

ПОИСК ТОЧНОГО РЕШЕНИЯ ЗАДАЧИ НЕСКОЛЬКИХ КОММИВОЯЖЕРОВ

О. А. Клименко¹ [0009-0000-5236-2415], Б. Я. Штейнберг² [0000-0001-8146-0479]

^{1,2}Южный федеральный университет, г. Ростов-на-Дону, Россия

¹oleg.klimenko.math@mail.ru, ²borsteinb@mail.ru

Аннотация

В работе рассмотрена задача нескольких коммивояжеров. Она состоит в том, чтобы на графе со взвешенными дугами найти набор из заранее заданного количества непересекающихся циклов, у которого сумма весов дуг наибольшего цикла будет минимальной. Разработан точный алгоритм решения поставленной задачи, основанный на методе ветвей и границ. В построенном алгоритме, как и в известном алгоритме Балаша – Кристофидеса решения задачи одного коммивояжера, использован венгерский алгоритм решения задачи о назначениях. Представлены результаты численных экспериментов со случайными графами большой размерности.

Ключевые слова: задача коммивояжера, задача о назначениях, венгерский алгоритм, метод ветвей и границ, графы, дискретная оптимизация, гамильтонов цикл.

ВВЕДЕНИЕ

Задача нескольких коммивояжеров является обобщением задачи коммивояжера, решению которой посвящено множество публикаций. Задача коммивояжера (замкнутая) — это задача нахождения гамильтонова цикла с минимальной суммой дуг в полном взвешенном ориентированном графе, стоимости дуг которого неотрицательны.

Задача коммивояжера относится к классу NP-трудных задач [1]. Из этого факта многие исследователи делают вывод, что точные алгоритмы решения этой задачи могут работать только для небольших размерностей. Это было справедливо 50 лет назад, когда компьютеры обладали в разы меньшей вычис-

лительной мощностью. На современных компьютерах для имеющих прикладное значение размерностей точное решение задачи коммивояжера можно получать в разумное время.

В основе некоторых эффективных алгоритмов лежит алгоритм Балаша – Кристофидеса [2] – точный алгоритм решения задачи коммивояжера, основанный на методе ветвей и границ (слово «точный» означает, что алгоритм всегда находит лучшее решение). Этот алгоритм на компьютере CDC-7600 в 1979 г. находил точное решение для графов размерности 325 вершин со случайными целыми весами из сегмента $[0, 100)$ за 82 с (в худшем случае).

Эффективная реализация, адаптированная к современным процессорам, представлена в [3–5]. Здесь, в частности, применялись оптимизация использования памяти и распараллеливание.

Обход дерева вариантов в методе ветвей и границ можно описать рекурсивной функцией. Распараллеливание рекурсивных функций рассмотрено в работах [6, 7].

В [8] предложен полиномиальный эвристический алгоритм решения задачи коммивояжера с оценкой качества найденного решения. Вычислительная сложность этого алгоритма $O(N^3)$, а найденное им решение хуже оптимального не более чем в 1.5 раза. В работе [4] описан алгоритм с оценкой сложности «найденное решение хуже оптимального не более чем в k раз (k – параметр)». Этот алгоритм для случайных графов размерности 50 тыс. вершин на компьютере с шестиядерным процессором находит решение задачи коммивояжера, которое хуже оптимального не более чем на 5% ($k = 1.05$). Некоторым другим эвристическим алгоритмам решения задачи коммивояжера посвящена работа [9].

Известны различные задачи нескольких коммивояжеров. В них требуется отыскать несколько циклов, покрывающих все вершины графа ровно один раз, т. е. найти набор оптимальных маршрутов для нескольких коммивояжеров. Формулировки этих задач отличаются выбором целевой функции, точек начала движения коммивояжеров или требованием замкнутости (разомкнутости) маршрутов (т. е. должен ли коммивояжер вернуться в ту точку, из которой вышел). Обзор таких работ представлен в работе [10].

Имеется множество работ, посвященных решению задачи нескольких коммивояжеров, в которых в качестве целевой функции выбирается сумма стоимостей всех маршрутов. В [11] и [12] рассмотрено точное решение при небольшом количестве коммивояжеров (не более 10). Имеются и недавние результаты [13]. В применении к оптимизации использования транспорта такая формулировка минимизирует суммарную длину (стоимость) пройденного пути, и может так случиться, что решением окажется один цикл: один коммивояжер обходит все вершины графа, а остальные не работают.

Более интересна для применения к транспортным задачам такая формулировка, при которой рабочий день коммивояжера (водителя транспортного средства) ограничен.

В настоящей работе в качестве целевой функции выбран максимум из длин путей, пройденных коммивояжерами. В такой постановке задача нескольких коммивояжеров имеет практические применения, например, если маршруты «коммивояжеров» следует ограничивать длительностью рабочего дня или запасом горючего. Для решения задачи использован венгерский алгоритм решения вспомогательной задачи о назначениях, как и в известном алгоритме Балаша–Кристофидеса [2].

Приведены результаты численных экспериментов со случайными графами, демонстрирующие эффективность представленного алгоритма. В отличие от программы Балаша – Кристофидеса веса случайных графов взяты, как и в [3–5], как случайные числа промежутка $[0, 1000000]$. Это существенно при рассмотрении графов большой размерности.

ФОРМУЛИРОВКА ЗАДАЧИ НЕСКОЛЬКИХ КОММИВОЯЖЕРОВ

Дан ориентированный граф $G = (V, w)$, где V – множество вершин, $w : V \times V \rightarrow \mathbb{R}^+$ – весовая функция дуг (\mathbb{R}^+ – множество действительных неотрицательных чисел). Набор k простых циклов c_1, c_2, \dots, c_k такой, что каждая вершина принадлежит ровно одному циклу, назовем *маршрутами k коммивояжеров*. Цикл c_i будем называть путем (или маршрутом) i -го коммивояжера. *Стоимостью набора маршрутов k коммивояжеров* назовем величину

$$\max_{i=1..k} d(c_i),$$

где $d(c_i)$ – стоимость маршрута i -го коммивояжера (сумма весов дуг).

Задача нескольких коммивояжеров. Даны ориентированный граф $G = (V, w)$ и натуральное число k . Необходимо отыскать набор маршрутов k коммивояжеров наименьшей стоимости.

ПОИСК РЕШЕНИЯ ЗАДАЧИ НЕСКОЛЬКИХ КОММИВОЯЖЕРОВ

Будем искать точное решение задачи нескольких коммивояжеров методом ветвей и границ. В этом методе исходная задача рекурсивно разбивается на другие задачи, которые решаются проще (например, меньшей размерности или с другой оценкой целевой функции). Рекурсивные вызовы алгоритма решения задачи образуют дерево вариантов, лежащее в основе метода ветвей и границ. В ходе работы алгоритма ветвей и границ встречаются различные наборы из k циклов, покрывающих все вершины, и запоминается лучший из них. Такой набор называется *рекордным*, а значение целевой функции рекордного набора назовем *рекордным значением*.

Для каждой подзадачи вычисляется *нижняя оценка целевой функции* на основе решения задачи о назначениях (по аналогии с алгоритмом Балаша – Кристофидеса для задачи одного коммивояжера). Нижняя оценка подзадачи не превосходит оптимального значения целевой функции этой подзадачи.

Если нижняя оценка для подзадачи превосходит текущее рекордное значение, то решение подзадачи не улучшит его, и подзадачу можно отбросить, сократив обход дерева вариантов и, следовательно, время работы алгоритма.

СТРУКТУРА ПОДЗАДАЧ

На очередной итерации алгоритм работает с подзадачей, характеризующейся следующими ограничениями.

1. Ребра $(v_1, u_1), (v_2, u_2), \dots, (v_t, u_t)$ заморожены. Это означает, что каждое из перечисленных ребер должно содержаться в каком-либо из искомым циклов c_1, c_2, \dots, c_k . Гарантируется, что в каждую вершину входит не более чем одно замороженное ребро, а также выходит не более чем одно замороженное ребро. Замороженные ребра могут образовывать циклы. Такие

циклы также будем называть *замороженными*, остальные циклы в графе – *незамороженными* (хотя они могут содержать замороженные ребра). Замороженные циклы соответствуют маршруту некоторого коммивояжера.

2. Ребра $(v'_1, u'_1), (v'_2, u'_2), \dots, (v'_l, u'_l)$ запрещены. Это означает, что каждое из перечисленных ребер не содержится ни в одном из искомым циклов C_1, C_2, \dots, C_k .

Таким образом, алгоритм ищет оптимальное решение с учетом отмеченных выше ограничений.

Для решения подзадачи мы разбиваем ее на подзадачи следующего уровня, накладывая на каждую из дочерних подзадач дополнительные ограничения (новые замороженные или запрещенные ребра). Исходная задача не имеет ограничений (т. е. все ребра не запрещены и не заморожены).

Подзадача решается непосредственно в двух случаях.

1. Из каждой вершины исходит ровно одно замороженное ребро. В этом случае все циклы являются замороженными. Если их ровно k , то замороженные циклы образуют единственное решение подзадачи. В противном случае решений нет.

2. Если невозможно разбить граф на циклы, в каждом из которых нет ни одного запрещенного ребра, то решений нет.

В оригинальном алгоритме Балаша – Кристофидеса (как и в алгоритме Литтла) используется «стягивание ребра». В этом случае пара вершин, связанных этим ребром, заменяется на одну вершину, а само ребро добавляется к будущему решению исходной задачи. Чтобы не модифицировать исходный граф в задаче, мы «замораживаем ребро». Это позволяет экономить память и вычисления, связанные с переписыванием матрицы весов графа.

ВЕТВЛЕНИЕ В МЕТОДЕ ВЕТВЕЙ И ГРАНИЦ

Разбиение на подзадачи происходит следующим образом:

1. Матрица весов исходного графа рассматривается как матрица весов двудольного графа: вершины одной доли – это строки исходной матрицы, а вершины другой доли – столбцы. Для матрицы весов графа решается задача о назначениях (AP) с помощью венгерского алгоритма [7]. При этом:

- вершины, являющиеся концами замороженных ребер, в задаче

о назначениях не принимают участия – замороженные ребра добавляются к оптимальному решению задачи о назначениях принудительно (условно можно считать, что вес каждого замороженного ребра равен отрицательному числу с большим абсолютным значением – это будет гарантировать то, что венгерский алгоритм всегда выберет это ребро);

- запрещенные ребра имеют большой положительный вес, поэтому они никогда не будут включены в оптимальное решение задачи о назначениях (за исключением случаев, когда не существует совершенного паросочетания из незапрещенных ребер).

2. Если набор ребер, полученных в результате решения AP, образует маршруты k коммивояжеров, сравним его длину с рекордным значением и, если нужно, обновим рекордное значение.

3. Пусть оптимальное решение AP состоит из t незамороженных циклов c_1, c_2, \dots, c_t и l замороженных циклов f_1, f_2, \dots, f_l в исходном графе. Тогда значение целевой функции оптимального решения этой подзадачи будет не менее чем $m_1 = \max_{i=1..l} d(f_i)$ и не менее чем $m_2 = \frac{d(c_1) + d(c_2) + \dots + d(c_t)}{k-l}$. Значение, равное $m = \max(m_1, m_2)$, будет нижней оценкой подзадачи.

4. Найдем в решении AP цикл, наименьший по весу. Пусть он состоит из ребер e_1, e_2, \dots, e_s в порядке возрастания весов. Сведем задачу к следующим s подзадачам:

1. Ребро e_1 запрещено.

2. Ребро e_1 заморожено, ребро e_2 запрещено.

3. Ребра e_1, e_2 заморожены, ребро e_3 запрещеною

...

$s - 1$. Ребра e_1, e_2, \dots, e_{s-1} заморожены, ребро e_m запрещено.

s . Ребра e_1, e_2, \dots, e_s заморожены.

Таким образом, рассматриваемая задача разбита на s подзадач (рассматриваемая подзадача разбита на s подзадач следующего уровня).

ОБ АЛГОРИТМЕ ВЕТВЛЕНИЯ В ЗАДАЧЕ НЕСКОЛЬКИХ КОММИВОЯЖЕРОВ

В настоящей работе для рассматриваемой задачи нескольких коммивояжеров разработан алгоритм, основанный на методе ветвей и границ и подобный алгоритму Балаша – Кристофидеса решения задачи одного коммивояжера [2], но с некоторыми отличиями.

Исходный граф в обоих случаях представлен матрицей весов.

В обоих случаях алгоритм ищет набор из n (количество вершин графа) дуг, покрывающих все вершины графа: в задаче коммивояжера все дуги принадлежат одному гамильтонову циклу; в задаче нескольких коммивояжеров все дуги принадлежат заранее заданному количеству k циклов.

Ветвление в дереве вариантов в обоих случаях, как и в алгоритме Литтла [14], строится на основе выбора некоторой дуги графа, для которой имеются два варианта: выбранная дуга 1) принадлежит искомому множеству n дуг или 2) не принадлежит. В результате такого ветвления в каждом узле дерева вариантов фиксируется некоторое множество дуг, принадлежащих искомому циклу (набору циклов), некоторое множество дуг, не принадлежащих искомому циклу (набору циклов), и остальные дуги. Количество узлов такого дерева вариантов можно оценить величиной 2^n (n – количество вершин графа).

Для алгоритма, представленного в настоящей работе, как и для алгоритма Балаша – Кристофидеса, дерево вариантов сокращается путем использования решения задачи о назначениях в каждом узле. В результате решения задачи о назначениях получается множество из n дуг, покрывающих все вершины графа. Такое множество дуг образует набор из нескольких циклов. Таким образом, использование задачи о назначениях сокращает дерево вариантов Литтла. Можно показать, что любое множество циклов, состоящее из n дуг и покрывающее все вершины графа, просматривается таким деревом вариантов. В итоге, точное решение данным алгоритмом не будет пропущено.

От выбора дуги для ветвления зависит дерево вариантов и, как следствие, быстродействие алгоритма, хотя при любом выборе дуги ветвления без отсеечения ветвей просматриваются все наборы из n дуг.

Принцип выбора дуги ветвления определяется оценкой целевой функции. Такая оценка в алгоритме Балаша – Кристофидеса определяется величиной решения задачи о назначениях. Этой же величиной можно оценить целевую функцию в задаче нескольких коммивояжеров mTSP [10, 13], если минимизируется сумма весов дуг. Для рассматриваемой нами задачи нескольких коммивояжеров для оценки целевой функции в каждом узле дерева вариантов нужно оценивать вес максимального цикла. В этом случае в качестве оценки целевой функции можно рассматривать величину решения задачи о назначениях, деленную на заданное количество коммивояжеров k .

Как и в других алгоритмах, основанных на методе ветвей и границ, отсечение ветвей выполняется в тех узлах, в которых оценка целевой функции хуже текущего рекорда.

В алгоритме Балаша – Кристофидеса, если в результате решения задачи о назначениях получен ровно один цикл, этот цикл является оптимальным решением. Если циклов несколько, то находится цикл с наименьшим количеством дуг, и для ветвления выбираются дуги этого цикла. При этом невозможно, чтобы все дуги цикла принадлежали оптимальному решению – это дополнительно сокращает дерево вариантов. В рассматриваемой задаче для ветвления выбирается цикл с наименьшей суммой весов дуг.

ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Выполнены численные эксперименты со случайными графами. Рассматривались полные ориентированные графы. Вес дуги выбирался случайным целым неотрицательным числом, меньшим 10^6 , равновероятно.

Эксперименты проводились на компьютере со следующими характеристиками: Windows 11, процессор Ryzen 7 5800X, 8 ядер, базовая частота 3.8 ГГц, объем L1 кэша 512 КБ, объем L2 кэша 4 МБ, объем L3 кэша 32 МБ. 32 Гб оперативной памяти, DDR4, частота 3200 МГц.

Программа разработана на языке C++. Для запуска компилятора gcc 15.1 использовалась директива O3.

Результаты численных экспериментов представлены в табл. 1. Время выполнения указано в секундах.

Табл. 1. Среднее время (в секундах) решения задачи нескольких коммивояжеров в зависимости от их количества и количества вершин случайного графа.

Количество вершин в графе	Количество коммивояжеров				
	1	2	3	4	5
30	0.001	0.01	0.164	3.277	109.95
40	0.001	0.022	0.621	13.197	—
50	0.002	0.059	1.708	57.85	—
60	0.003	0.105	4.482	88.506	—
80	0.004	0.294	17.4	—	—
100	0.007	0.839	52.152	—	—
150	0.017	2.175	—	—	—

В таблице указано среднее время решения на 100 тестах за исключением случаев, где среднее время превышает 20 с – для них среднее время бралось по 5 тестам. Прочерк в таблице означает, что для таких размерностей численные эксперименты не проводились.

Представленный алгоритм, несмотря на хорошее быстродействие, имеет экспоненциальную сложность, что наблюдается даже в табл. 1. Однако мала вероятность того, что на сгенерированной случайной матрице размерностей, представленных в табл. 1, алгоритм будет работать более часа. Такие данные во время экспериментов не встретились.

В [13] представлены алгоритм решения задачи mTSP (несколько коммивояжеров с минимизацией суммы весов маршрутов) и численные эксперименты для программной реализации алгоритма. Табл. 1 демонстрирует производительность, которая выше производительности программы [13] более чем на порядок.

ЗАКЛЮЧЕНИЕ

Представленный алгоритм, как и в случае с задачей одного коммивояжера, при использовании вспомогательной задачи о назначениях имеет преимущества по сравнению с алгоритмами, которые эту задачу не используют.

Проведенные эксперименты демонстрируют это превосходство.

В дальнейшем предполагается рассмотреть распараллеливание и другие оптимизации, увеличивающие быстродействие программы.

Благодарность

Работа поддержана грантом государственного задания № FENW-2026-0031.

СПИСОК ЛИТЕРАТУРЫ

1. *Гэри М., Джонсон Д.* Вычислительные машины и трудно решаемые задачи. М., Мир, 1982. 416 с.
2. *Balas E., Christofides N.* A restricted lagrangean approach to the traveling salesman problem // *Mathematical Programming*. 1981. Vol. 21. No. 1. P. 19–46. <https://doi.org/10.1007/BF01584228>.
3. *Бурховецкий В.В.* Оптимизация и распараллеливание упрощенного алгоритма Балаша–Кристофидеса для задачи коммивояжера // *Программные системы: теория и приложения*. 2020. Т. 11. Вып. 4. С. 3–16. <https://doi.org/10.25209/2079-3316-2020-11-4-3-16>.
4. *Бурховецкий В.В., Штейнберг Б.Я.* Точное и приближенное решения задачи коммивояжера большого размера // *Вычислительные методы и программирование*. 2024. Т. 25, Вып. 4. С. 476–482. <https://doi.org/10.26089/NumMet.v25r436>.
5. *Steinberg B., Baglij A., Petrenko V., Burhovetckij V., Steinberg O., Metelica E.* An Analyzer for Program parallelization and Optimization // 3-rd International Conference on Applications in Information Technologies ICAIT 2018, Tokyo, November 1–3, 2018. <https://doi.org/10.1145/3274856.3274875>.
6. *Feautrier P.* A parallelization framework for recursive tree programs // In: Pritchard, D., Reeve, J. (eds) *Euro-Par'98 Parallel Processing*. Euro-Par 1998. Lecture Notes in Computer Science. 1998. Vol 1470. Springer, Berlin, Heidelberg. P. 470–479. <https://doi.org/10.1007/BFb0057890>
7. *Kuhn H.W.* The Hungarian Method for the assignment problem // *Naval Research Logistics Quarterly*. 1955. Vol. 2. P. 83–97. <https://doi.org/10.1002/nav.3800020109>.

8. *Christofides N.* Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem // *Oper. Res. Forum* 3, 20 (2022).
<https://doi.org/10.1007/s43069-021-00101-z>
9. *Курейчик В.В., Логунова Ю.А.* Задача коммивояжера. Обзор и методы ее решения. Palmarium Academic Publishing, 2020. – 60 с.
10. *Bektas T.* The multiple traveling salesman problem: an overview of formulations and solution procedures // *Omega*. 2006. Vol. 34, Issue 3. P. 209–219.
<https://doi.org/10.1016/j.omega.2004.10.004>
11. *Kennington JL A.A.* The asymmetric m-traveling salesmen problem: a duality based branchand-bound algorithm // *Discrete Applied Mathematics* 1986. Vol. 13. P. 259–276. [https://doi.org/10.1016/0166-218x\(86\)90087-9](https://doi.org/10.1016/0166-218x(86)90087-9)
12. *Laporte G, Nobert Y.* A cutting planes algorithm for the m-salesmen problem // *Journal of the Operational Research Society*. 1980. Vol. 31. P. 1017–1023.
<https://doi.org/10.1057/jors.1980.188>
13. *Ahmeda Z.H., Al-Dayel I.* An Exact Algorithm for the Single Depot Multiple Travelling Salesman Problem // *IJCSNS International Journal of Computer Science and Network Security*. 2020. Vol. 20. No. 9.
<https://doi.org/10.22937/IJCSNS.2020.20.09.9>
14. *Little J., Murty K., Sweeney D., Karel C.* An Algorithm for the Traveling Salesman Problem // *Operations Research*. 1963. Vol. 11, No. 6. P. 972–989.
<https://doi.org/10.1287/opre.11.6.972>
15. *Böhm M., Friggstad Z., Mömke T., Spoerhase J.* Approximating Traveling Salesman Problems Using a Bridge Lemma // *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms*.
<https://doi.org/10.1137/1.9781611978322.34>.

AN ALGORITHM FOR FINDING AN EXACT SOLUTION TO THE PROBLEM OF MULTIPLE TRAVELLING SALESMEN

O. A. Klimenko¹ [0009-0000-5236-2415], B. J. Steinberg² [0000-0001-8146-0479]

¹ Southern Federal University, Rostov-on-Don, Russia

¹oleg.klimenko.math@mail.ru, ²borsteinb@mail.ru

Abstract

This article considers the problem of several traveling salesmen. The task is to find a set of a predetermined number of disjoint cycles on a graph with weighted arcs, in which the weight (the sum of the weights of the arcs) of the largest cycle is minimal. An accurate algorithm for solving the problem based on the method of branches and boundaries has been developed. The constructed algorithm, as well as the well-known Balas' and Christofides' algorithm for solving the traveling salesman problem, uses the Hungarian algorithm for solving the assignment problem. Numerical experiments with large-dimensional random graphs have been carried out.

Keywords: *traveling salesman problem, assignment problem, Hungarian algorithm, branch and bound method, graphs, discrete optimization, Hamiltonian cycle.*

REFERENCES

1. Geri M., Dzhonson D. Vychislitel'nye mashiny i trudno reshaemye zadachi. M.: Mir, 1982. 416 s.
2. Balas E., Christofides N. A restricted lagrangean approach to the traveling salesman problem // Mathematical Programming. 1981. Vol. 21. No. 1. P. 19–46. <https://doi.org/10.1007/BF01584228>
3. Burhoveckij V.V. Optimization and Parallelization of Simplified Balas' and Christofides' Algorithm for the Traveling Salesman Problem. Program Systems: Theory and Applications // Program Systems: Theory and Applications. 2020. T. 11, Vyp. 4. S. 3–16. <https://doi.org/10.25209/2079-3316-2020-11-4-3-16>
4. Burhoveckij V.V., Steinberg B.J. Tochnoe i priblizhennoe resheniya zadachi kommivoyazhyora bol'shogo razmera // Vychislitel'nye metody i programmirovaniye. 2024. T. 25, Vyp. 4. S. 476–482. <https://doi.org/10.26089/NumMet.v25r436>
5. Steinberg B., Baglij A., Petrenko V., Burhovetckij V., Steinberg O., Metelica E.

An Analyzer for Program parallelization and Optimization // 3-rd International Conference on Applications in Information Technologies ICAIT 2018, Tokyo, November 1–3, 2018. <https://doi.org/10.1145/3274856.3274875>.

6. *Feautrier P.* A parallelization framework for recursive tree programs / Feautrier P // EuroPar'98 Parallel Processing: 4th International Euro-Par Conference Southampton, UK, September 1–4, 1998 Proceedings 4. Springer, 1998. P. 470–479. <https://doi.org/10.1007/BFb0057890>

7. *Kuhn H.W.* The Hungarian Method for the assignment problem // Naval Research Logistics Quarterly. 1955. Vol. 2. P. 83–97. <https://doi.org/10.1002/nav.3800020109>

8. *Christofides N.* Worst-case analysis of a new heuristic for the travelling salesman problem. Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group (1976). <https://doi.org/10.1007/s43069-021-00101-z>

9. *Kurejchik V.V., Logunova Y.A.* Zadacha kommivoyazhyora. Obzor i metody eyo resheniya. Palmarium Academic Publishing, 2020.

10. *Bektas T.* The multiple traveling salesman problem: an overview of formulations and solution procedures // Omega. 2006. Vol. 34, Issue 3. P. 209–219. <https://doi.org/10.1016/j.omega.2004.10.004>

11. *Ali Al, Kennington JL.* The asymmetric m-traveling salesmen problem: a duality based branchand-bound algorithm // Discrete Applied Mathematics 1986. Vol. 13. P. 259–276. [https://doi.org/10.1016/0166-218x\(86\)90087-9](https://doi.org/10.1016/0166-218x(86)90087-9)

12. *Laporte G, Nobert Y.* A cutting planes algorithm for the m-salesmen problem // Journal of the Operational Research Society. 1980. Vol. 31. P. 1017–1023. <https://doi.org/10.1057/jors.1980.188>

13. *Ahmeda Z.H., Al-Dayel I.* An Exact Algorithm for the Single Depot Multiple Travelling Salesman Problem // IJCSNS International Journal of Computer Science and Network Security. 2020. Vol. 20. No. 9. <https://doi.org/10.22937/IJCSNS.2020.20.09.9>

14. *Little J., Murty K., Sweeney D., Karel C.* An Algorithm for the Traveling Salesman Problem // Operations Research. 1963. Vol. 11, No. 6. P. 972–989. <https://doi.org/10.1287/opre.11.6.972>

15. *Böhm M., Friggstad Z., Mömke T., Spoerhase J.* Approximating Traveling

Salesman Problems Using a Bridge Lemma // Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms.
<https://doi.org/10.1137/1.9781611978322.34>

СВЕДЕНИЯ ОБ АВТОРАХ



КЛИМЕНКО Олег Александрович – аспирант мехмата Южного федерального университета.

Oleg Alexandrovich KLIMENKO – postgraduate student at the Faculty of Mechanics and Mathematics, Southern Federal University.

email: oleg.klimenko.math@mail.ru

ORCID: 0009-0000-5236-2415



ШТЕЙНБЕРГ Борис Яковлевич – доктор технических наук, заведующий кафедры алгебры и дискретной математики мехмата Южного федерального университета.

Boris Yakovlevich STEINBERG – Doctor of Technical Sciences, head at the Faculty of Mechanics and Mathematics, Southern Federal University.

email: borsteinb@mail.ru

ORCID: 0000-0001-8146-0479

Материал поступил в редакцию 19 января 2026 года