

Алгоритм поиска точного решения задачи нескольких коммивояжеров

Клименко О. А.¹, Штейнберг Б.Я.¹

¹ Южный Федеральный Университет

Аннотация. В работе рассмотрена задача нескольких коммивояжеров. Задача состоит в том, чтобы на графе со взвешенными дугами найти набор из заранее заданного количества непересекающихся циклов, у которого вес (сумма весов дуг) наибольшего цикла минимален. Разработан точный, основанный на методе ветвей и границ алгоритм решения поставленной задачи. В построенном алгоритме, как и в известном алгоритме Балаша-Кристофидеса решения задачи одного коммивояжера, используется венгерский алгоритм решения задачи о назначениях. Проведены численные эксперименты со случайными графами большой размерности.

Ключевые слова: задача коммивояжера, задача о назначениях, венгерский алгоритм, метод ветвей и границ.

An algorithm for finding an exact solution to the problem of multiple traveling salesmen

O. A. Klimenko¹, B. J. Steinberg¹

¹ Southern Federal University

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.

Введение

В данной работе рассмотрена задача нескольких коммивояжеров. Эта задача является обобщением задачи коммивояжера, алгоритмам решения которой посвящено множество публикаций.

Задача коммивояжёра (замкнутая) — задача нахождения гамильтонова цикла с минимальной суммой дуг в полном взвешенном ориентированном графе, стоимости дуг которого неотрицательны.

В основе некоторых эффективных алгоритмов лежит алгоритм Балаша-Кристофидеса [1].

Алгоритм Балаша-Кристофидеса — точный алгоритм решения задачи коммивояжёра, основанный на методе ветвей и границ («точный» означает, что он всегда находит оптимальное решение).

Эффективная реализация, адаптированная к современным процессорам, представлена в [3]. В частности, применяются оптимизация использования иерархии памяти и распараллеливание.

Обход дерева вариантов в методе ветвей и границ можно описывать рекурсивной функцией. Распараллеливание рекурсивных функций рассматривалось в [9], [10].

В [8] рассмотрен полиномиальный эвристический алгоритм решения задачи коммивояжёра с оценкой качества найденного решения. Вычислительная сложность этого алгоритма $O(N^3)$, а найденное этим алгоритмом решение хуже оптимального не более чем в 1.5 раза. В работе [11] есть алгоритм с оценкой сложности «найденное решение хуже оптимального не более, чем в k раз (k - параметр)». Этот алгоритм для случайных графов размерности 50 тысяч вершин на компьютере с шестиядерным процессором находит решение задачи коммивояжера, которое хуже оптимального не более, чем на 5% ($k = 1.05$).

Существуют различные задачи нескольких коммивояжеров. В этих задачах требуется отыскать набор оптимальных маршрутов для нескольких коммивояжёров. Формулировки этих задач отличаются выбором целевой функции, точек старта коммивояжёров или требованием замкнутости (разомкнутости) маршрутов. Обзор таких работ представлен в статье [2].

Имеется множество работ, посвящённых решению задачи, где в качестве целевой функции выбирается сумма стоимостей маршрутов коммивояжёров. В работах [6] и [7] рассматривается точное решение при небольшом количестве коммивояжёров (не более 10). Имеются и недавние результаты [4].

В данной работе в качестве целевой функции выбирается максимум из длин путей, пройденных коммивояжёрами. В такой постановке задаче нескольких коммивояжеров имеет практические применения, например, если маршруты «коммивояжеров» следует ограничивать длительностью рабочего дня или запасом горючего. Для решения задачи используется

венгерский алгоритм решения вспомогательной задачи о назначениях, как и в известном алгоритме Балаша-Кристофидеса [1].

Проведены численные эксперименты со случайными графами, демонстрирующие эффективность представленного алгоритма.

Формулировка задачи нескольких коммивояжёров

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, назовём *маршрутами k коммивояжёров*. Цикл c_i будем называть путём (или маршрутом) i -того коммивояжёра. *Стоимостью набора маршрутов k коммивояжёров* назовём величину

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

где $d(c_i)$ — стоимость маршрута i -того коммивояжёра (сумма весов дуг).

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

Замечание. Задача нескольких коммивояжёров NP-трудна, так как NP-трудна задача одного коммивояжёра, являющаяся частным случаем при $k = 1$.

Решение задачи нескольких коммивояжёров

Будем искать точное решение задачи нескольких коммивояжёров методом ветвей и границ. В методе ветвей и границ исходная задача рекурсивно разбивается на другие задачи, которые решаются проще (например, меньшей размерности или с другой оценкой целевой функции). В ходе работы алгоритма ветвей и границ встречаются различные наборы покрывающих все вершины k циклов и запоминает лучший из них. Такой набор назовём *рекордным*, а значение целевой функции рекордного набора назовём *рекордным значением*.

Для каждой подзадачи вычисляется *нижняя оценка целевой функции на основе решения задачи о назначениях*. **Гарантируется, что** (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.

Структура подзадач

На очередной итерации алгоритма мы будем работать с подзадачей, характеризующейся следующими ограничениями:

1. Рёбра $(v_1, u_1), (v_2, u_2), \dots, (v_t, u_t)$ — заморожены. Это означает, что каждое из перечисленных рёбер должно содержаться в каком-либо цикле $c_1, c_2, \dots, c_k$. Гарантируется, что в каждую вершину входит не более чем одно замороженное ребро, а также выходит не более чем одно замороженное ребро. Замороженные рёбра могут образовывать циклы. Такие циклы мы также будем называть *замороженными*, остальные циклы в графе — *незамороженные* (хотя они могут содержать замороженные рёбра). Замороженные циклы соответствуют маршруту некоторого коммивояжёра.

2. Рёбра $(v'_1, u'_1), (v'_2, u'_2), \dots, (v'_l, u'_l)$ — запрещены. Это означает, что каждое из перечисленных рёбер не содержится ни в одном из циклов $c_1, c_2, \dots, c_k$.

Таким образом, мы ищем оптимальное решение с учётом упомянутых выше ограничений.

Для решения подзадачи мы разбиваем её, накладывая в каждой из дочерних подзадач дополнительные ограничения (новые замороженные или запрещённые рёбра). Исходная задача не имеет ограничений (то есть все рёбра 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 запрещённого ребра, то решений нет.

В оригинальном алгоритме Балаша-Кристофидеса используется термин «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».

Ветвление в методе ветвей и границ

Разбиение на подзадачи происходит следующим образом:

1. Для матрицы графа решается задача о назначениях (AP) с помощью венгерского алгоритма [10]. При том

- вершины, являющиеся концами замороженных рёбер, в задаче о назначениях участия не принимают — замороженные рёбра добавляются к оптимальному решению задачи о назначениях принудительно (условно можно считать, что вес каждого замороженного ребра равен $-\infty$).
- запрещённые рёбра имеют вес $+\infty$, потому они никогда не будут включены в оптимальное решение задачи о назначениях (за исключением случаев, когда не существует совершенного паросочетания из незапрещённых рёбер).

2. Если набор рёбер, полученных в результате решения АР, образует маршруты k коммивояжёров, сравним его длину с рекордным значением и, если нужно, обновим его.

3. Пусть оптимальное решение АР 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 не менее чем $m_1 = \max_{i=1..l} d(f_i)$ и не менее чем $m = \frac{d(c_1) + d(c_2) + \dots + d(c_t)}{k-l}$. Значение, равное, $m = \max(m_1, m_2)$ будет нижней оценкой подзадачи.

4. Найдём в решении АР наименьший по весу цикл. Пусть он состоит из рёбер $e_1, e_2, \dots, e_m$ в порядке возрастания весов. Сведём задачу к следующим m подзадачам:

1. Ребро e_1 — запрещено;
2. Ребро e_1 — заморожено, ребро e_2 — запрещено;
3. Рёбра e_1, e_2 — заморожены, ребро e_3 — запрещено;
- ...

$m - 1$. Рёбра $e_1, e_2, \dots, e_{m-1}$ — заморожены, ребро e_m — запрещено;

m . Рёбра $e_1, e_2, \dots, e_m$ — заморожены.

Об алгоритме решения задачи нескольких коммивояжёров.

В данной работе для рассматриваемой задачи нескольких коммивояжёров разработан алгоритм, основанный на методе ветвей и границ подобный алгоритму Балаша-Кристофидеса решения задачи одного коммивояжера [1], но с некоторыми неизбежными отличиями.

Исходный граф в обоих случаях представлен матрицей весов.

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

Ветвление в дереве вариантов в обоих случаях, как и в алгоритме Литтла [12], строится на основе выбора некоторой дуги графа, для которой строятся два варианта: 1) выбранная дуга 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 представленного в данной работе алгоритма как и для алгоритма Балаша-Кристофидеса дерево вариантов сокращается использованием решения задачи о назначениях в каждом узле. В результате решения задачи о назначениях получается множество из n дуг, покрывающих все вершины графа. Получаемое 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. Таким образом, 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 отсечения ветвей просматриваются все наборы из n дуг.

Принцип выбора дуги ветвления определяется оценкой целевой функции.

Оценка целевой функции в алгоритме Балаша-Кристофидеса определяется величиной решения задачи о назначениях. Этой же величиной можно оценивать целевую функцию в задаче нескольких коммивояжеров $mTSP$ [4], [2], если минимизируется сумма весов дуг. Для рассматриваемой в данной работе 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 - то 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 принадлежат оптимальному решению - это дополнительно сокращает дерево вариантов.

В рассматриваемой задаче для ветвления выбирается цикл с наименьшей суммой весов дуг.

Численные эксперименты

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

Численные эксперименты проводились на компьютере со следующими характеристиками.

Windows 11, процессор Ryzen 7 5800X, 8 ядер, базовая частота 3.8 ГГц, объём L1 кэша 512 КБ, объём L2 кэша 4 мб, объём L3 кэша 32 Мб. 32 Гб оперативной памяти, DDR4, частота 3200 МГц.

Программа разработана на языке C++. Компилятор gsc 15.1, компилировал с директивой -O3.

Результаты численных экспериментов представлены в Таблице 1. По строкам указано количество вершин в графе, по столбцам — количество коммивояжёров. Время выполнения указано в секундах.

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

	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 секунд – 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 даже в таблице 1 наблюдается. Однако мала вероятность того, что на сгенерированной случайной матрице представленных в таблице 1 размерностей алгоритм будет работать более часа - такие данные во время экспериментов не встретились.

В [4] представлен алгоритм решения задачи mTSP (несколько коммивояжеров с минимизацией суммы весов маршрутов) и численные эксперименты для программной реализации алгоритма. Численные эксперименты Таблицы 1 демонстрируют производительность, которая выше производительности программы [4], 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.

Список литературы

1. Egon Balas. A restricted lagrangean approach to the traveling salesman problem / Egon Balas, Nicos Christofides // Mathematical Programming. — 1981. — Vol. 21, no. 1. — Pp. 19–46.
2. Tolga Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega, Volume 34, Issue 3, 2006, Pages 209-219, ISSN 0305-0483.
3. Бурховецкий. Optimization and Parallelization of Simplified Balas'; and Christofides'; Algorithm for the Traveling Salesman Problem. Program Systems: Theory and Applications. Программные системы: теория и приложения. 10.25209/2079-3316-2020-11-4-3-16. 11:4. (3-16).
4. Zakir Hussain Ahmeda and Ibrahim Al-Dayel An Exact Algorithm for the Single-Depot Multiple Travelling Salesman Problem. IJCSNS International Journal of Computer Science and Network Security, VOL.20 No.9, September 2020
5. Martin Böhm, Zachary Friggstad, Tobias Mömke, and Joachim Spoerhase, Approximating Traveling Salesman Problems Using a Bridge Lemma. Proceedings of the 2025 Annual ACM SIAM Symposium on Discrete Algorithms.
6. Ali Al, Kennington JL. The asymmetric m-traveling salesmen problem: a duality based branch-and-bound algorithm. Discrete Applied Mathematics 1986;13:259–76.
7. Laporte G, Nobert Y. A cutting planes algorithm for the m-salesmen problem. Journal of the Operational Research Society 1980;31:1017–23.
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)

9. 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. — Pp. 470–479.
10. Kuhn, H.W. (1955) The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly, 2, 83-97.
11. Бурховецкий В.В., Штейнберг Б.Я. Точное и приближенное решения задачи коммивояжёра большого размера. Вычислительные методы и программирование, 2024, т.25, вып. 4, с. 476-482.
12. An Algorithm for the Traveling Salesman Problem / Little J., Murty K., Sweeney D., Karel C. // Operations Research. — 1963. — dec. — Vol. 11, no. 6. — Pp. 972–989.