# Евклидова задача коммивояжёра

Гросул Кирилл

Декабрь 2019

## 1 Введение

Задача коммивояжёра (Travelling salesman problem или TSP) заключается в поиске самого выгодного маршрута, проходящего через заданные города хотя бы по одному разу с последующим возвратом в исходный город. Задачи относится к классу NP-трудных задач, как и большинство её частных случаев. До сих пор не известен ни один алгоритм с полиномиальным временем, который бы гарантировал точность лучшую, чем 1,5 от оптимальной в общем случае, однако в 2010 американские математик Санджив Арора и Джозеф Митчел независимо показали, что, если все расстояния между городами евклидовы, то существует схема полиномиального времени PTAS для поиска приближённого решения, за что были удостоены премии Гёделя.

## 2 Задача

Даны n точек в Евклидовом пространестве. Требуется найти цикл минимальной длины, который проходит через все из них как минимум по одному разу.

## 3 Теорема Ароры

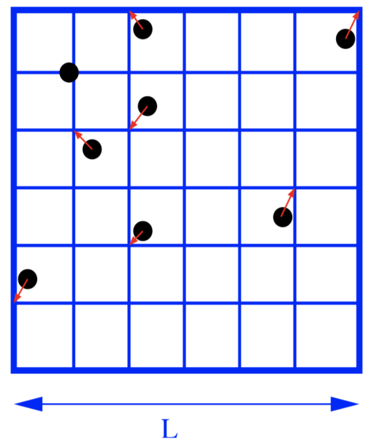
Для любого $\varepsilon > 0$ существует алгоритм, работающий за полиномиальное время, который позволяет найти цикл длины $<= (1+\varepsilon)OPT$, где $OPT$ - длина минимального такого цикла.

## 4 Доказательство

Докажем теорему конструктивно - предъявив алгоритм и обосновав, почему его ошибка укладывается в заданные рамки.

## 4.1 Дополнительные условия

Будем доказывать теорему для $\mathbb{R}^2$. Для удобства скажем, что количество точек $n = 2^t$. Поместим все точки в квадрат минимальной площади. Пусть длина его стороны равна $L = 4n^2$.

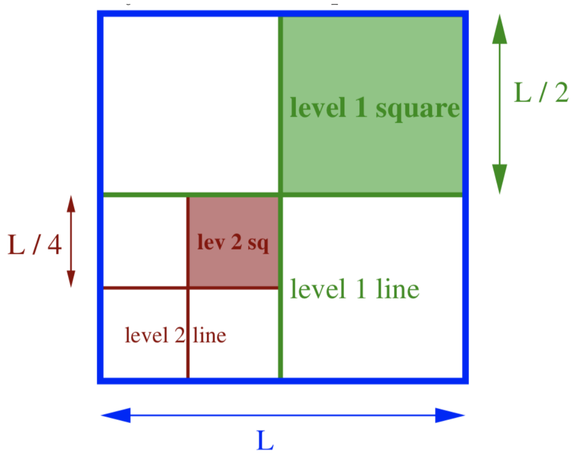



Для этого нам может понадобиться увеличить или уменьшить все расстояния между точками в s раз. Понятно, что при таком преобразовании оптимальный путь переходит в оптимальный. В квадрате введем целочисленную сетку, каждую вершину передвинем в ближаший к ней узел. Надо показать, что при таком преобразовании относительная ошибка укаладывается в заданные ограничания. Заметим, что каждая точка была подвинута на расстояние не превышающее $\sqrt{2}$, а длина оптимального цикла $OPT >= 2L$(как минимум две точки находятся на противоположных стронах выбранного нами минимального квадрата). Тогда относительная ошибка $r >= \frac{\sqrt{2}n}{OPT} >= \frac{\sqrt{2}n}{2L} = \frac{\sqrt{2}n}{8n^2} = \frac{1}{4\sqrt{2}}$. При достачно большом $n$, это велична будет значительно меньше $\varepsilon$.

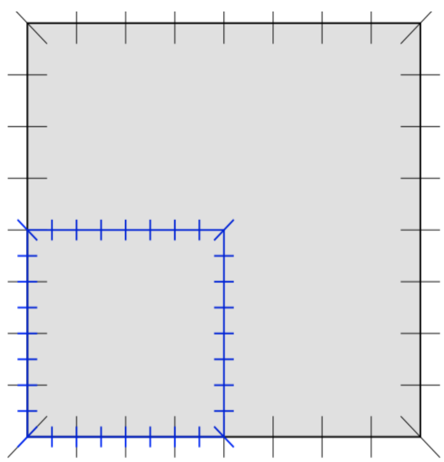
## 4.2 Разбиения и порталы

Благодаря удобному выбору L мы можем разделить наш квадрат на 4 квадрта, каждый из них еще на 4 и так далее как показано на изображении вплоть до единичных квадратов. Удобно думать о таком разбиении как о дереве, каждый узел которого имеет четырех сыновей.

Скажем, что глубина такого дерева равка k, а количство вершин составляет $\frac{4^{k+1}-1}{4-1} = O(n^4)$.




Теперь перейдем к порталам. Для того, чтобы решать поставленную задачу за полиномиальное время будет удобно воспользоваться техникой динамического программирования. Для этого надо ограничить количество мест, в которых путь может пересекать сетку. Назовем такие места порталами. Скажем, что на каждой стороне единичного квадрата m - 1 таких порталов, равноудаленных друг от друга, и еще по 1 в каждой вершине. m выберем в промежутке от $\frac{k}{\varepsilon}$ до $\frac{2k}{\varepsilon}$. Так же потрубуем, чтобы m было степенью 2-ки. Это позволит гарантировать, что портал квадрата уровня i в нашем дереве является так же порталом уровня i + 1.

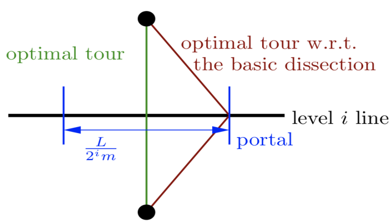


## 4.3 Поиск пути

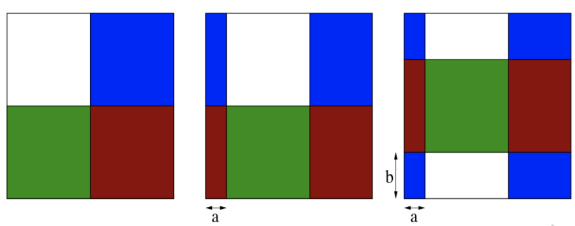
Попробуем найти путь минимальной длины, который посещает все вершины, при этом пересекает сетку только в порталах. Понятно, что такой путь не должен содержать самопересечений, а каждый портал он должен посещать не более 2 раз. Рассмотрим отдельный квадрат. Он содержит 4m порталов, каждый их них может быть посещён 0, 1 или 2 раза. Всего вариантов посещения порталов в одном квадрате $3^{O(m)} = n^{O(\frac{1}{\varepsilon})}$. Так как пути внутри квадрата не должны пересекаться, то все посещения порталов внутри квадрата можно рассматривать как правильную скобочную последовательность. Тогда всевозможные правильных посещений порталов в квадрате с $2r$ порталами - не более $C(r)$ - r-того числа Каталана. $C(r) <= 2^{2r} <= 2^{4m} = n^{O(\frac{1}{\varepsilon})}$. Тогда для каждого портала нам надо рассмотреть $n^{O(\frac{1}{\varepsilon})} * n^{O(\frac{1}{\varepsilon})} = n^{O(\frac{1}{\varepsilon})}$ вариантов. Будем делать это с помощью динамического программирования и подъема по нашему дерево от листьев к корню. Количество вершин уже было оценено, так что сложность алгоритма составит $O(n^4 n^{O(\frac{1}{\varepsilon})})$.

## 4.4 Рандомизация сетки

То, что мы разрешили нашему пути пересекать сетку только в определнных точках, могло сильно ухудшить длину нашего пути. Будем говорить, что линия сетки принадлежит уровню i, если i - минимальный уровень квадратов из разбиения, которые её содержат. Тогда "ухудшение"пути при пересечении линии уровня i составляет не более $\frac{L}{2^i m}$.




Случайно выберем целые числа a и b, такие что $0 <= a, b < L$ и сдвинем сетку на a по горизонтали и на b по вертикали.




Вероятность того, что случайно выбранная линия сетки имеет уровень i равна $p(i) = \frac{2^i}{2^{k+1}-2} = \frac{2^i}{2L-2} <= \frac{2^i}{L}$. Случайная величина X - ошибка, внесенная пересечением случайно выбранной линии. Тогда

$$E(X) = \sum_{i=1}^{k} (\frac{L}{2^i m} \frac{2^i}{L}) = k/m <= \varepsilon$$

Как уже было сказано, OPT - длина оптимального пути. Заметим, что тогла оптимальный путь пересекает линии сетки не более 2OPT раз, обозначим количество этих пересечений за N. Пусть тогда Y - случайная величина, показывающая суммарную ошибку, вызванную пересечениями в нашем пути.

$$E(Y) = N * E(X) <= N\varepsilon <= 2\varepsilon OPT$$

. Используем неравенство Маркова.

$$P(Y >= a) <= \frac{E(Y)}{a}$$

Пусть $a = 4\varepsilon OPT$.

$$P(Y >= 4\varepsilon OPT) <= \frac{2\varepsilon OPT}{4\varepsilon OPT} = \frac{1}{2}$$

Получается, вероятность того, что относительная ошибка внесенная нашим ограничением на пересечение линий сетки с вероятностью $\frac{1}{2}$ не превышает $4\varepsilon$.

# 5 Итог

Мы показали, как с помощью недетерменированного алгоритма найти приближенное решение с точностью не хуже $1 + \varepsilon$. Можно убрать недетерменированность перебрав все возможные значения a и b, тогда итоговая сложность составит $O(n^8 n^{O(\frac{1}{\varepsilon})})$. Понятно, что такой алгоритм скорее теоретический и на практике не применятся. Поэтому мы реализуем алгоритм иммитации отжига.

# 6 Источники


- Euclidean Traveling Salesman Problem, Dominik Schultes:
  http://algo2.iti.kit.edu/schultes/lebenslauf/publications/euclTSPsummary.pdf
- PTAS for Euclidean Traveling Salesman and Other Geometric Problems, Sanjeev Arora