующем комплексе, разработанном в рамках проекта «Маркер-Группа».

Анализ эффективности применения группы РТП по поражению наземных целей выявил существенные улучшения показателей эффективности по сравнению с дистанционно управляемыми системами:

снижение времени расчёта оптимального решения задачи целераспределения до 95%;

повышение качества найденного решения до 27%.

Председатель



(подпись)

Дудоров Е.А.

Члены



(подпись)

Чеха В.В.



(подпись)

Рубан М.С.

УТВЕРЖДАЮ

Ген. директор АО «НКБ Робототехники и систем управления»

«27 августа» 2019 г. Е.Ю. Косенко



АКТ

использования результатов диссертации по теме «Разработка и исследование методов решения многоагентной задачи коммивояжера» на соискание ученой степени кандидата технических наук младшего научного сотрудника АО «Научно-конструкторское бюро Робототехники и систем управления» Хуссейна Фираса Аймановича

Настоящим актом подтверждаю, что при выполнении гранта Российского научного фонда (РНФ) № 24-29-00492 «Разработка методов оптимального целераспределения в группе подвижных робототехнических комплексов», <https://rscf.ru/project/24-29-00492/>, реализуемого на базе АО «НКБ Робототехники и систем управления» (АО «НКБ РнСУ») использованы следующие научные результаты диссертации по теме «Разработка и исследование методов решения многоагентной задачи коммивояжера» младшего научного сотрудника АО «НКБ Робототехники и систем управления» Хуссейна Фираса Аймановича:

- метод решения многоагентной задачи коммивояжера, реализующий стратегию «сначала маршрутизация, затем кластеризация» и использующий алгоритм локального поиска для улучшения качества полученного решения;
- гибридный метод решения многоагентной задачи коммивояжера, объединяющий стратегии «сначала маршрутизация, затем кластеризация» и «сначала кластеризация, затем маршрутизация», что позволяет совместить их преимущества и компенсировать недостатки.

Использование предложенных решений позволило повысить качество выполнения задачи целераспределения при разработке и исследовании сценариев группового применения робототехнических платформ (РТП).

Эффективность предложенных Хуссейном Ф.А. решений подтверждена результатами моделирования применения РТП при решении типовых задач управления группой РТП.

Анализ эффективности применения предлагаемых методов выявил существенные улучшения показателей эффективности по сравнению с другими методами решения многоагентной задачи коммивояжера, а именно:

- снижение времени программного решения многоагентной задачи коммивояжера на 95% (т.е. итоговое время составляет 5% от времени аналогов);
- повышение качества получаемого решения по критерию длины самого протяжённого маршрута среди агентов на 27%.

Руководитель гранта РНФ № 24-29-00492,
с.н.с АО «НКБ РнСУ»

(подпись)

В. А. Костюков

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024680400

Программа распределения задач в мультиагентной
системе на основе уменьшения размера пространства
решений

Правообладатель: *Хуссейн Фирас Айманович (RU)*

Автор(ы): *Хуссейн Фирас Айманович (RU)*



Заявка № 2024669753

Дата поступления 22 августа 2024 г.

Дата государственной регистрации

в Реестре программы для ЭВМ 28 августа 2024 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Документ подписан электронной подписью
Сертификат: 06924703062300154126167016012026
Владелец: **Зубов Юрий Сергеевич**
Действителен с 10.01.2014 по 03.10.2026

Ю.С. Зубов