

«Метрическая задача коммивояжёра»
ФПМИ МФТИ

Тимур Зыков, Б05-121

Весна 2023

Аннотация

Задача коммивояжёра (или TSP от англ. travelling salesman problem) — одна из самых известных задач комбинаторной оптимизации, заключающаяся в поиске самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. Вместе с простотой определения и сравнительной простотой нахождения хороших решений задача коммивояжёра отличается тем, что нахождение действительно оптимального пути является достаточно сложной задачей. Учитывая эти свойства, начиная со второй половины XX века исследование задачи коммивояжёра имеет не столько практический смысл, сколько теоретический в качестве модели для разработки новых алгоритмов оптимизации.

Содержание

1 Введение	3
2 Основные понятия и утверждения	3
2.1 Определения и постановка задачи	3
2.2 NP-трудность	4
3 Алгоритмы приближений метрической задачи коммивояжёра	4
3.1 Алгоритм 2-приближения	5
3.1.1 Описание	5
3.1.2 Корректность	5
3.1.3 Асимптотика	6
3.2 Алгоритм $\frac{3}{2}$ -приближения	6
3.2.1 Описание	6
3.2.2 Корректность	7
3.2.3 Асимптотика	7
3.2.4 Анализ работы алгоритма	8
3.2.5 Достижимость верхней оценки	10
4 Выводы	10
5 Приложение	11
Список литературы	11

1 Введение

В 1950-е и 1960-е годы задача коммивояжёра привлекла внимание ученых в США и Европе. Важный вклад в исследование задачи принадлежит Джорджу Данцигу, Делберту Рею Фалкерсону и Селмеру Джонсону, которые в 1954 году в институте RAND Corporation сформулировали задачу в виде задачи дискретной оптимизации и применили для её решения метод отсечений. Используя этот метод, они построили путь коммивояжёра для одной частной постановки задачи с 49 городами и обосновали его оптимальность. В 1960-е и 1970-е годы задача изучалась многими учеными как теоретически, так и с точки зрения её приложений в информатике, экономике, химии и биологии.

Ричард Карп в 1972 году доказал NP-полноту задачи поиска гамильтоновых путей, из чего, благодаря полиномиальной сводимости, вытекала NP-трудность задачи коммивояжёра. На основе этих свойств им было приведено теоретическое обоснование сложности поиска решений задачи на практике.

Больших успехов удалось достичь в конце 1970-х и 1980-х годах, когда Мартин Грётчел, Манфред Падберг и Джованни Ринальди и другие, с применением новых методов деления плоскостью, ветвей и границ вычислили решение для отдельного случая задачи с 2393 городами.

В марте 2005 года задача с 33810 узлами была решена программой: был вычислен путь длиной в 66048945 и доказано отсутствие более коротких путей. В апреле 2006 было найдено решение для экземпляра с 85900 узлами.

На практике поиск строго оптимального маршрута требуется не всегда. Существуют алгоритмы поиска приближенных решений для метрической задачи за полиномиальное время и нахождения маршрута максимум вдвое длиннее оптимального.

2 Основные понятия и утверждения

2.1 Определения и постановка задачи

Определение 2.1. Графом $G = (V, E)$ называется пара множества вершин и множества ребер.

Определение 2.2. Граф $K_n = (V, E)$, $|V| = n$ называется полным, если у него есть все возможные ребра. То есть $|E| = C_n^2$.

Определение 2.3. Маршрутом в графе $G = (V, E)$ будем называть чередующуюся последовательность вершин и ребер, которая начинается и заканчивается на вершинах.

$$v_1 e_1 v_2 \dots e_n v_{n+1}, \forall e_i : e_i = (v_i, v_{i+1})$$

Определение 2.4. Циклом назовем замкнутый маршрут, где все ребра разные.

Определение 2.5. Простой цикл это цикл, где все вершины разные.

Определение 2.6. Гамильтонов цикл - простой цикл, проходящий через все вершины по одному разу.

Определение 2.7. Классом $\mathbf{DTIME}(T(n))$ называется класс языков, которые распознаются за время $O(T(n))$.

Определение 2.8. $P = \bigcup_{i=1}^{\infty} DTIME(n^i)$

Определение 2.9. Классом $NTIME(T(n))$ называется класс языков, которые распознаются на недетерминированной машине Тьюринга за время $O(T(n))$.

Определение 2.10. $NP = \bigcup_{i=1}^{\infty} NTIME(n^i)$

Замечание. $P \subset NP$

Определение 2.11. Язык B является NP -трудным, если для любого $A \in NP$ выполнено $A \leq_p B$. Язык B является NP -полным, если он NP -трудный и лежит в NP .

Замечание. Язык $HAMCYCLE = \{G \mid \text{в графе } G \text{ существует гамильтонов цикл}\}$ является NP -полным.

Задача. В общем случае задача коммивояжёра ставится так: дан граф $G = (V, E)$ и веса на рёбрах $w : E \rightarrow \mathbb{R}_+$. Требуется найти гамильтонов цикл минимального веса. В метрическом случае граф полный, а функция весов метрическая (то есть для любых трёх вершин x, y и z выполнено $w(x, z) \leq w(x, y) + w(y, z)$).

Определение 2.12. Алгоритм даёт α -приближение, если он даёт результат не более чем в α раз превосходящий истинный оптимальный.

Замечание. Можно рассматривать $\alpha = \alpha(n)$, где обычно $n = |V|$.

2.2 NP-трудность

В общем случае, задача коммивояжера не может быть приближена за полином, если $P \neq NP$.

Теорема 2.1. Для стандартной задачи коммивояжера не существует константных алгоритмов приближения, работающих за полином и если $P \neq NP$.

Доказательство. Сведем задачу $HAMCYCLE$ к поиску α -приближения в задаче коммивояжера. Пусть дан граф $G = (V, E)$. Сделаем его полным $G' = (V, E')$, где рёбра из E имеют вес 1, другие имеют вес $K > \alpha n$. Тогда, если в исходном графе был гамильтонов цикл, в новом графе будет цикл коммивояжера длины n . Иначе в любом цикле будет ребро не из E — не было гамильтонова цикла изначально. \square

3 Алгоритмы приближений метрической задачи коммивояжёра

Если мы ограничимся графами, функции веса ребёр которых удовлетворяет неравенству треугольника, т.е. рассмотрим метрическую задачу коммивояжера, задача остается NP -трудной, но ее уже не трудно аппроксимировать. Для таких задач построены полиномиальные приближённые алгоритмы, так называемые PTAS (polynomial-time approximation schemes — аппроксимирующие схемы, работающие за полиномиальное время).

Определение 3.1. Деревом называется связный ациклический граф.

Определение 3.2. Минимальное остовное дерево (MST) — подграф-дерево минимального веса, которое содержит все исходные вершины.

Определение 3.3. Эйлеров цикл — цикл, проходящий через каждое ребро графа ровно по одному разу.

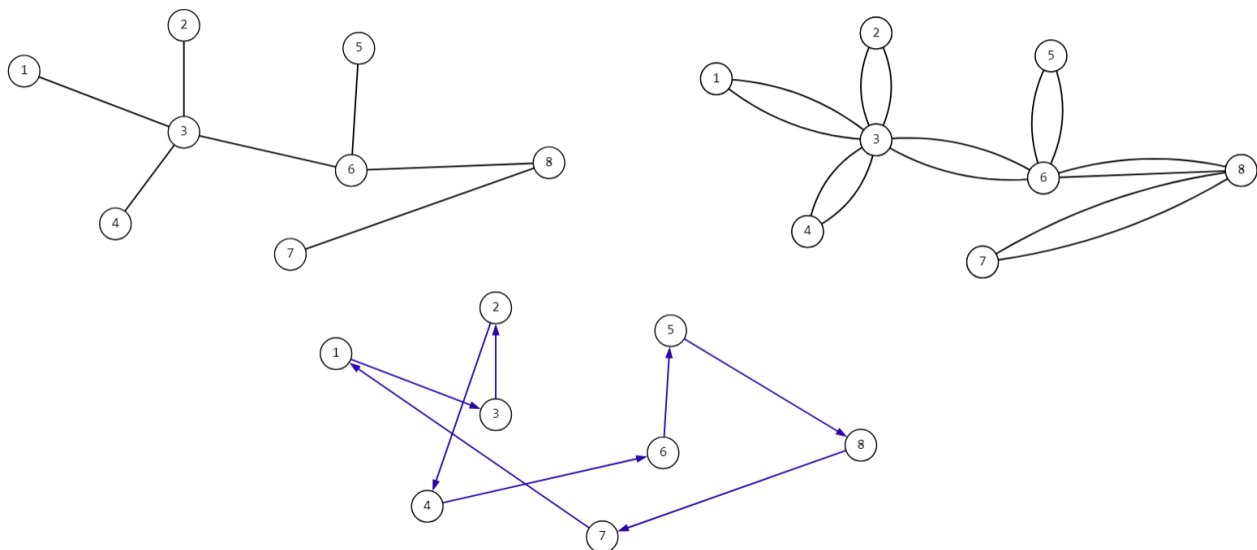
3.1 Алгоритм 2-приближения

Пусть G – граф, с весовой функцией между любыми вершинами, удовлетворяющий неравенству треугольника.

3.1.1 Описание

Algorithm 1: Алгоритм 2-приближения МЗК

1. Пусть $T = \text{MST}(G)$.
 2. Удвоим все ребра в T чтобы затем найти эйлеров цикл.
 3. Строим эйлеров цикл τ на нем.
 4. В любом порядке обойти цикл и вывести первые вхождения вершин.
-



3.1.2 Корректность

Утверждение 3.1. Пусть H – гамильтонов цикл и T – MST. Тогда $w(H) \geq w(T)$, где w – сумма весов ребер.

Доказательство. Если убрать 1 ребро из гамильтонова цикла получим дерево, вес которого не меньше веса MST по определению. \square

Утверждение 3.2. Алгоритм 1 дает 2-приближение МЗК.

Доказательство. Пусть ОПТ это истинное решение МЗК – гамильтонов цикл. Тогда по предыдущему утверждению, $2 \cdot w(\text{ОПТ}) \geq 2 \cdot w(T) = w(\tau)$. Но мы обходим граф не по ребрам, а сразу напрямую от вершины к вершине. Очевидно, наша оценка не ухудшится по неравенству треугольника. \square

3.1.3 Асимптотика

Утверждение 3.3. Алгоритм 1 работает за $\mathcal{O}(n^2)$.

Доказательство. MST строится алгоритмом Прима за $\mathcal{O}(m + n \log n)$. Поиск эйлерова цикла и обходы графов, в том числе добавление ребер, работает за линейное от числа вершин. Но так как мы работаем в метрическом пространстве, то ребер всего n^2 . \square

3.2 Алгоритм $\frac{3}{2}$ -приближения

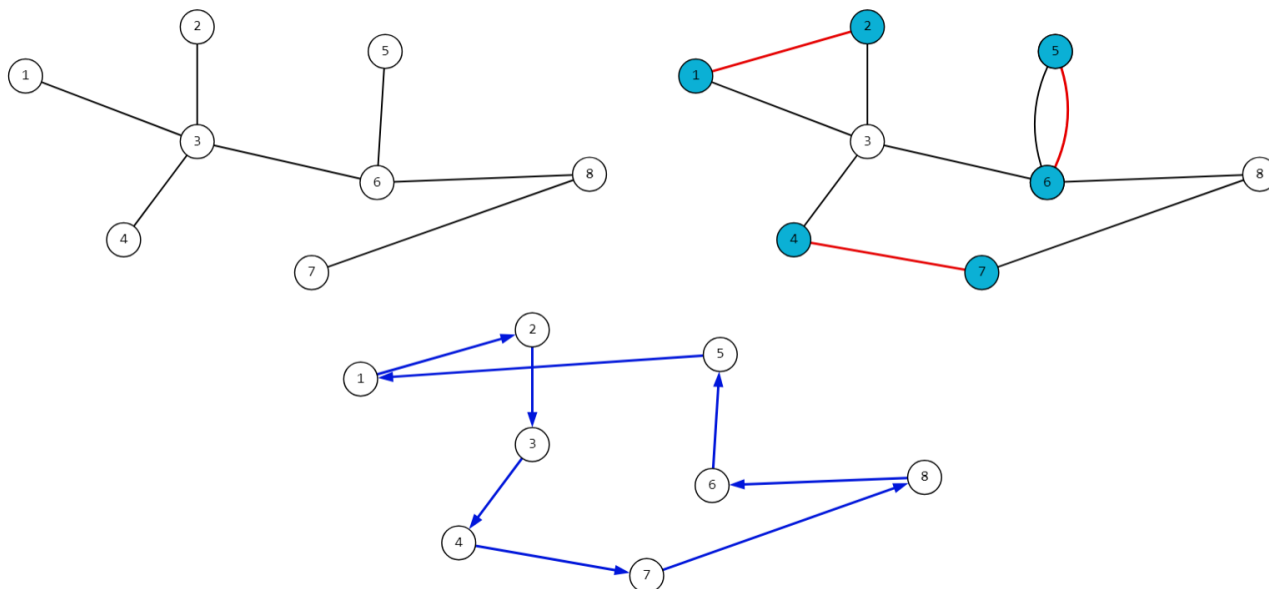
В первом алгоритме мы искали эйлеров цикл и обходили его, но что если существует эйлеров цикл весом меньше? Мы знаем, что эйлеров цикл существует \Leftrightarrow если все вершины четной степени. Улучшим наш алгоритм, используя эти знания.

Определение 3.4. Совершенным паросочетанием называется паросочетание, которое включает все вершины. То есть любая вершина инцидента какому-либо ребру из паросочетания.

3.2.1 Описание

Algorithm 2: Алгоритм $\frac{3}{2}$ -приближения МЗК

1. Пусть $T = \text{MST}(G)$.
 2. Выделим вершины ODD с нечетными степенями из T и построим совершенное паросочетание M минимального веса на $G|_{ODD}$.
 3. Добавим ребра из паросочетания в T и строим эйлеров цикл τ на нем.
 4. В любом порядке обойти цикл и вывести первые вхождения вершин.
-



3.2.2 Корректность

Утверждение 3.4. Для вершин T нечетных степеней существует совершенное паросочетание.

Доказательство. $\sum_{i=1}^n \deg(v_i) = 2|E|$. Отнимем вершины четных степеней и получим опять же четное число, то есть нечетных вершин всегда четное количество. \square

Утверждение 3.5. Пусть $V' \subset V$ такой, что $|V'|$ четно, и пусть M совершенное паросочетание минимального веса на V' . Тогда, $w(M) \leq \text{OPT}/2$.

Доказательство. Пусть существует решение МЗК – обход τ , где $w(\tau) = \text{OPT}$. Ограничим наш обход вершинами из V' , то есть $\tau' = \tau|_{V'}$. Снова получим какой-то обход, где $w(\tau') \leq w(\tau)$ по неравенству треугольника. Заметим, что обход τ' это объединение двух совершенных паросочетаний, ребра которых чередуются в цикле. Из этого факта и из того, что M совершенное паросочетание минимального веса следует, что $w(M) \leq w(\tau')/2 \leq w(\tau)/2 = \text{OPT}/2$. \square

Утверждение 3.6. Алгоритм 2 дает $\frac{3}{2}$ -приближение МЗК.

Доказательство. Пусть τ – наш эйлеров цикл из алгоритма, ans – вывод алгоритма. Тогда выполняется цепочка неравенств:

$$w(ans) \leq w(\tau) = w(T) + w(M) \leq \text{OPT} + \text{OPT}/2 = \frac{3}{2}\text{OPT}$$

\square

3.2.3 Асимптотика

Утверждение 3.7. Алгоритм 2 работает за $\mathcal{O}(n^4)$.

Доказательство. Имеем то же слагаемое в виде $\mathcal{O}(n^2)$ из Алгоритма 2, но теперь нам нужно найти совершенное паросочетание минимального веса. Для этого можно использовать комбинаторный алгоритм^[2], который использует невзвешенный алгоритм Эдмондса в качестве подпрограммы за $\mathcal{O}(n^2m)$, где $m = \mathcal{O}(n^2)$ для полного графа. \square

3.2.4 Анализ работы алгоритма

Будем генерировать точки на плоскости в пределах прямоугольника от $(0, 0)$ до $(100, 100)$. Напишем два генератора, один из которых кидает точку равномерно, а другой генерирует несколько кластеров точек с помощью гауссовских векторов.

Попробуем сначала оценить точность случайных тестах — на равномерных и гауссовских векторах. Будем тестировать при малых n чтобы мы нашли точное решение перебором за адекватное время.

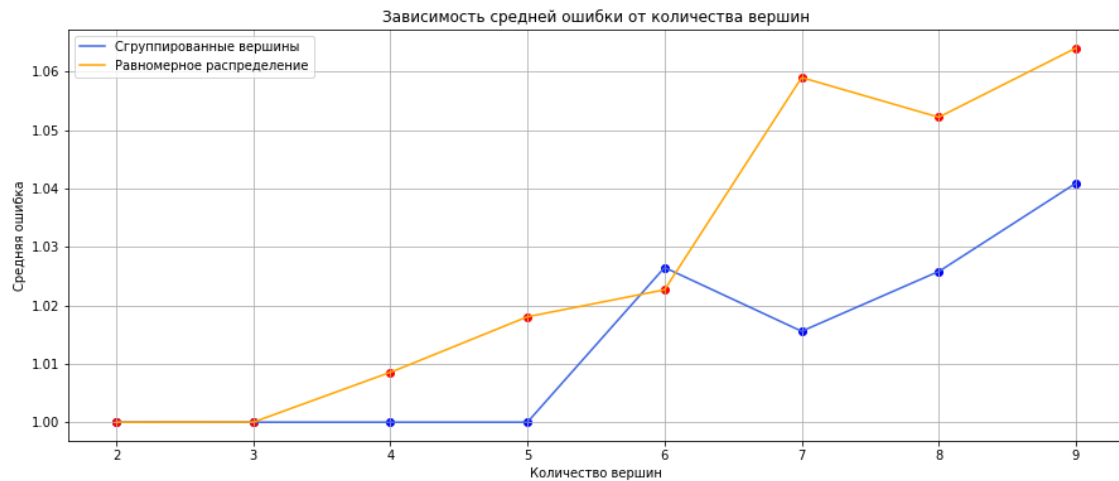


Рис. 1: Точности алгоритмов.

Заметим, что на сгруппированных вершинах наш алгоритм приближает точнее чем на равномерно раскиданных точек. Это логично, так как чем ближе находятся точки, тем меньше ошибка. Интуитивно, можно просто сжимать вершины принадлежащие одному кластеру.

Теперь рассмотрим время работы для больших n . Уже можно не запускать переборный алгоритм, так как мы уже не считаем ошибку. Запустим сначала на равномерно распределенных вершинах.



Рис. 2: Время работы для случайных вершин.

Но интереснее будет посмотреть на зависимость времени работы от количества кластеров.



Рис. 3: Время работы для сгруппированных вершин.

На этих тестах количество кластеров выбирался как $\lfloor n/k \rfloor$, где k — среднее количество вершин на кластер. Не трудно заметить, что скачок вниз происходит тогда, когда k делит $n = 200$ примерно нацело. Это означает, что количество кластеров увеличивается.

Получается, что наш алгоритм очень хорошо работает для кластеризованных графов — когда вершины неявно сгруппованы.

3.2.5 Достижимость верхней оценки

Рассмотрим граф из треугольников как на рисунке 4 когда n - нечетное. Пусть MST стала центральной кривая, которая идет по зигзагу. Тогда всего 2 вершины (конечные) будут иметь нечетную степень. В итоге имеем цикл длиной $(n + 1) + \lfloor n/2 \rfloor$.

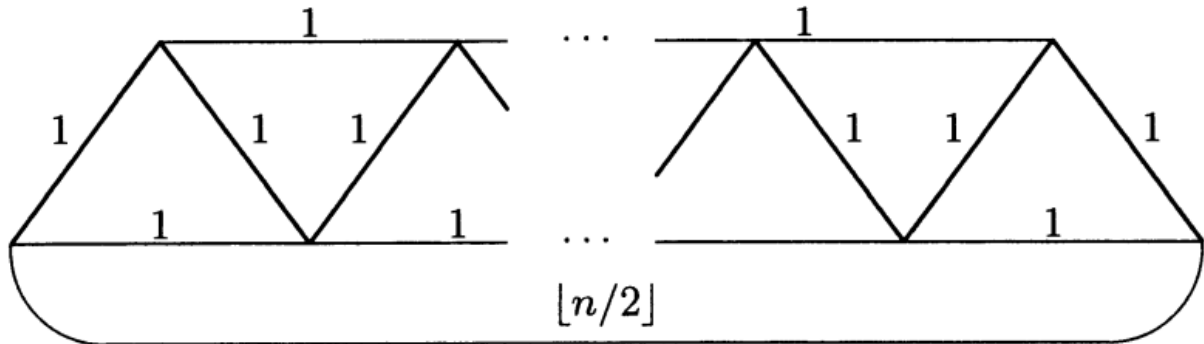


Рис. 4: Граф для верхней оценки.

4 Выводы

Мы рассмотрели два полиномиальных алгоритма, которые неплохо приближают решение метрической задачи коммивояжёра. Они довольно простые, но работают на разных графах по-разному. Все эффективные (сокращающие полный перебор) методы решения задачи коммивояжёра — методы эвристические. Большинство эвристических методов находит не самый эффективный маршрут, а приближённое решение. Я считаю, что $\frac{3}{2}$ уже отличный результат для таких сложных задач.

5 Приложение

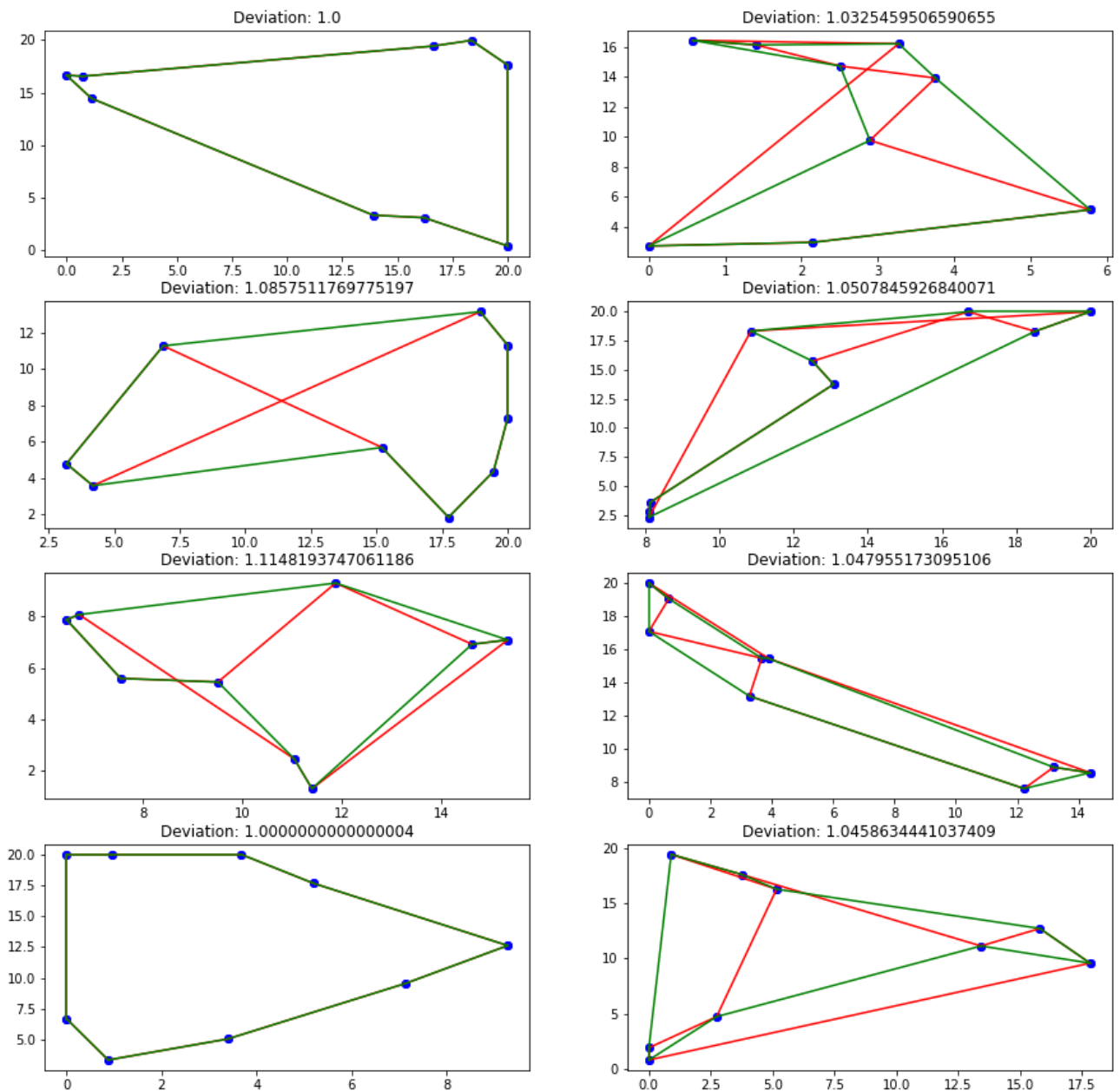


Рис. 5: Несколько примеров работы алгоритма.

Список литературы

- [1] Vazirani, V. Approximation Algorithms, Springer, 2001.
- [2] Kolmogorov, Blossom V: A new implementation of a minimum cost perfect matching algorithm. <https://pub.ista.ac.at/~vnk/papers/BLOSSOM5.html>