УДК 004.89; 519.854.2 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait/1995-5499/2023/3/32-50
Поступила в редакцию 28.03.2023
Подписана в печать 30.09.2023

ПОСТРОЕНИЕ МНОГОАГЕНТНЫХ МАРШРУТОВ В СЕТИ С ИЕРАРХИЕЙ ВЕРШИН

© 2023 М. Г. Козлова✉, В. А. Лукьяненко, О. О. Макаров

Крымский федеральный университет им. В. И. Вернадского
пр-т Академика Вернадского, 4, 295007 Симферополь, Российская Федерация

Аннотация. Многоагентная задача маршрутизации типа коммивояжера является NP-трудной. Для построения приближенных решений важен учет всей имеющейся информации о структуре сети, ограничениях и целях. Выделяется класс задач (характерный для распределения ресурсов в инфраструктурных сетях), в которых введен иерархический порядок вершин. Рассматриваются прикладные задачи многоагентной маршрутизации *mTSP* на таких сетях с учетом разного уровня иерархии вершин и кластеризации сети (*HCmTSP*). Построение маршрутов *HCmTSP* согласовано с естественной кластеризацией сложной инфраструктурной сети. Приводится обзор задач, методов и алгоритмов, основанных на разных эвристиках. Показано, что в зависимости от логистических целей может быть выбран различный тип кластеризации, согласованной с *mTSP*. Сравниваются результаты вычислительного эксперимента по типам кластеризации и маршрутам. Приводятся результаты синтеза маршрутов двух уровней иерархии. Разработаны и реализованы алгоритмы построения маршрутов *mTSP* с одним центром (базой) и синтеза двухуровневых маршрутов обхода кластеров и маршрутов *TSP* на кластерах. В задаче *mTSP* с одним выделенным центром для построения начальных маршрутов исследуется гибридный алгоритм распределения вершин по кругу в зависимости от угла, с дальнейшим разбиением на кластеры и поиском *TSP* агентами на каждом кластере. Используются эвристики нескольких муравьиных колоний. В задаче синтеза базовый алгоритм имитации отжига сравнивается с решателем Concorde. Приведены численные результаты для известных тестовых наборов данных и для реальных данных городской инфраструктуры. Работа предназначена для отбора алгоритмов, их тестирования и проведения вычислительных экспериментов и наполнения программного комплекса: построение многоагентных маршрутов в сложных сетях с иерархией вершин.

Ключевые слова: задача коммивояжера *TSP* и *mTSP*, согласованная иерархическая кластеризация, алгоритмы решения *mTSP*, *HCmTSP*.

ВВЕДЕНИЕ

Рассматриваются проблемы моделирования реальных задач прикладной алгоритмической логистики с помощью многоагентных иерархических задач маршрутизации. Логистика перевозок на сложных сетях включает в себя задачи планирования транспортных маршрутов с учетом: необходимых транспортных средств; распределения грузов по потребителям (клиентам); количества перевозок; расписания (график) работы водителей (если это автомобильные перевозки); времени в пути, временных окон для погрузочных и разгрузочных работ и др. Глобальная модель содержит огромное число переменных, множество локальных и технологических критериев оптимальности, большое число ограничений и т. п. В этом случае проверка разрешимости, устойчивости, разработка и реализация алгоритмов является практически нереализуемой задачей. Необходим набор достаточно простых прикладных моделей, направленных на снижение размерности (например, с помощью декомпозиции) таких,

✉ Козлова Маргарита Геннадьевна
e-mail: kozlovamg@cfuv.ru, art-inf@mail.ru

чтобы было возможно использовать полиномиальные алгоритмы или экспоненциальные (точные) на выделенных кластерах малой размерности. Поставленной цели можно добиваться с помощью иерархической кластеризации HC (Hierarchical Clustering) и межкластерной многоагентной маршрутизацией. Существенным является использование любой доступной информации о структуре сети, прецедентных моделях, использованию мобильных и спутниковых коммуникационных технологий, а также возможность снятия некоторых ограничений для упрощения модели по управлению логистическими процессами и оптимального использования всех ресурсов в режиме реального времени.

Актуальность темы связана с большим числом прикладных задач построения многоагентных маршрутов в сложно организованных сетях [1–4]. Например, из некоторого центра необходимо доставить товар в региональные центры, где они будут перегружены и доставлены потребителям за наименьшее время или с минимальной стоимостью. Здесь сочетаются задачи обхода кластеров (кластерная задача коммивояжера) и локальные задачи коммивояжеров на каждом кластере. Иерархия вершин различного типа с выделением распределительных центров или центров доставки вносит специфику в математические модели и применяемые методы.

В работе рассматриваются решения проблемы со многими различного уровня узлами (вершинами сети) и многими агентами. Многоагентные *TSP* остаются предметом текущих исследований, на них можно пробовать проверять новые эвристические стратегии. Существуют различные приложения в задачах с большим количеством вершин (узлов). Например, задача доставки заказов, формулировка которой мотивирована проблемой комплектования заказов на складах, где продукты одного и того же типа хранятся на разных складах или в разных местах на одном и том же складе. Кроме того, различные постановки задачи могут быть преобразованы в *TSP* с большим количеством вершин (узлов), например, проблема маршрутизации транспортных средств или проблема составления расписания работы магазинов. С другой стороны, для решения задач с большим числом вершин можно использовать различные методы кластеризации, основанные на прототипах, центрах, компонентах графа и плотностях.

Для исследования иерархических *mTSP* применяются современные подходы, основанные на точных и приближенных алгоритмах, с использованием эвристик, метаэвристик, генетических алгоритмов и нейросетевого подхода, интеллектуализированной обработки больших данных.

1. КЛАСТЕРИЗАЦИЯ, СОГЛАСОВАННАЯ С МНОГОАГЕНТНОЙ ЗАДАЧЕЙ КОММИВОЯЖЕРА

Задачи коммивояжера (Travelling Salesman Problem, *TSP*) и многоагентная TSP (*mTSP*) хорошо известны в литературе и являются NP-трудными [5]. Впервые для численного решения *TSP* был предложен алгоритм Данцига, Фалкерсона и Джонсона, где использовался метод ветвей и границ (МВГ), с помощью которого было обнаружено, что среднее время вычисления слишком велико, чтобы МВГ можно было применять в реальных задачах *TSP*. Поэтому *TSP* решалась с помощью различных метаэвристик, таких как колонии муравьев *ACO*, имитация отжига *RS*, генетические алгоритмы *GA* и другие. Новые алгоритмы продолжают появляться, и их интересно использовать в классических задачах и различных обобщениях *mTSP*.

Из-за высокой сложности *TSP* не существует алгоритма глобальной точной оптимизации с полиномиальной сложностью. Например, алгоритм Хельда-Карпа приближенно решает задачу с $O(4n^2)$ сложностью. Для того, чтобы обеспечить приемлемое практическое решение реальных задач типа *TSP*, обычно используются некоторые эвристические алгоритмы.

Для решения задачи *TSP* применяются различные методы и приемы, например, алгоритм LinKernighan, предложенный в [6]. Кроме того, *GA* с кластерами *CAG* был представлен в работе [7], где авторы отмечают, что с помощью *CAG* удается найти оптимальное

решение за меньшее время, чем стандартным *GA* (алгоритм *SGA*). Также в [7] описан механизм обучения, используемый для группировки похожих объектов в кластеры, гарантируя, что, несмотря на различные доступные методы кластеризации, существует общая стратегия, которая работает одинаково для разных задач. Однако напрашивается вывод, что лучше использовать простые механизмы. В некоторых источниках предложено решение *TSP* с помощью кластеризации, например, в [8] предложен подход, названный *CTSP* (кластерная задача коммивояжера). Широко представлены исследования решения задачи *TSP* с использованием колонии муравьев, имитации отжига и генетическими алгоритмами. Некоторые авторы считают, что наилучшие результаты были получены с помощью генетических алгоритмов, что верно только для выделенных классов задач.

В задаче маршрутизации с несколькими центрами [9] строится общий маршрут с последующим разрезанием на кластеры. Иерархическая декомпозиция применена в [10]. В работе [11] метод ветвей и границ применяется в задаче с иерархией вершин (центров), правилами чередования прохождения вершины с общим местом сбора (депо). Другие модели связаны с применением в задачах мониторинга инфраструктурных сетей беспилотных летательных аппаратов — дронов (БПЛА). В многоагентных задачах для дронов существенным является их согласованное поведение и возможность взаимодействия (обмен информацией на определенном расстоянии). Исходной сети ставится в соответствие более простая сеть облета и упрощается исходная задача многих коммивояжеров (Multiple Travelling Salesman Problem, *mTSP*).

В работе [13] авторы предложили метод иерархической кластеризации, очень похожий на *CTSP*, в котором используется стратегия постепенного группирования объектов и построения структуры классификации, называемой дендрограммой. Тем не менее, качество получаемых кластеров ненадежно. Для преодоления этой проблемы применяется глобальная оптимальная стратегия построения дендрограммы, которая заключается в нахождении оптимального кругового маршрута, который минимизирует общее расстояние для посещения всех объектов вдоль ветвей дендрограммы. Построенный маршрут моделирует *TSP* и решает задачу с использованием метода переменного поиска в окрестности. Моделируемая кластерная дендрограмма, основана на информации, определяемой заявками. Авторы [14] обсудили методы кластеризации, которые могут быть использованы для обработки пространственных и временных шаблонов в большом объеме данных, что позволяет наблюдать существование различных пространственных и временных кластеров. Авторы [15] реализовали алгоритмы кластеризации для методов, применяемых в интеллектуальном анализе данных, что позволяет исследовать наборы данных, используя алгоритм *k-means* для вычисления значения стоимости маршрута, подобного *TSP*, на основе евклидова расстояния.

В [16] предложено использовать алгоритм *k-means* для решения задачи поступления данных с несколькими кластерами, генерируемыми динамически и без повторения, что сокращает время вычислений, обеспечивая более точные результаты. Поэтому первоначальная группировка выполняется на основе статистических данных с использованием *k-means*. Затем, наибольшее расстояние между центром тяжести и самой дальней точкой используется для определения следующей точки, которая находится в кластере, процесс повторяется для охвата всех данных.

Исходя из упомянутых работ, можно сделать вывод о возможности использования различных эвристик, которые позволяют решать *TSP* эффективно. Например, в [9] эвристика *NEH* (первые буквы имен авторов) предназначена для решения проблемы планирования работы магазина. Авторы [17] улучшили этот алгоритм с помощью двух методов. Во-первых, для сокращения времени вычисления разработаны и вводятся некоторые уточнения в алгоритм *NEH*. Во-вторых, правила тайм-брейка применяются для получения хороших решений. Представленные результаты моделирования показывают, что эти два метода улучшают результаты, получен-

ные в алгоритме *NEH*. Представляет интерес решение *CTSP*, применяющее комбинацию эвристики в алгоритме *NEH* [9] и модификацию метаэвристического локального поиска с многократным перезапуском *MRSILS* [18].

В [19] также предложен эвристический метод решения *CTSP*, который является обобщением *TSP*, где множество узлов делится на кластеры с целью нахождения минимальной стоимости гамильтонова цикла. Авторы [19] предложили алгоритм *ILS* решения *CTSP*, в котором генерируются случайные потомки итерационным локальным поиском. Утверждается, что эвристические методы являются конкурентоспособными при параллельном использовании программного обеспечения. Алгоритм *ILS* [20] является одним из самых популярных, использующих простую эвристику. Алгоритм *ILS* успешно применяется для обеспечения практически оптимальных решений различных задач логистики, транспортировки, производства и т. д. Так как он разработан для решения проблем в детерминированных сценариях, то он не отражает фактическую стохастическую природу реальных систем. Сочетание алгоритмов, основанных на кластеризации и различных эвристиках, более эффективно, чем простое применение генетических алгоритмов.

Близкий класс иерархических многоагентных задач типа коммивояжера с временными окнами HmTSPTW (Hierarchical multiple Traveling Salesman Problem with Time Windows) представлен авторами на международной конференции ИОИ-14 [21]. Здесь базовыми является алгоритм оптимизации с несколькими муравьиными колониями ACS (Ant Colony System), который применяется для решения задач маршрутизации транспортных средств с двумя целевыми функциями: минимизация количества транспортных средств (агентов) и минимизация общего времени в пути. Тестирование проводилось на данных Solomon [22]. Такие задачи можно отнести к задачам маршрутизации с временными окнами MVRPTW (Multi-Vehicle Routing Problem with Time Windows). Приведем несколько работ, в которых применяются близкие к данной работе подходы.

В [23] рассматривается VRPTW с известными требованиями клиентов, центральным депо и набором транспортных средств (ТС) с ограниченной вместимостью. Сводится к минимуму общее расстояние и общее время ожидания транспортных средств при сохранении пропускной способности и временных ограничений. Применяемые методы решения состоят из трех этапов: кластеризация, маршрутизация и оптимизация. Используя *k-means*, эвристику на основе метода центроид, алгоритмы кластеризации *dbscan* (Density-Based Spatial Clustering of Applications with Noise) и *SNN* (Shared Nearest Neighbor) на начальной стадии генерации популяции генетического алгоритма, клиенты разделяются на возможные кластеры. Затем для каждого кластера строятся возможные маршруты. Наконец, в качестве исходной популяции берутся возможные маршрутные решения, а для оптимизации используется генетический алгоритм. Для сравнения полученных результатов используется набор хорошо известных эталонных данных. По результатам исследования замечено, что использование алгоритма кластеризации *k-means* при генерации начальной популяции генетического алгоритма является эффективным для решаемой задачи.

В [24] для решения VRPTW применяется модифицированный алгоритм муравьиной колонии (ACO). Предложенный метод использует свой выбор клиента, чтобы уменьшить или устранить неэффективность выбора клиента, которая возникает в классическом методе ACO. Повышена производительность поиска ACO для исключения небольших маршрутов. Кроме того, предложен метод повторной инициализации для понижения размерности задачи или решения проблемы захвата в локальном оптимуме. Эксперименты на данных для пятидесяти шести типов сетей показывают, что предлагаемый метод имеет большую производительность по сравнению с другими методами ACO.

В работе [25] представлен эффективный меметический алгоритм для VRPTW большой размерности. Классический меметический алгоритм состоит из эволюционного алгоритма глобального поиска и алгоритма

локального уточнения. В работе представлен механизм переключения между глобальным и локальным поисками для улучшения производительности. Для выбора кроссовера определяются мера сходства и сигмовидная функция. Экспериментальные результаты на тестах Геринга и Хомбергера показывают, что этот алгоритм превосходит другие подходы и улучшает 34 наиболее известных решения из 180 случаев задач большой размерности.

Заметим, что согласование кластеризации сети с многоагентной спецификой задачи коммивояжера приводит в общем случае к многокритериальной *mTSP* и зависит от постановки исходной прикладной задачи. Наиболее часто в качестве требований предполагается одинаковое количество вершин в каждом кластере; близкие длины маршрутов на каждом кластере и минимальная длина маршрута обхода кластеров и др. Выбор алгоритмов кластеризации отвечает цели работы и поставленным задачам. От типа кластеризации зависит решение *mTSP* с одним или несколькими депо, особенно в случае иерархической структуры вершин и их специализации.

2. ИЕРАРХИЧЕСКАЯ МНОГОАГЕНТНАЯ ЗАДАЧА КОММИВОЯЖЕРА

Прежде чем решать задачу *mTSP* на сети с графом G различной иерархической структурой вершин, рассмотрим типичные ситуации. Часть вершин $D \subset V \subset G$ являются складами (депо), а остальные — простыми вершинами. Пусть заданы вершины верхнего уровня $D_1 \subset D$ и нижнего уровня $D_2 \subset D$, $D_1 \bigcup D_2 = D$, $D_1 \bigcap D_2 = \varnothing$. Например, из вершин D_1 доставляется ресурс в вершины D_2, откуда он перераспределяется по остальным узлам $V_1 \subset V$, $D_1 \bigcup D_2 \bigcup V_1 = V$ каким-то количеством агентов-коммивояжеров. При этом все узлы и их специфика могут быть заданы, требуется только распределить вершины V_1 между агентами для построения каждым из них замкнутых маршрутов *TSP* так, чтобы общий вес маршрутов был минимальным. Задача синтеза иерархической структуры сети состоит в поиске иерархии вершин, обеспечивающих наилучшее решение *mTSP*.

Задача построения одного замкнутого маршрута по всем вершинам сети G $(D = \varnothing)$ с помощью m агентов-коммивояжеров предполагает решение *TSP* на каждом кластере одним агентом с последующим объединением маршрутов.

Реальные задачи приводят к различным постановкам, не укладывающимся в рассмотренные варианты. Может ставиться задача об определении количества различного типа коммивояжеров, например, большегрузные транспортные средства (ТС) обеспечивают доставку грузов на узлы нижнего уровня, где они перегружаются на ТС меньшей грузоподъемности и доставляются потребителям по кольцевым (или радиальным) маршрутам на выделенных узлах (кластерах) (см. рис. 1).

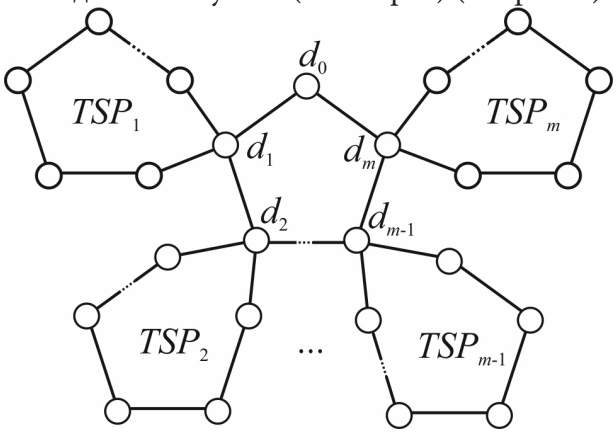


Рис. 1. Иерархическая mTSP с m кластерами, базами нижнего уровня ($d_i, i = \overline{1,m}$) и базой высшего уровня d_0

[Fig. 1. Hierarchical mTSP with m clusters, lower level bases ($d_i, i = \overline{1,m}$) and higher level bases d_0]

Рассмотренным вариантам маршрутизации соответствуют схемы включения этапа кластеризации для снижения сложности решения *mTSP* с учетом специфики вершин (узлов). В целом соответствующие алгоритмы являются эвристическими и требуют анализа их эффективности для конкретного варианта иерархической кластеризации *mTSP* (*HCmTSP*).

Рассмотрим двухуровневую логистическую сеть по доставке грузов. Предполагается наличие центрального склада (базы) d_0 или распределительного центра (РЦ), с которого

груз доставляется на региональные распределительные центры (РРЦ) d_i — склады, перевалочные пункты, с которых, в свою очередь, грузы распределяются по потребителям. Вариант такой структуры сети приведен на рис. 1. РЦ (база высшего уровня) перераспределяет грузы по РРЦ (базам нижнего уровня), которые обслуживают один или несколько кластеров потребителей. Такую задачу будем называть иерархической многоагентной задачей коммивояжера *HmTSP* (Hierarchical multiple Traveling Salesman Problem). Более сложной является задача синтеза иерархической сети такого типа, где необходимо найти РЦ и РРЦ.

Потребители объединены в кластеры, обслуживаемые РРЦ (см. рис. 1). Задача сводится к *mTSP* по обходу РЦ и РРЦ; кластеризации потребителей, согласованной с маршрутами коммивояжера на каждом кластере, привязанном к своей базе (РРЦ). Более точная постановка задачи зависит от того, какая информация является исходной.

Пусть задана транспортная сеть $S=(D,V,R)$, $D=(d_0,d_1,...,d_k)$, $V=(v_1,v_2,...,v_n)$, $R=\{r_{ij}\}$, $i,j\in I_D\bigcup I_V$ — веса (расстояния, время прохождения дуги (i,j)), m — число кластеров потребителей, I_D, I_V — индексы множеств D и V. Предлагается следующий алгоритм 1 решения иерархической кластерной многоагентной задачи коммивояжера.

Алгоритм 1. *HCmTSP*

Вход: Сеть $S=(D,V,R)$.
Выход: Маршруты доставки грузов.

1. Построить m кластеров потребителей C_j (вершин из V) с учетом их структуры и требований близости вершин.

2. На каждом кластере C_j найти решение TSP_j, $j=1,2,...,m$.

3. Сравнить маршруты TSP_j, $j=1,2,...,m$ между собой (по количеству вершин, длинам маршрутов и др.), согласно заданным условиям.

4. Уточнить кластеры C_j, перебрасывая вершины:

ЕСЛИ достигнуто близкое количество вершин или близких длин маршрутов, то перейти на шаг 2,

ИНАЧЕ если маршруты на кластерах удовлетворяют заданным требованиям, то перейти на шаг 5.

5. Для каждого кластера C_j найти ближайшую базу d_i, $i=1,2,...,k$.

6. Найти решение *TSP* на вершинах d_i, $i\in I_D=\{0,1,2,...,k\}$.

7. Передать на выход оптимальные маршруты TSP_j, $j=1,2,...,m$ и маршрут обхода кластеров.

Количество кластеров m может быть равно числу РРЦ с вершинами из D. Вершины D могут быть фиксированными или определяться. Например, d_j — вершины кластеров C_j, $j=1,2,...,m$, а d_0 ближайшая ко всем d_j, $j=1,2,...,m$.

В алгоритме *HCmTSP* критерием оптимальности считается минимальность расстояния по всем маршрутам (РЦ — РРЦ и маршрутам TSP_j, $j=1,2,...,m$).

3. РЕАЛИЗАЦИЯ АЛГОРИТМОВ ИЕРАРХИЧЕСКОЙ КЛАСТЕРИЗАЦИИ *mTSP*

Вычислительные эксперименты, проведенные в рамках поставленных задач, направлены на отбор эффективных компьютерных технологий, разработку методов и алгоритмов, ориентированных на применение в комплексе проблемно-ориентированных программ «Программный комплекс построения иерархических маршрутов *TSP* в сложных сетях большой размерности». А также для проведения расчетов по снижению размерности (сложности) исходной задачи на основе распределения агентов *TSP* по кластерам, с учетом иерархии вершин. В зависимости от исходной информации о сети для кластеризации применим весь спектр известных алгоритмов (алгоритмы максимального разреза, *k-means*, иерархической кластеризации, геометрического распределения вершин в круге по углу относительно центра масс сети и др.). В данной работе, аналогично статьям авторов [4, 26, 27], для кластеризации используется алгоритм *k-means*, согласованный с *TSP* на кластерах, а для синтеза иерархических

маршрутов на кластерах (нижний уровень) и обхода кластеров (верхний уровень) вначале используется эвристический жадный алгоритм геометрического распределения вершин в круге по углу относительно центра масс.

В разработке алгоритмов кластеризации сети для решения задачи *mTSP* использована библиотека Scikit-learn для Python, что позволяет использовать некоторые оптимальные алгоритмы кластеризации (*k-means*, *dbscan*), реализованные и «отточенные» сообществом разработчиков на протяжении долгого времени, для тестовых примеров из библиотеки TSPLIB [30].

Выбор Concorde TSP Solver для проведения численных экспериментов обусловлен тем, что это известный программный комплекс, разработанный для решения *TSP*. Этот продукт является одним из самых быстрых и эффективных решателей. Исходный код комплекса написан на языке программирования ANSI C Куком и др. [28]. В решателе используется алгоритм ветвей и границ с параллельной генерацией, тестированием серии решений-кандидатов и постепенным уточнением решения до тех пор, пока не найдено оптимальное решение. Решатель TSP в Concorde использовался для получения оптимальных решений для всех 110 экземпляров TSPLIB. Самый большой из них имеет 85 900 городов.

Кроме решения задач *TSP* с десятками тысяч городов [28] Concorde широко применяется исследователями и практиками для задач логистики и дискретной оптимизации. Concorde использовался для решения на соревнованиях по *TSP* и установил ряд мировых рекордов.

Следует отметить, что Concorde TSP Solver является эффективным и действенным инструментом для решения *TSP*. Его способность обрабатывать крупномасштабные экземпляры *TSP*, а также широкое использование исследователями и практиками делают Concorde полезным ресурсом для тех, кто работает в областях, связанных с комбинаторной оптимизацией и логистикой.

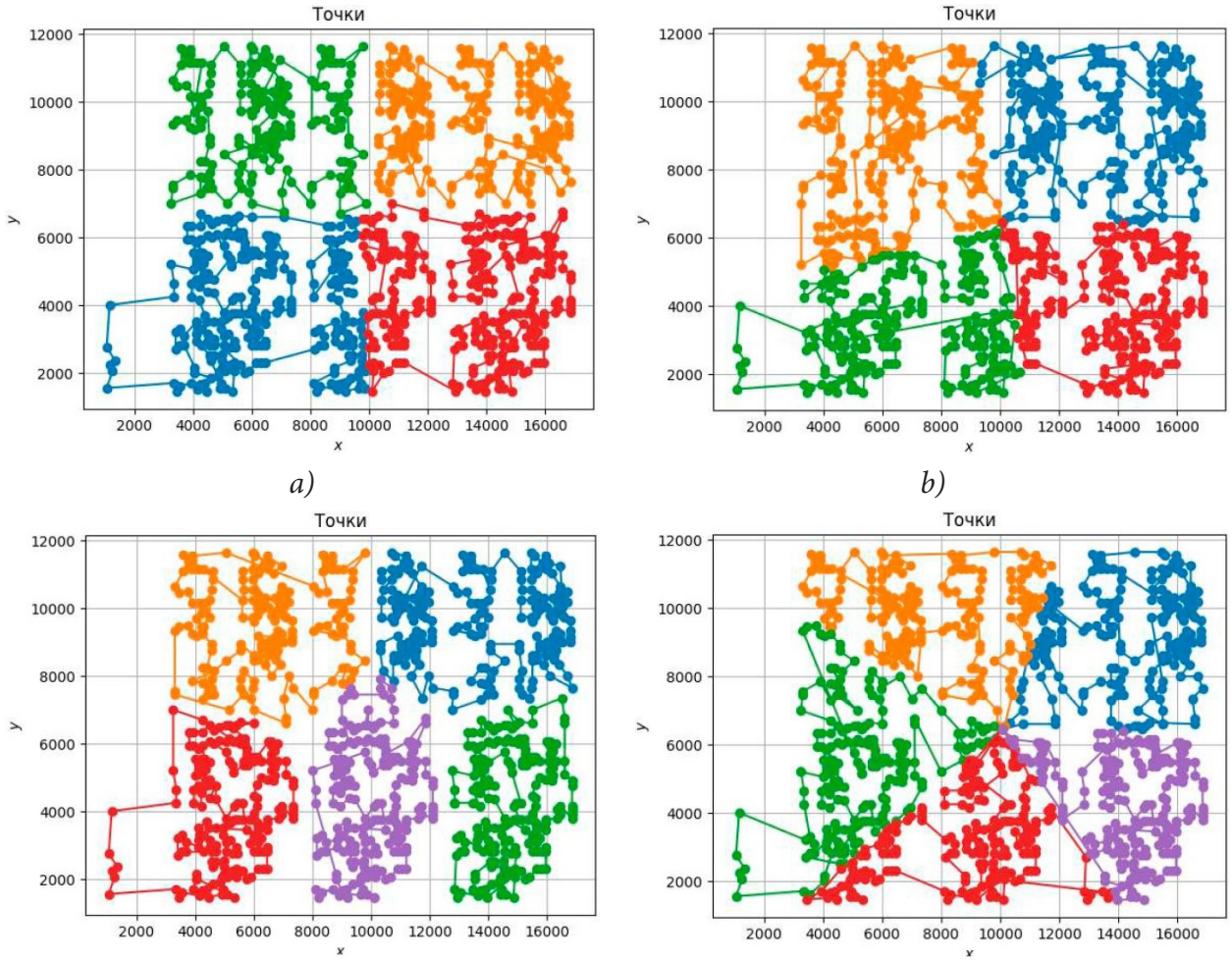
Для вычислений в работе использовался адаптер для Python PyConcorde [29]. PyConcorde — это «обертка» Python для Concorde TSP Solver. PyConcorde предоставляет интерфейс Python к решателю Concorde, позволяя пользователям легко интегрировать оптимизацию *TSP* в свои программы. Пакет PyConcorde включает инструменты для чтения экземпляров *TSP*, вызова решателя Concorde и получения решений. Эта обертка является свободной и распространяется под лицензией Modified BSD license.

В первой серии вычислительных экспериментов решается задача *HCmTSP* с одной распределительной базой d_0. Здесь приоритетом являются методы кластеризации и количество агентов. Данные факторы влияют на сложность решения задач.

В вычислительных экспериментах используется сеть (более 1000 вершин) из библиотеки TSPLIB. Проведена кластеризация для 2, 3, 4, 5 агентов различными методами. Базовыми являются *k-means* и геометрическое распределение вершин, расположенных в круге по углу.

На рис. 2 приведен один из примеров работы алгоритма нахождения маршрутов *TSP* для различных кластеризаций и различного количества агентов. На данном рисунке в левой части отмечены независимые решения *TSP* на каждом из кластеров (*a*, *c*) и решение задачи многих коммивояжеров (*mTSP*), у которых выражена база, как вершина ближайшая к центру масс (*b*, *d*). Хорошо заметны различия в применяемых алгоритмах кластеризации. На рис. 2 *a*) и *c*) можно наблюдать ярко выраженные кластеры, близкие к усредненным центрам масс каждого из кластеров (алгоритм *k-means*). В то же время на рис. 2 *b*) и *d*) кластеры похожи на сектора окружности, это связанно с применением алгоритма полярной кластеризации. На рис. 2 выбор маршрутов TSP_j, $j = 1, 2, ..., m$ на каждом кластере каждым агентом осуществляется с помощью жадных алгоритмов и уточняется с помощью локальных преобразований. Кластеры уточняются по результатам сравнения найденных маршрутов с помощью процедуры перебрасывания вершин из одного в другой кластер (согласованная с TSP_j кластеризация).

Вычислительный эксперимент подтверждает необходимость согласованной с иерархической структурой сети кластеризации с



a) *b)* *c)* *d)*

Рис. 2. Маршруты коммивояжеров для различных кластеризаций
[Fig. 2. Traveling salesmen routes for various clusterings]

целью снижения времени решения *mTSP*. На рис. 3 можно увидеть, что время построения маршрута для четырех агентов превосходит время построения маршрута для пяти или семи агентов (слово «маршрут» в легенде следует понимать как «агент», так как считается, что каждый агент строит свой независимый маршрут).

Таким образом, для построения маршрута коммивояжера в сети большой размерности разумно использовать *mTSP* и у каждого агента может быть своя стратегия поиска маршрута на локальном кластере (применение роевых, генетических алгоритмов и др.).

Объединение локальных маршрутов в один общий является самостоятельной задачей, решаемой с помощью эвристик за разумное время.

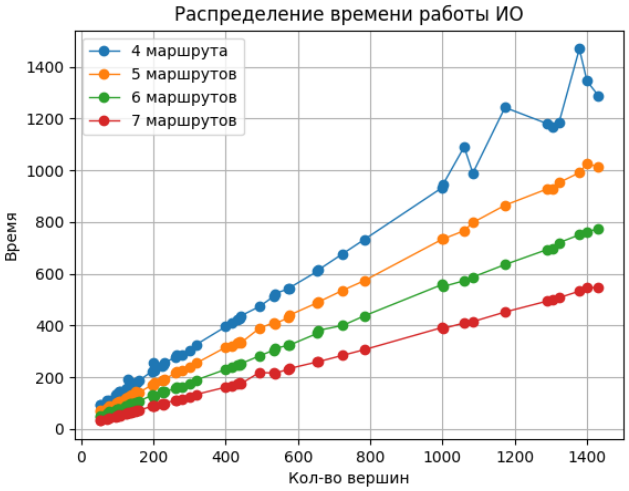


Рис.3. Зависимость времени работы от количества вершин
[Fig. 3. Dependence of operating time on the number of vertices]

4. СИНТЕЗ ИЕРАРХИЧЕСКИХ МАРШРУТОВ В МНОГОАГЕНТНОЙ ЗАДАЧЕ ТИПА КОММИВОЯЖЕРА

Выбор РЦ может определяться заданными кластерами C_j, $j=1,2,...,m$ на которых решаются задачи коммивояжера TSP_j, $j=1,2,...,m$. В предыдущем разделе статьи РЦ определялся как вершина d_0, ближайшая к центру масс всей сети. В алгоритме 1 региональные распределительные центры d_i находились как ближайшие вершины маршрутов TSP_j, $j=1,2,...,m$ к вершине d_0. А маршрут обхода вершин d_i (доставка грузов из РЦ в РРЦ) строится в виде решения задачи коммивояжера.

Приведем алгоритм построения иерархических маршрутов многоагентной задачи *TSP*, применяемый в численном эксперименте (уточнение алгоритма 1 *HCmTSP*). Задается сеть $S=(G,R)$, $G=(V,U)$, V — множество вершин, U — множество дуг, R — матрица весов.

Алгоритм 2.
Синтез иерархических маршрутов

Вход: Путь к файлу с данными для построения графа.

Выход: Список найденных оптимальных маршрутов для каждого из кластеров.

1. Считать граф G, соответствующий сети $S=(G,R)$.

2. Найти базу *center_point* как вершину ближайшую к центру масс графа G.

3. Исключить базу из списка вершин графа G.

4. Произвести разбиение вершин G по кластерам C_j, $j=1,2,...,m$, используя любой удобный алгоритм.

5. Запустить поиск решения TSP_j на каждом полученном кластере C_j, $j=1,2,...,m$.

6. Найти множество баз второго уровня *secondary_bases*, по алгоритму:

6.1. Для каждого кластера C_j, найти ближайшую точку к *center_point*.

6.2. Добавить найденные точки в *secondary_bases*.

7. Добавить *center_point* в *secondary_ bases*.

8. Запустить поиск решения TSP на множестве *secondary_bases*.

9. Передать на выход найденные оптимальные маршруты.

Далее приведем алгоритм имитации отжига, который применяется в сравнении с решателем Concorde.

Алгоритм 3. *Решение задачи TSP методом имитации отжига*

Вход: Список точек вершин полносвязного графа G.

Выход: Найденный оптимальный маршрут.

1. Задаем параметры: *T_start* — начальная, *T_finish* — конечная температуры.

2. Задаем произвольное начальное состояние S и считаем энергию для этого состояния

$$E=\sum_{i=1}^{n-1} weight_{i,i+1}+weight_{n,0},$$

$weight_{i,j}$ — расстояние от узла i до узла j.

3. Создаем лучшее состояние S^* с энергией E^*.

4. Инициализируем цикл «остывания» до тех пор, пока $T>T\_finish$.

4.1. Случайным образом генерируем новое состояние-кандидат CS, считаем его энергию CE.

4.2. Сравниваем CE с E.

4.2.1. Если $CE<E$, то переходим в состояние-кандидат и делаем его текущим, т. е. $E=CE$; $S=CS$.

4.2.2. Если $CE>E$, то переходим в это состояние с вероятностью P:

$$P_i=e^{\frac{-\Delta E}{T_i}},\ \Delta E=CE-E.$$

4.3. Сравниваем энергию текущего состояния E с энергией лучшего состояния E^*.

4.3.1. Если $E<E^*$, то заменяем лучшее состояние на текущее, т. е. $E^*=E$; $S^*=S$.

4.4. Изменяем T по правилу:

$$T_{i+1}=\frac{T_i\cdot 0.1}{i},\ T_0=T\_start.$$

5. Конец цикла «остывания».

6. На вывод лучшая энергия E^* и соответствующее состояние S^*.

Заметим, что на шаге 4.1 алгоритма 3 случайный выбор описывает общий подход к решению *TSP* данным алгоритмом. В реализации используется применение *2-opt* операторов к уже полученному решению (для первой итерации это будет состояние S шага 2). Таким образом, используется итеративное улучшение решения, полученного в самом начале, но с правилами имитации отжига. Эти правила позволяют с определенной вероятностью принимать решения, которые не являются «лучшими», тем самым не давая скатиться в локальный минимум.

Приведем результаты построения маршрутов с помощью алгоритма имитации отжига и решателя Concorde. На рис. 4 приведены результаты работы алгоритма в сравнении с Concorde.

В табл. 1 приведен сравнительный анализ работы алгоритма 2 построения иерархических маршрутов алгоритмами имитации отжига и Concorde по построению маршрутов *TSP* иерархического типа. Результаты приведены для пяти РРЦ, что соответствует разбиению сети на пять кластеров.

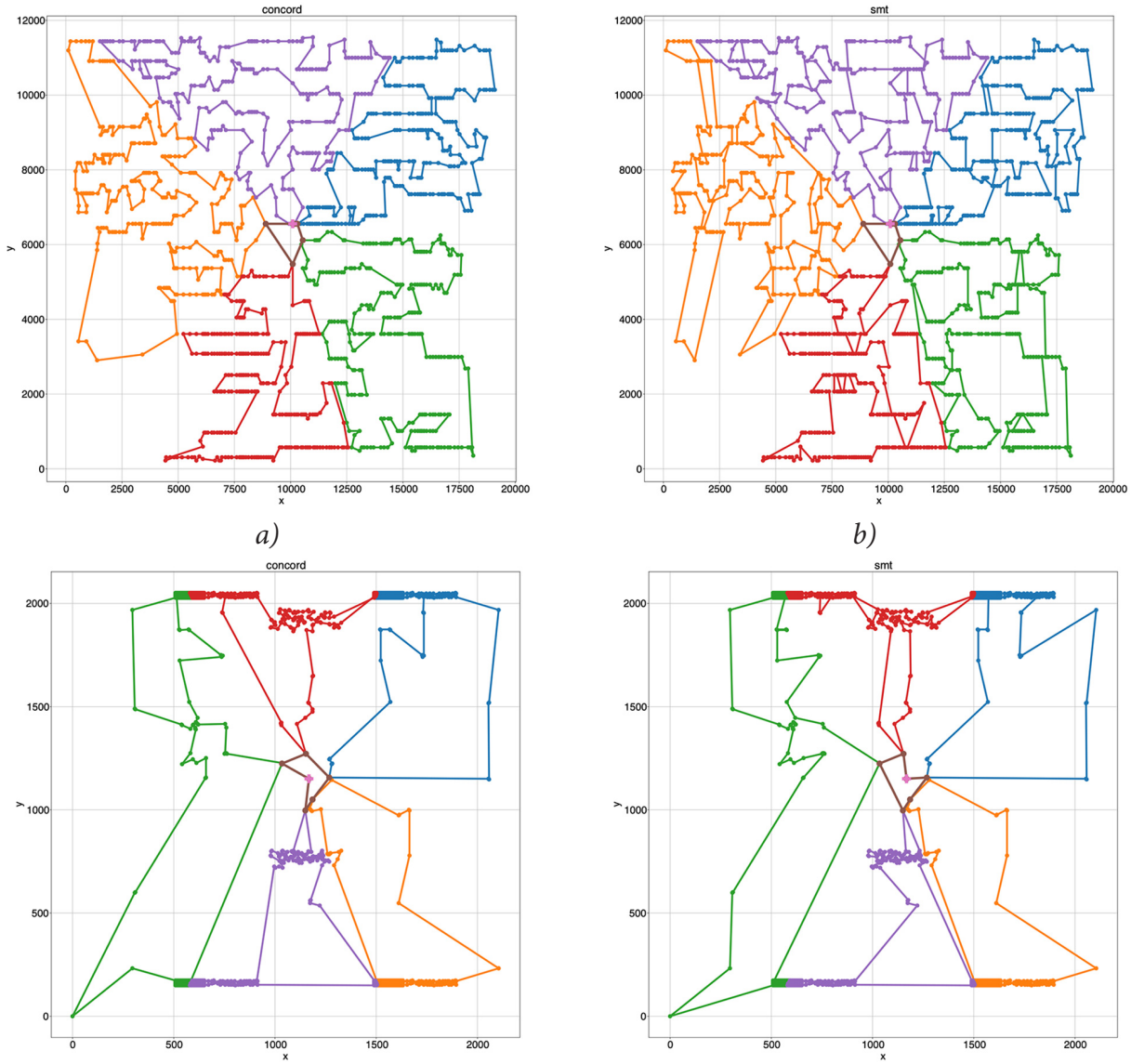
В табл. 1 приведены реальные значения, полученные для графов библиотеки TSPLIB. Основываясь на этих результатах, можно увидеть, что решатель Concorde во всех случаях превосходит метод имитации отжига по полученной длине маршрута и затраченному времени на работу. Сводная таблица представляет точные численные значения эксперимента. Таким образом, при повторении или улучшении приведенного в данной статье подхода, у исследователя будут результаты для сравнения.

Отметим, что выбор параметров в алгоритме имитации отжига в зависимости от количества вершин существенно влияет на время работы (рис. 5), по расстоянию (рис. 6) расхождение менее ощутимо. В данном случае наиболее важен параметр начальной температуры. Чем больше количество вершин, тем больше задается начальная температура. Это связанно с тем, что большое количество вершин нуждается в большем количестве итераций *2-opt* для приближения к оптимальному решению. Если кластеров много, их размерность небольшая, то возможно использование точных алгоритмов.

Также стоит отметить, что любые манипуляции с параметрами для алгоритма проводятся, только в том случае, если количество вершин в кластере превышает 15. Если вершин меньше, то рекомендуется использовать метод ветвей и границ для нахождения точного решения на кластере. Но при большой размерности графа такая ситуация практически невозможна.

Таблица 1. Сравнительный анализ построения иерархических маршрутов алгоритмами имитации отжига и Concorde
[Table 1. Comparative analysis of constructing hierarchical routes using simulated annealing and Concorde algorithms]

N	Concorde		Метод имитации отжига	
	T	*D*	*T*	*D*
439	0.922	124732	22.176	131033
1002	2.175	278390	49.218	303414
1323	24.097	292037	64.370	355572
783	1.872	9372	39.053	10427
657	1.152	52643	34.092	57165
535	1.522	2325	27.818	2490
198	0.729	20874	11.106	21374
1379	6.852	58941	67.117	69241
1400	6.706	25594	68.004	27454
318	0.821	49000	16.908	52322



a) *b)* *c)* *d)*

Рис. 4. Сравнение алгоритмов построения маршрутов для пяти РРЦ на различных графах:
1323 вершин — a) Concorde, b) метод имитации отжига,
1400 вершин — c) Concorde, d) метод имитации отжига.
Маршруты обхода РРЦ выделены в центре
[Fig. 4. Comparison of route construction algorithms for five RRCs on different graphs:
1323 vertices — a) Concorde, b) simulated annealing method,
1400 vertices — c) Concorde, d) simulated annealing method.
Routes to bypass the RRC are highlighted in the center

Заметим, что линейный рост времени работы алгоритма 2 с использованием алгоритма 3 по сравнению с Concorde (на рис. 5) связан с выбором параметров алгоритма имитации отжига. На рис. 6 расхождение по расстоянию алгоритмов Concorde и имитации отжига увеличивается с количеством вершин, но не так сильно, как растет время расчетов (рис. 5).

Выбор алгоритмов построения иерархических маршрутов зависит от наличия исходной информации о структуре сети и предъявляемых ограничений. В соответствующем комплексе программ предусматривается блок предобработки исходной информации, который включает:

1) определение метрических характеристик сети маршрутизации (висячие вершины,

мосты, точки сочленения, связные компоненты, центры и др.);

2) сравнительный анализ по гистограммам распределения весов (степеней вершин, расстояний между вершинами и др.) сложной структуры сети;

3) распознавание локальных прецедентных структур сети и др. для использования в алгоритмах реоптимизации;

4) выявление полиномиально различимых случаев.

Такой подход направлен на разработку инструментов, необходимых для выбора прикладных маршрутов в сложных сетях различной природы.

В работе [26] по нахождению многоагентных маршрутов в чрезвычайных ситуациях для реальных данных Большой Ялты исходной городской инфраструктурной сети ставится в соответствие сеть в виде полного графа, для которого расстояние между вершинами находится по прямой. Проводится согласованная с *mTSP* кластеризация, а затем осуществляется проекция найденных маршрутов сети облета на реальную городскую сеть (рис. 7).

Приведенные на рис. 8 гистограммы дают общее представление о распределении расстояний в графах. Основываясь на этих данных, выбирается диапазон, который включает в себя наибольшее количество ребер, затем на основе этого значения вычисляется параметр отдаленности для кластеров, который требую некоторые алгоритмы кластеризации, например, *dbscan*. С помощью такого подхода, удалось автоматизировать выбор оптимального параметра такого рода для алгоритмов кластеризации. Например, распределение степеней вершин подтверждает специфику инфраструктуры Большой Ялты (пути вдоль моря и гор). Таким образом, предобработка позволяет получить хорошее разбиение по умолчанию. Естественно, встречаются более сложные случаи, когда есть несколько явно преобладающих диапазонов расстояний, здесь приходится прибегать к анализу (опыту, прецедентам) и выбору оптимального параметра для конкретного графа. То есть существует несколько вариантов работы: либо брать длину каждого из преобладающих столбцов по очереди и сравнивать полученные решения, либо предоставить выбор человеку, который проанализирует конкретный случай и выберет предпочтительный вариант.

Принятый авторами подход позволил улучшить решения *mTSP* для Большой Ялты полученные в [26]. Если раньше применялась обычная полярная кластеризация или *k-means*, которые не основывались на структуре сети (в данном случае на распределении расстояний между вершинами), то в данной работе была использована кластеризация, основанная на данных предобработки, что позволило получить более оптимальные маршруты для задачи синтеза РРЦ. Это практическое приложение показывает преимущества анализа данных, полученных в процессе предобработки для улучшения решения.

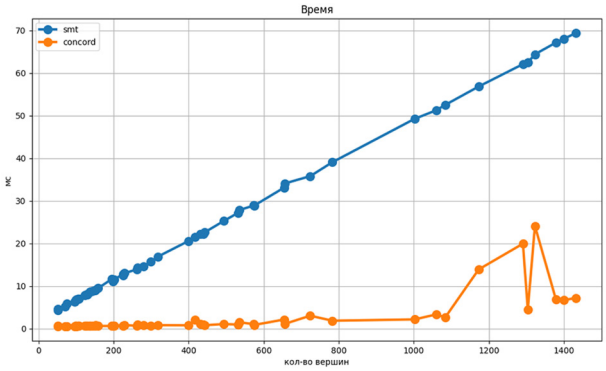


Рис. 5. Зависимость времени (мл.сек) от количества вершин
[Fig. 5. Dependence of time (ml.sec) on the number of vertices]

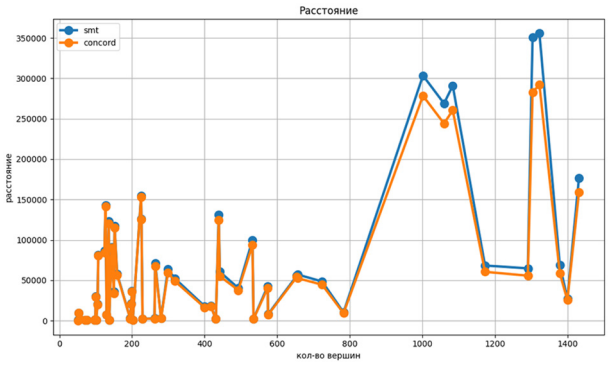


Рис. 6. Сравнение расстояния по алгоритму 2 в зависимости от количества вершин
[Fig. 6. Comparison of distance using Algorithm 2 depending on the number of vertices]

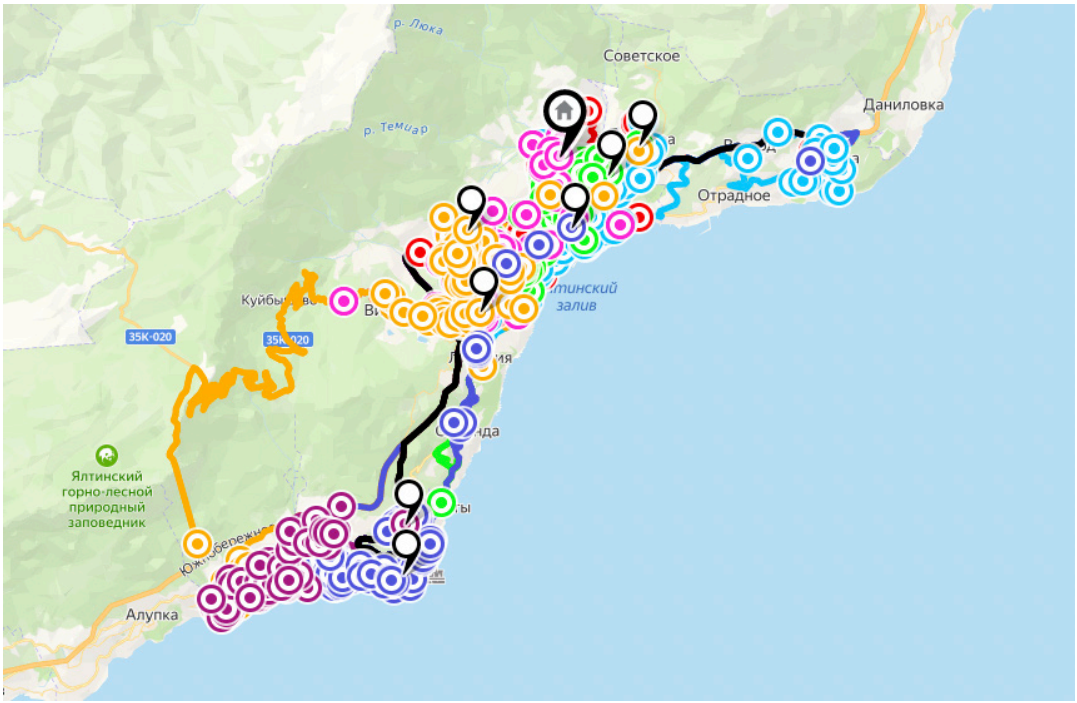
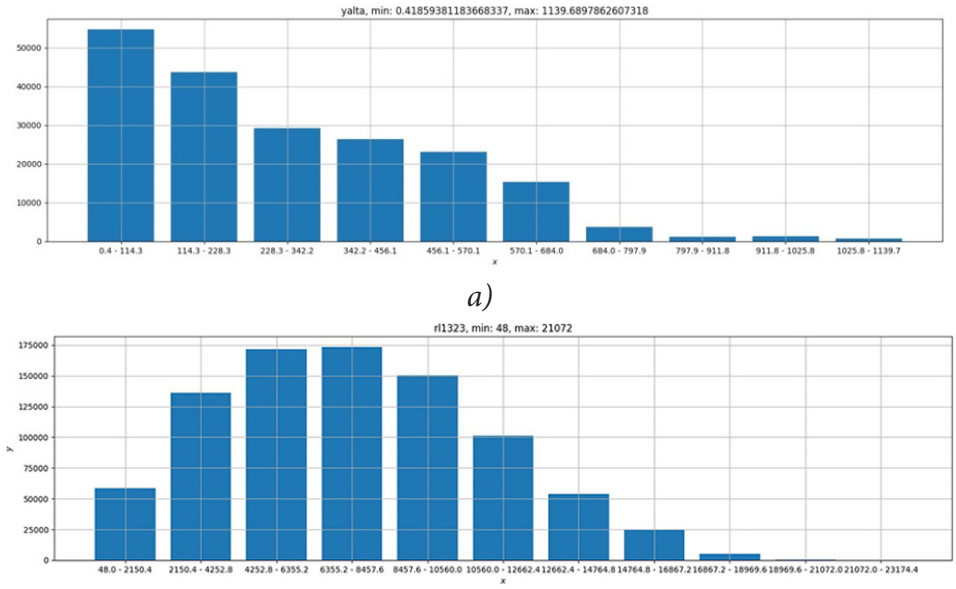


Рис. 7. Построение многоагентных маршрутов в чрезвычайных ситуациях для реальных данных по Большой Ялте
[Fig. 7. Construction of multi-agent routes in emergency situations for real data on Greater Yalta]



a)

b)

Рис. 8. Гистограмма распределения расстояний:
a) по Большой Ялте; b) на графе из TSPLIB с 1323 вершинами
[Fig. 8. Distance distribution histogram:
a) in Greater Yalta; b) on a graph from TSPLIB with 1323 vertices]

В результате проведенных экспериментов можно сделать вывод о том, что в зависимости от информации о структуре сети, иерархичности задачи *mTSP*, количества РРЦ необходимо выбирать подходящий алгоритм кластеризации, согласованной с *mTSP*.

Более подробные варианты тестирования алгоритмов для разного количества агентов

на графах с различной структурой приведены авторами по ссылке https://github.com/ssstelsss/TSP_test_solutions. По указанной ссылке в папке *data* содержатся наборы данных, на которых производилось тестирование. Эти данные могут быть использованы исследователями для тестирования полученных результатов. В папке *tsp_results* помещены найденные маршруты для данных из директории *data/tsp* с различным количеством кластеров (агентов). Папка *yalta_all_results* содержит результаты расчетов для данных по Большой Ялте с различным числом агентов (кластеров).

ЗАКЛЮЧЕНИЕ

Для задач *TSP* с большим количеством узлов многоуровневая кластеризация способствует снижению сложности решения задачи *mTSP*. В рассмотренной многоуровневой оптимизации множество узлов разбивается на кластеры или на иерархию кластеров. Для каждого кластера выполняется отдельная локальная оптимизация *TSP*, а затем локальные маршруты объединяются в глобальный маршрут с помощью решения задачи *TSP* по обходу кластеров. Процесс иерархической кластеризации может быть совмещен с поиском маршрутов коммивояжеров, уточнение маршрута при добавлении вершины в кластер может не приводить к сильному изменению построенного ранее маршрута. В зависимости от наличия одного РЦ или многих РЦ, РРЦ и структуры сети необходимо выбирать соответствующую кластеризацию. Для реальных задач учет всей имеющейся информации позволит разработать эффективный алгоритм для конкретной задачи или класса задач.

БЛАГОДАРНОСТИ

Работа поддержана Министерством науки и высшего образования Российской Федерации, соглашение № 075-02-2023-1799.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Germanchuk M. S.* Metaheuristic Algorithms for Multiagent Routing Problems / M. S. Germanchuk, D. V. Lemtyuzhnikova, V. A. Lukianenko // Automation and Remote Control. – 2021. – 10 (82). – P. 1787–1801. – DOI: 10.1134/S0005117921100155.

2. *Данильченко М. Н.* Нейросетевой подход к построению маршрута в автоматизированной системе управления специального назначения / М. Н. Данильченко, А. Б. Муравник // Наукоемкие технологии в космических исследованиях Земли. – 2021. – Т. 13, № 1. – С. 58–66. DOI: 10.36724/2409-5419-2021-13-1-58-66.

3. *Козлова М. Г.* Использование технологий искусственного интеллекта в многоагентных задачах управления / М. Г. Козлова, В. А. Лукьяненко, Л. И. Руденко, М. С. Германчук // Дистанционные образовательные технологии: сборник трудов VI Международной научно-практической конференции / отв. ред. В. Н. Таран. – Симферополь, ИТ «АРИАЛ», 2021. – С. 251–255.

4. *Лукьяненко В. А.* Специфика задач маршрутизации в условиях локальных преобразований сети / В. А. Лукьяненко, М. С. Германчук, О. О. Макаров // Математические методы распознавания образов: Тезисы докладов 20-й Всероссийской конференции с международным участием, г. Москва 2021 г. – М. : Российская академия наук, 2021. – С. 460–462.

5. *Кормен Т.* Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 2001. – 960 с.

6. *Karapetyan D.* LinKernighan heuristic adaptations for the generalized traveling salesman problem / D. Karapetyan, G. Gutin // European Journal of Operational Research. – 2011. – V. 208, No. 3. – P. 221–232.

7. *Sivaraj R.* Solving Traveling Salesman Problem using Clustering Genetic Algorithm / R. Sivaraj, T. Ravichandran, R. Devipriya // International Journal on Computer Science and Engineering. – 2012. – V. 4, No. 7. – P. 1310–1317.

8. *Phienthrakul T.* Clustering Evolutionary Computation for Solving Traveling Salesman

Problem / T. Phienthrakul T // International Journal of Advanced Computer Science and Information Technology. – 2014. – V. 3, No 3. – P. 243–262.

9. *Nawaz M.* A heuristic algorithm for the m-Machine, n-Job flow-shop sequencing problem / M. Nawaz, E. Enscore, I. Ham // Omega-International Journal of Management Science. – 1983. – V. 11. – P. 91–95.

10. *Tang K.* Scalable Approach to Capacitated Arc Routing Problems Based on Hierarchical Decomposition / K. Tang, J. Wang, X. Li, X. Yao // Research of Birmingham. – University of Birmingham. – 2015. – P. 15. DOI: 10.1109/TCYB.2016.2590558.

11. *Медведев С. Н.* Математическая модель и алгоритм решения задачи маршрутизации транспортных средств с несколькими центрами с чередованием и единым местом сбора // Вестник ВГУ, Серия: Системный анализ и информационные технологии. – 2021. – № 1. – С. 21–32. DOI: https://doi.org/10.17308/sait.2021.1/3368

12. *Amberg A.* Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees / A. Amberg, W. Domschke, S. Voss // European Journal of Operational Reseach. – 2000. – V. 124. – P. 360–376. DOI: 10.1016/S0377-2217(99)00170-8.

13. *Tsai C.* A VNS based Hierarchical Clustering Method / C. Tsai, C. Chiu // International Conference on Computational Intelligence, 2006.

14. *Nagy M.* Using clustering software for exploring spatial and temporal patterns in non-communicable diseases / M. Nagy, D. Negru // European Scientific Journal. – 2014. – V. 10, No 33. – P. 3747.

15. *Vishnupriya N.* Data Clustering using MapReduce for Multidimensional Datasets / N. Vishnupriya, F. Sagayaraj // International Advanced Research Journal in Science, Engineering and Technology. – 2015. – V. 2, No 8. – P. 39–42. DOI: 10.17148/IARJSET.2015.2810

16. *Nidhi S.* A modified Approach for Incremental k-Means Clustering Algorithm / S. Nidhi // International Journal of Engineering Development and Research. – 2015. – V. 3, No 2. – P. 1081–1084.

17. *Liu G.* Two Techniques to Improve the NEH Algorithm for Flow-Shop Scheduling Problems / G. Liu, S. Song, C. Wu // Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence of the series Lecture Notes in Computer Science. – 2011. – P. 41–48. DOI: 10.1007/978-3-642-25944-9_6

18. *Seck-Touh-Mora J.* Improving a multi restart local search algorithm by permutation matrices and sorted work times for the flow shop scheduling problem / J. Seck-Touh-Mora, L. Garcia-Lechuga, J. Medina-Marin // World Comp Proceedings, 2014. – URL: http://worldcomp-proceedings.com/proc/p2014/GEM2351.pdf

19. *Mestria M.* Heuristic methods using variable neighborhood random local search for the clustered traveling salesman problem / M. Mestria // Revista Cientifica y Electronica de ingenierı´a de produccion. – 2014. – P. 1511–1536.

20. *Grasas A.* SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization / A. Grasas, A. Juan, H. Lorenzo // Journal of Simulation. – 2014. DOI: 10.1057/jos.2014.25

21. *Kozlova M.* Multiple hierarchical routing with time windows / M. Kozlova, V. Lukianenko, O. Makarov // Интеллектуализация обработки информации: Тезисы докладов 14-й Международной конференции, г. Москва 2022 г. – М.: Российская академия наук, 2022. – С. 430–432.

22. Solomon benchmark. – [Электронный ресурс]. – Режим доступа: https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark – Свободный (дата обращения 30.05.2022)

23. *Gocken T.* Comparison of different clustering algorithms via genetic algorithm for VRPTW / T. Gocken, M. Yaktubay // International Journal of Simulation Modelling. – 2019. – V. 18, № 4. – P. 574–585. DOI: 10.2507/IJSIMM18(4)485.

24. *Kosolsombat S.* Modified ant colony optimization with selecting and elimination customer and re-initialization for VRPTW / S. Kosolsombat, C. Ratanavilisagul // Bulletin of Electrical Engineering and Informatics. – 2022. – V. 11. – № 6. – P. 3471–3482. DOI: 10.11591/eei.v11i6.3943

25. *Zhang Y. [et al.]* Balancing Exploration and Exploitation in the Memetic Algorithm via a Switching Mechanism for the Large-Scale VRPTW // International Conference on Database

Systems for Advanced Applications. – Springer, Cham, 2020. – P. 324–331. DOI: 10.1007/978-3-030-59410-7_23

26. *Германчук М. С.* Задачи типа многих коммивояжеров в изменяющихся условиях / М. С. Германчук, М. Г. Козлова, В. А. Лукьяненко, О. О. Макаров // Сборник материалов международной конференции КРОМШ-2020: ПОЛИПРИНТ, 2020. – С. 241–245.

27. *Германчук М.* С. Задачи маршрутизации в чрезвычайных условиях / М. С. Германчук, М. Г. Козлова, В. А. Лукьяненко // Анализ, моделирование, управление, развитие социально-экономических систем: сборник научных трудов XIV Всероссийской с международным участием школы-симпозиума АМУР-2020, Симферополь-Судак, 14–27 сентября 2020 / ред. совет: А. В. Сигал (предс.) и др. – Симферополь : ИП Корниенко А. А., 2020. – С. 98–107.

28. Concorde TSP Solver. – [Электронный ресурс]. – Режим доступа: https://www.math.uwaterloo.ca/tsp/concorde.html – Свободный (дата обращения 31.01.2023)

29. PyConcorde – [Электронный ресурс]. – Режим доступа: https://github.com/jvkersch/pyconcorde – Свободный (дата обращения 31.01.2023)

30. TSPLIB – [Электронный ресурс]. – Режим доступа: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/ – Свободный (дата обращения 31.06.2022).

Козлова Маргарита Геннадьевна — канд. физ.-мат. наук, доц., доцент кафедры информатики Физико-технического института ФГАУО ВО «Крымский федеральный университет им. В. И. Вернадского».
E-mail: kozlovamg@cfuv.ru, art-inf@mail.ru
ORCID iD: https://orcid.org/0000-0002-7467-8389

Лукьяненко Владимир Андреевич — канд. физ.-мат. наук, доц., доцент кафедры математического анализа Физико-технического института ФГАУО ВО «Крымский федеральный университет им. В. И. Вернадского».
E-mail: lukyanenkova@cfuv.ru, art-inf@yandex.ru
ORCID iD: https://orcid.org/0000-0001-5271-031X

Макаров Олег Олегович — ассистент кафедры информатики Физико-технического института ФГАУО ВО «Крымский федеральный университет им. В. И. Вернадского».
E-mail: fantom2.00@mail.ru
ORCID iD: https://orcid.org/0000-0002-0502-631X

DOI: https://doi.org/10.17308/sait/1995-5499/2023/3/32-50 ISSN 1995-5499
Received 28.03.2023
Accepted 30.09.2023

MULTI-AGENT ROUTE PLANNING IN HIERARCHICAL NETWORKS

© 2023 M. G. Kozlova✉, V. A. Lukianenko, O. O. Makarov

V. I. Vernadsky Crimea Federal University, Institute of Physics and Technology
4, Vernadsky Avenue, 295007 Simferopol, Russian Federation

Annotation. The article presents a study on the multi-agent routing problem of the traveling salesman type, which is known to be *NP*-hard. To construct approximate solutions, it is essential to consider all available information about the network structure, constraints, and goals. In this regard, the research identifies a class of problems characterized by a hierarchical order of nodes, which is typical for resource allocation in infrastructure networks. The study focuses on applied problems of multi-agent *mTSP* routing in such networks, considering different levels of vertex hierarchy and network clustering (*HCmTSP*). The paper emphasizes the significance of constructing *HCmTSP* routes that are consistent with the natural clustering of complex infrastructure networks. An overview of tasks, methods, and algorithms based on different heuristics is provided. The analysis indicates that the choice of clustering consistent with *mTSP* depends on logistic goals, and the paper presents computational experiment results for clustering types and routes. Furthermore, the research offers synthesis results for routes of two hierarchy levels and develops and implements algorithms for constructing single-center *mTSP* routes and the synthesis of two-level cluster avoidance routes and *TSP* routes on clusters. In the *mTSP* problem with a single dedicated center, the paper investigates a hybrid algorithm for distributing vertices in a circle, followed by partitioning into clusters and searching for *TSP* by agents on each cluster, using several ant colony heuristics. For the synthesis problem, the basic annealing simulation algorithm is compared with the *Concorde* solver, and numerical results are provided for known test datasets and real urban infrastructure data. The research is aimed at aiding in algorithm selection, testing, and computational experiments, as well as contributing to the development of a software package for constructing multi-agent routes in complex networks with a hierarchy of vertices.
Keywords: Multi-agent routing problem, Traveling salesman problem (*TSP*), Multiple Traveling salesman problem (*mTSP*), consistent hierarchical clustering, mTSP solution algorithms, Hierarchical Clustering Multiple Traveling salesman problem (*HCmTSP*), algorithm selection.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCE

1. *Germanchuk M. S., Lemtyuzhnikova D. V. and Lukianenko V. A.* (2021) Metaheuristic Algorithms for Multiagent Routing Problems. *Automation and Remote Control*. 10 (82). P. 1787–1801. DOI: 10.1134/S0005117921100155.

2. *Danilchenko M. N. and Muravnik A. B.* (2021) Neural network approach to route construction in an automated control system for special purposes. *High-tech technologies in space research of the Earth*. 13 (1). P. 58–66. DOI: 10.36724/2409-5419-2021-13-1-58-66. (In Russian)

3. *Kozlova M. G., Lukyanenko V. A., Rudenko L. I. and Germanchuk M. S.* (2021) The use of artificial intelligence technologies in multi-agent control tasks. *Distance educational technologies: Proceedings of the VI International Scientific and Practical Conference* / ed. V. N. Taran. Simferopol, IT «ARIAL». P. 251–255. (In Russian)

4. *Lukyanenko V. A., Germanchuk M. S. and Makarov O. O.* (2021) Specifics of routing tasks in conditions of local network transformations. *Mathematical methods of pattern recognition: Abstracts of the 20th All-Russian Conference with International Participation*, Moscow 2021 – Mos-

✉ Kozlova Margarita G.
e-mail: kozlovamg@cfuv.ru, art-inf@mail.ru

cow : Russian Academy of Sciences. P. 460–462. (In Russian)

5. *Cormen T.* (2009) Introduction to Algorithms. ISBN 978-0-262-03384-8

6. *Karapetyan D. and Gutin G.* (2011) LinKernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*. V. 208, No. 3. P. 221–232.

7. *Sivaraj R., Ravichandran T. and Devipriya R.* (2012) Solving Traveling Salesman Problem using Clustering Genetic Algorithm. *International Journal on Computer Science and Engineering*. V. 4, No. 7. P. 1310–1317.

8. *Phienthrakul T.* (2014) Clustering Evolutionary Computation for Solving Traveling Salesman Problem. *International Journal of Advanced Computer Science and Information Technology*. V. 3, No 3. P. 243–262.

9. *Nawaz M., Enscore E. and Ham I.* (1983) A heuristic algorithm for the m-Machine, n-Job flow-shop sequencing problem. *Omega-International Journal of Management Science*. V. 11. P. 91–95.

10. *Tang K., Wang J., Li X. and Yao X.* (2015) Scalable Approach to Capacitated Arc Routing Problems Based on Hierarchical Decomposition. *Research of Birmingham*. University of Birmingham. P. 15. DOI: 10.1109/TCYB.2016.2590558.

11. *Medvedev S. N.* (2021) Mathematical model and algorithm for solving the problem of routing vehicles with multiple centers with alternation and a single collection point. *Vestnik VSU, Series: System analysis and information technologies*. 1. P. 21–32. DOI: https://doi.org/10.17308/sait.2021.1/3368 (In Russian)

12. *Amberg A., Domschke W. and Voss S.* (2000) Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees. *European Journal of Operational Reseach*. V. 124. P. 360–376. DOI: 10.1016/S0377-2217(99)00170-8.

13. *Tsai C. and Chiu C.* (2006) A VNS based Hierarchical Clustering Method. *International Conference on Computational Intelligence*.

14. *Nagy M. and Negru D.* (2014) Using clustering software for exploring spatial and temporal patterns in non-communicable diseases. European Scientific Journal. V. 10, No. 33. P. 3747.

15. *Vishnupriya N. and Sagayaraj F.* (2015) Data Clustering using MapReduce for Multidimensional Datasets. *International Advanced Research Journal in Science, Engineering and Technology*. V. 2, No. 8. P. 39–42. DOI: 10.17148/IARJSET.2015.2810

16. *Nidhi S.* (2015) A modified Approach for Incremental k-Means Clustering Algorithm. *International Journal of Engineering Development and Research*. V. 3, No. 2. P. 1081–1084.

17. *Liu G., Song S. and Wu C.* (2011) Two Techniques to Improve the NEH Algorithm for Flow-Shop Scheduling Problems. *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence of the series Lecture Notes in Computer Science*. P. 41–48. DOI: 10.1007/978-3-642-25944-9_6

18. *Seck-Touh-Mora J., Garcia-Lechuga L. and Medina-Marin J.* (2014) Improving a multi restart local search algorithm by permutation matrices and sorted work times for the flow shop scheduling problem. *World Comp Proceedings*. Available at: http://worldcomp-proceedings.com/proc/p2014/GEM2351.pdf

19. *Mestria M.* (2014) Heuristic methods using variable neighborhood random local search for the clustered traveling salesman problem. *Revista Cientifica y Electronica de ingenierı´a de produccion*. P. 1511–1536.

20. *Grasas A., Juan A. and Lorenzo H.* (2014) SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*. DOI: 10.1057/jos.2014.25

21. *Kozlova M., Lukianenko V. and Makarov O.* (2022) Multiple hierarchical routing with time windows. Интеллектуализация обработки информации: Тезисы докладов 14-й Международной конференции, г. Москва 2022 г. – М.: Российская академия наук, P. 430–432.

22. Solomon benchmark. Available at: https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark (accessed: 30.05.2022)

23. *Gocken T. and Yaktubay M.* (2019) Comparison of different clustering algorithms via genetic algorithm for VRPTW. *International Journal of Simulation Modelling*. V. 18. No 4. P. 574–585. DOI: 10.2507/IJSIMM18(4)485.

24. *Kosolsombat S. and Ratanavilisagul C.* (2022) Modified ant colony optimization with selecting and elimination customer and re-initialization for VRPTW. *Bulletin of Electrical Engineering and Informatics*. V. 11., No 6. P. 3471–3482. DOI: 10.11591/eei.v11i6.3943

25. *Zhang Y. [et al.]* (2020) Balancing Exploration and Exploitation in the Memetic Algorithm via a Switching Mechanism for the Large-Scale VRPTW. *International Conference on Database Systems for Advanced Applications*. Springer, Cham. P. 324–331. DOI: 10.1007/978-3-030-59410-7_23

26. *Germanchuk M. S., Kozlova M. G., Lukyanenko V. A. and Makarov O. O.* (2020) Tasks of the type of many traveling salesmen in changing conditions. *Collection of materials of the international conference KROMSH-2020: POLYPRINT*, 2020. P. 241–245. (In Russian)

27. *Germanchuk M. S., Kozlova M. G. and Lukyanenko V. A.* (2020) Routing tasks in emergency conditions. *Analysis, modeling, management, development of socio-economic systems: collection of scientific papers of the XIV All-Russian School-Symposium with international participation AMUR-2020*, Simferopol-Sudak, September 14–27, 2020 / ed. Council: A.V. Sigal (pres.) and others. Simferopol : IP Kornienko A.A. P. 98–107. (In Russian)

28. Concorde TSP Solver. Available at: https://www.math.uwaterloo.ca/tsp/concorde.html

29. PyConcorde. Available at: https://github.com/jvkersch/pyconcorde

30. TSPLIB. Available at: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

Kozlova Margarita G. — PhD in Physics and Mathematics, Associate Professor of Computer Science Department, V. I. Vernadsky Crimea Federal University.
E-mail: kozlovamg@cfuv.ru, art-inf@mail.ru
ORCID iD: https://orcid.org/0000-0002-7467-8389

Lukianenko Vladimir A. — PhD in Physics and Mathematics, Associate Professor of Mathematical Analysis Department, V. I. Vernadsky Crimea Federal University
E-mail: lukyanenkova@cfuv.ru, art-inf@yandex.ru
ORCID iD: https://orcid.org/0000-0001-5271-031X

Makarov Oleg O. — assistant of Computer Science Department, V. I. Vernadsky Crimea Federal University
E-mail: fantom2.00@mail.ru
ORCID iD: https://orcid.org/0000-0002-0502-631X