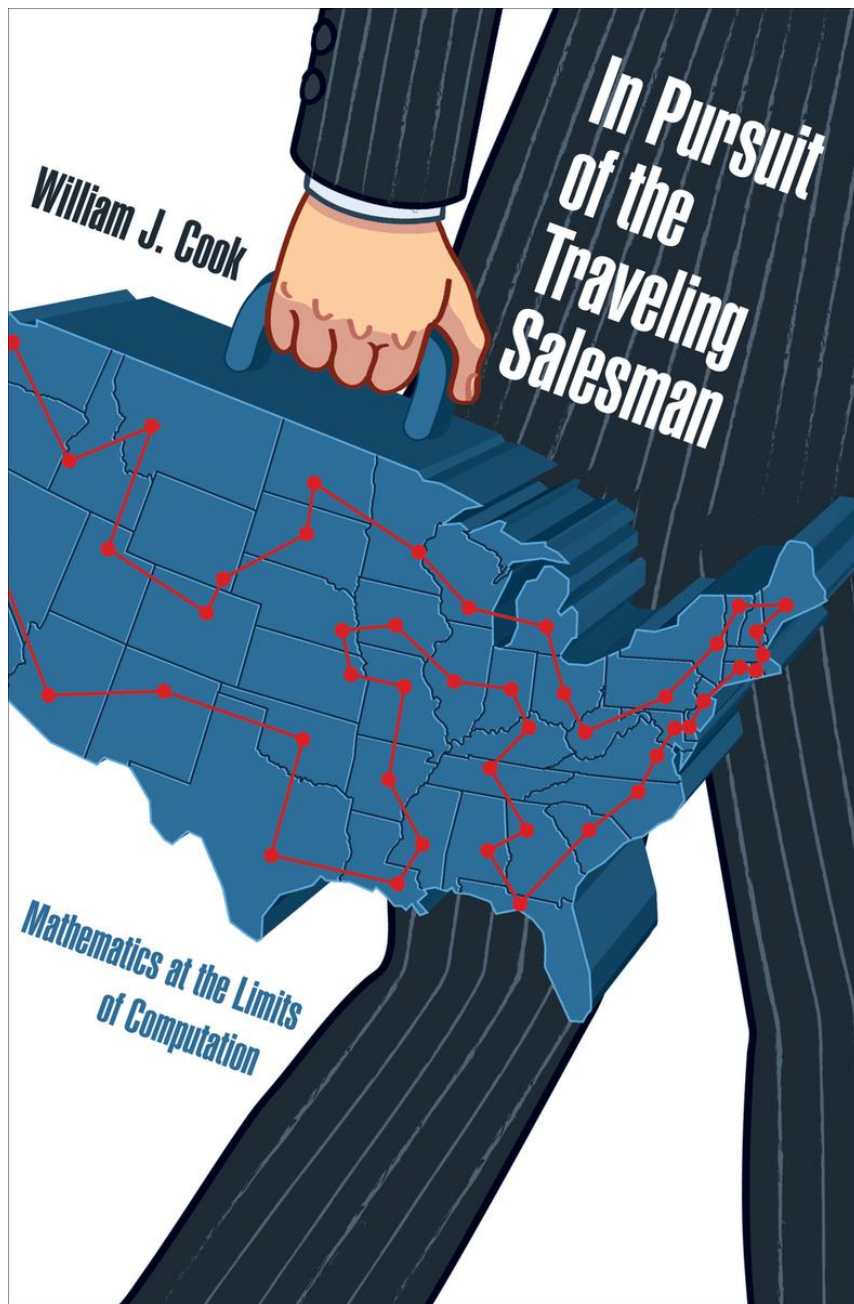


# Метрическая задача коммивояжёра - 1

Ткаченко Дмитрий

17 декабря 2017 г.



In Pursuit of the Traveling Salesman

# Содержание

1 Введение	2
2 Поставленные задачи и вопросы	2
3 Теория и необходимые определения	2
4 Несуществование алгоритма, дающего константное приближение для стандартной задачи коммивояжёра	4
5 Алгоритм, дающий 2-приближение для метрической задачи	4
6 Алгоритм, дающий 1.5-приближение для метрической задачи	5
7 Реализация алгоритма, дающего 1.5-приближение для метрической задачи	6
8 Источники	7

## 1 Введение

Проблема коммивояжёра - одна из самых известных задач комбинаторной оптимизации, с которой ежедневно сталкиваются курьеры, почтальоны, путешественники. Их цель - найти наиболее выгодный маршрут, проходящий через заранее известные места, расстояния между которыми тоже известны. В данной работе рассмотрен метрический случай данной задачи и алгоритмы нахождения её приближенного решения.

## 2 Поставленные задачи и вопросы

- (а) Несуществование полиномиального алгоритма для стандартной задачи коммивояжера, дающего константное приближение, в предположении  $P \neq NP$ .
- (б) Построение алгоритма, дающего 2-приближение для метрической задачи коммивояжёра на основе остовного дерева.
- (в) Построение алгоритма, дающего 1.5-приближение для метрической задачи коммивояжёра на основе остовного дерева и паросочетания.
- (г) Реализация алгоритма, дающего 1.5-приближение для метрической задачи коммивояжёра.

## 3 Теория и необходимые определения

Для начала сформулируем **общую задачу коммивояжёра**: в данном взвешенном графе  $G = (V, E)$  с весами на рёбрах  $w : V \times V \rightarrow \mathbb{R}_+$  найти гамильтонов цикл минимального веса.

**Определения из теории графов:**

**Определение 1** *Путем* в графе  $G = (V, E)$  называется упорядоченный список  $(v_1, e_1, v_2, \dots, e_{k-1}, v_k)$  (иногда ребра в записи опускают), где любые две соседствующие в списке вершины  $v_i$  и  $v_j$  соединены ребром, записанным между ними.

**Определение 2** *Циклом* в графе  $G = (V, E)$  называется путь, в котором  $v_1 = v_k$ .

**Определение 3** *Цикл* в графе  $G = (V, E)$  называется **простым**, если все вершины и ребра в нем встречаются не более, чем один раз.

**Определение 4** *Цикл* в графе называется **гамильтоновым**, если он проходит через все вершины графа ровно по одному разу.

**Определение 5** Граф  $G = (V, E)$  с функцией весов  $w : V \times V \rightarrow \mathbb{R}_+$  называется **метрическим**, если  $\forall x, y, z \in V$  выполнено (неравенство треугольника)  $w(x, z) \leq w(x, y) + w(y, z)$ .

**Определение 6** *Связный граф без простых циклов* называется **деревом**.

**Определение 7** Подграф  $H$  в графе  $G = (V, E)$  называется **остовным деревом**, если  $H$  - дерево и множество вершин  $H$  совпадает с множеством вершин  $G$ .

### Определения из сложности вычислений:

**Определение 8** Классом  $DTIME(T(n))$  называется класс языков, которые распознаются за время  $O(T(n))$ .

**Определение 9** Классом  $NTIME(T(n))$  называется класс языков, которые распознаются на недетерминированной машине Тьюринга за время  $O(T(n))$ .

**Определение 10** Классом  $P$  называется множество языков, которые распознаются за полиномиальное время, или, более формально,  $P = \bigcup_{c=1}^{\infty} DTIME(n^c)$

**Определение 11** Класс  $NP$  есть  $NP = \bigcup_{c=1}^{\infty} NTIME(n^c)$ .

**Определение 12** Язык  $A$  называется  **$NP$ -полным**, если для  $\forall B \in NP$  существует полиномиально вычислимая функция  $f_B : \{0, 1\}^* \rightarrow \{0, 1\}^*$  такая, что  $\forall x \in \{0, 1\}^* : x \in A \Leftrightarrow f_B(x) \in B$ .

**Определение 13** *Метрической задачей коммивояжера* называется язык пар  $L = \{(G, k) \mid \text{в метрическом графе } G \text{ с заданными весами ребер существует гамильтонов цикл веса не более } k\}$ .

**Факт 1** Язык  $HAMCYCLE = \{G \mid \text{в графе } G \text{ существует гамильтонов цикл}\}$  является  **$NP$ -полным**.

### Определения из теории алгоритмов:

**Определение 14** Полиномиальный алгоритм дает  **$c$ -приближение**, если результат его работы отличается от верного не более, чем в  $c$  раз.

## 4 Несуществование алгоритма, дающего константное приближение для стандартной задачи коммивояжёра

Этот раздел будет представлен лишь теоремой из заглавия:

**Теорема 1** (В предположении  $P \neq NP$ ) Для стандартной задачи коммивояжера не существует алгоритма, дающего константное приближение и работающего за полиномиальное время.

**Доказательство:**

▷ Предположим, что такой алгоритм существует (назовем его  $A$ ) и дает  $c$ -приближение. Построим с помощью него полиномиальный алгоритм, который определяет принадлежность графа  $G$  к языку  $HAMCYCLE$ .

Рассмотрим произвольный граф  $G = (V, E)$ . Достроим его до полного взвешенного графа  $Q$ . Ребрам, которые входили в  $G$ , сопоставим вес 1, а всем остальным дадим вес  $|V|c + 1$ . Применим  $A$  к новому графу  $Q$ . Результат - гамильтонов цикл  $H$  веса  $w(H)$ .

Пусть  $w(H) \leq |V|c$ . В таком случае кратчайший гамильтонов цикл имеет вес  $w_{opt} \leq w(H)$ , а значит в нем нет новых ребер (так как вес каждого из них превосходит  $w(H)$ ), тогда  $H$  присутствовал и в графе  $G$ .

Пусть  $w(H) > |V|c$ , тогда наименьший гамильтонов цикл имеет вес  $w_{opt} \geq \frac{w(H)}{c} > |V|$ , значит он содержит хотя бы одно новое ребро, но тогда в исходном графе гамильтонова цикла не было (его вес равен  $|V|$ ).

Таким образом, предъявлен способ решения  $NP$ -полной задачи с помощью полиномиального алгоритма, а значит  $P = NP$ , что противоречит предположению  $P \neq NP$ . ◀

## 5 Алгоритм, дающий 2-приближение для метрической задачи

Для метрической (в отличие от стандартной) задачи коммивояжёра существуют алгоритмы, дающие константное приближение. Для начала рассмотрим алгоритм, дающий 2-приближение, а затем обратимся к алгоритму Кристофидеса [2] и реализуем его.

**Теорема 2** Пусть дан произвольный полный взвешенный метрический граф  $G = (V, E)$  и цикл  $C$ , проходящий по всем его вершинам (возможно, не раз). Тогда граф  $G$  содержит гамильтонов цикл веса  $w_{Ham} \leq w(C)$ .

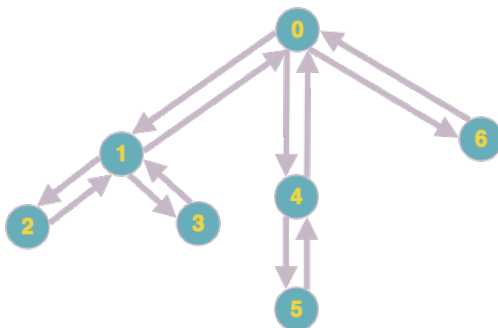
**Доказательство:**

▷ Пусть  $C = \{v_0, \dots, v_k\}$ . Оставим в  $C$  вершины лишь в позициях их первого вхождения. Тогда получим какой-то гамильтонов цикл  $H$  в графе  $G$ . Оценим  $w(H)$ . При выкидывании вершин между  $l$ -ой и  $r$ -ой получили замену рёбер  $(v_l, v_{l+1}), (v_{l+1}, v_{l+2}), \dots, (v_{r-1}, v_r)$  на ребро  $(v_l, v_r)$ . По неравенству треугольника имеем:  $w(v_l, v_r) \leq w(v_l, v_{l+1}) + w(v_{l+1}, v_r) \leq \dots \leq w(v_l, v_{l+1}) + w(v_{l+1}, v_{l+2}) + \dots + w(v_{r-1}, v_r)$ . Отсюда следует, что  $w(H) \leq w(C)$ . ◀

**Обратимся теперь к самому алгоритму:**

1. В данном графе  $G = (V, E)$  найдем остовное дерево минимального веса  $T$  (это делается за полином с помощью алгоритма Краскала или алгоритма Прима).

2. Зафиксируем произвольную вершину  $T$ , обозначим ее  $v_0$ . Подвесим за нее наше дерево и запустим из нее обход в глубину: рекурсивно обходим поддеревья, после чего возвращаемся в предка. Обход работает до тех пор, пока все вершины не будут посещены и обход не вернется в  $v_0$ .



Получим цикл  $K$ , в котором встречаются все вершины нашего графа, причем  $w(K) = 2w(T)$ , так как каждое ребро мы посещаем в момент входа в вершину и момент выхода из нее. Таким образом по [теореме 2](#) получаем, что в нашем графе есть гамильтонов цикл  $H$  такой, что  $w(H) \leq w(K) \leq 2w(T)$ . Построить сам цикл  $H$  мы можем по доказательству [теоремы 2](#).

Какое приближение у нашего алгоритма? Пусть  $\hat{H}$  - гамильтонов цикл минимального веса.  $w(\hat{H}) \geq w(T)$ , так как если убрать любое ребро из  $\hat{H}$ , то получим какое-то остовное дерево  $G$ , а  $T$  - остовное дерево минимального веса. Тогда  $w(\hat{H}) \geq w(T) = \frac{w(H)}{2}$ . Таким образом,  $w(H) \leq 2w(\hat{H})$ , значит наш алгоритм дает 2-приближение метрической задачи коммивояжера.

## 6 Алгоритм, дающий 1.5-приближение для метрической задачи

Рассмотрим алгоритм Кристофидеса<sup>[2]</sup>:

1. Находим минимальное остовное дерево  $T$ .
2. Выделяем в дереве  $T$  подмножество вершин  $N(T)$  нечетной степени. В полном графе  $G$  среди вершин  $N(T)$  ищем совершенное паросочетание  $M$  минимального веса. (делается за полиномиальное время с помощью [blossom-алгоритма](#))
3. Построим Эйлеров граф  $G_E$  на вершинах  $V(G)$  и ребрах  $E(T) \cup E(M)$ . Он Эйлеров, так как у каждой вершины нечетной степени мы добавили по ребру  $\Rightarrow$  степени всех вершин четны  $\Rightarrow$  граф эйлеров.
4. Строим Эйлеров цикл  $K$  в графе из предыдущего пункта. (ему естественным образом соответствует цикл  $U$ , проходящий по всем вершинам)
5. Строим Гамильтонов цикл  $H$  из цикла  $U$  как описывает это [теорема 2](#).

**Теорема 3** *Описанный выше алгоритм Кристофидеса даёт 1.5-приближение для метрической задачи коммивояжера.*

### Доказательство:

▷ Заметим, что справедлива оценка  $w(H) \leq w(U) \leq w(T) + w(M)$ . Рассмотрим гамильтонов цикл  $L$  на вершинах из  $M$ . Аналогично [теореме 2](#)  $w(L) \leq w(\hat{H})$  (где  $\hat{H}$  - гамильтонов цикл в  $G$  минимального веса), причем  $L$  - объединение двух непересекающихся паросочетаний в графе, индуцированном на вершины  $M$ . Вес каждого из них не менее, чем вес минимального паросочетания  $M$ , значит  $2w(M) \leq w(L) \leq w(\hat{H}) \Rightarrow w(M) \leq \frac{w(\hat{H})}{2}$ . Аналогично доказательству приближения прошлого алгоритма,  $w(T) \leq w(\hat{H})$ , значит  $w(H) \leq w(T) + w(M) \leq w(\hat{H}) + \frac{w(\hat{H})}{2} = 1.5w(\hat{H})$ . Значит, алгоритм Кристофидеса даёт 1.5-приближение. ◀

## 7 Реализация алгоритма, дающего 1.5-приближение для метрической задачи

### Реализация:

Имплементируем алгоритм на C++. Проект лежит в репозитории [github](#). Алгоритм декомпозирован на три задачи:

1. Минимальное остовное дерево с помощью алгоритма Краскала <sup>[3]</sup> за время  $\mathcal{O}(N \log N)$ .
2. Совершенное паросочетание минимального веса в полном взвешенном неориентированном графе. Сам алгоритм описан в статье <sup>[4]</sup> В. Колмогорова от 2009 года, реализация <sup>[5]</sup> взята с сайта Institute of Science and Technology, Austria. (время  $\mathcal{O}(N^2 \log N)$ )
3. Эйлеров цикл в неориентированном мультиграфе <sup>[6]</sup> за время  $\mathcal{O}(N^2)$ .

Итоговая асимптотика алгоритма -  $\mathcal{O}(N^2 \log N)$ .

### Тестирование:

В ходе тестирования было сгенерировано 10 тестов с графами различного размера по общему принципу: в случайном порядке соединяются (ребрами веса 1) в гамильтонов цикл вершины. Затем, всем остальным ребрам присваивается случайный вес 1 или 2 (для сохранения неравенства треугольника). Данные по тестам приведены в таблице:

Номер	Число вершин	Точное решение	Решение алгоритма	Ошибка
1	10	10	11	10%
2	120	120	151	26%
3	230	230	294	28%
4	340	340	425	25%
5	450	450	550	22%
6	560	560	693	24%
7	670	670	832	24%
8	780	780	984	26%
9	890	890	1123	26%
10	1000	1000	1248	25%

Как видно из таблицы, на всех тестах алгоритм уложился в отведенные ему 50% ошибки, а средняя ошибка составила примерно 24%.

## 8 Источники

1. William J. Cook: In Pursuit of the Traveling Salesman, 2014.
2. N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
3. Kruskal's algorithm.
4. Vladimir Kolmogorov, Blossom V: a new implementation of a minimum cost perfect matching algorithm, 2009.
5. Blossom5-v2.05
6. E-maxx algo, Нахождение Эйлера пути/цикла.