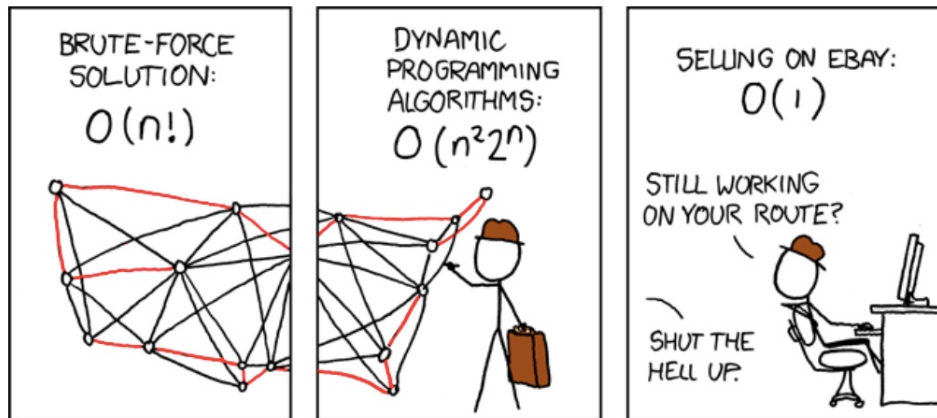


Евклидова задача коммивояжёра

Вадим Плахтинский

Ноябрь 2017



1 Постановка задачи

TSP:

Дан граф $G = (V, E)$ с неотрицательными весами ребер. В этом графе нужно найти Гамильтонов цикл (цикл графа, проходящий по всем вершинам) минимального веса.

Euclidean TSP:

$V \subset \mathbb{R}^k$, а E — это множество всех пар евклидовых расстояний между вершинами графа.

Так как мы будем решать TSP для $k = 2$, то нам надо найти минимальный по весу гамильтонов цикл у n точек на плоскости. Расстояние мы будем понимать, как евклидову метрику: $d(x, y) = \|x - y\|_2$

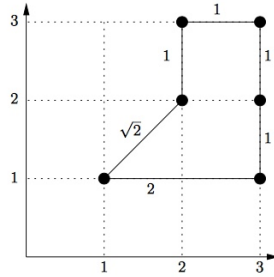


Рис. 1: Стоимость минимального пути составляет $6 + \sqrt{2}$.

2 Приближенное решение

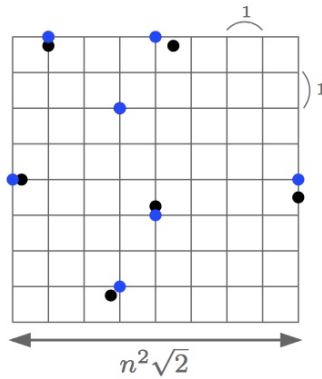
Th. (Aurora, 1996): Для euclidean TSP существует PTAS.

Теорема утверждает, что для $\forall \epsilon > 0$ существует полиномиальный алгоритм, зависящий от n , который вычисляет гамильтонов цикл веса $\leq OPT(1 + \epsilon)$

Доказательство:

1. Изменение координат:

Поместим "решетку" на нашу плоскость с точками. Каждую точку переместим в ближайшую точку решетки. Наименьший квадрат, который содержит все наши точки имеет сторону $n^2\sqrt{2}$. Под действием нашего преобразования каждая вершина переместилась на расстояние $\leq \frac{\sqrt{2}}{2}$. Значит, каждое ребро изменилось не более чем в $\sqrt{2}$ раз. Построим по нему новое оптимальное решение OPT' .



$$h : G \longrightarrow G'$$

Посмотрим как изменилось оптимальное решение для нового графа:

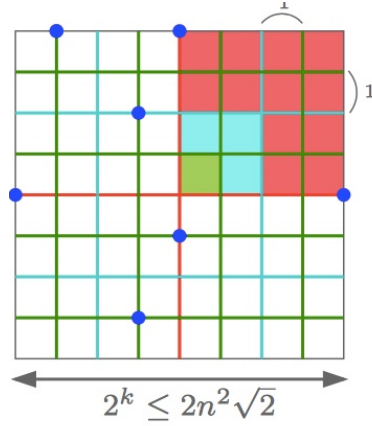
$$|OPT'| \leq h(OPT) \leq |OPT| + n\sqrt{2}$$

$$|h^{-1}(OPT')| \leq |OPT'| + n\sqrt{2} \leq |h(OPT)| + n\sqrt{2} \leq |OPT| + 2n\sqrt{2} \leq |OPT|(1 + \frac{1}{\epsilon})$$

Значит, $h^{-1}(OPT')(1 - \epsilon)$ аппроксимирует OPT при $n \geq \frac{1}{\epsilon}$.

2. Разбиение на квадраты:

Начнем разбивать минимальный квадрат, который охватывает наш граф таким образом, чтобы на каждом уровне предыдущий квадрат разбивался на 4 равных. Таких разбиений сделаем k штук, где k находится из $2^{k-1} \leq n^2\sqrt{2} \leq 2^k$.

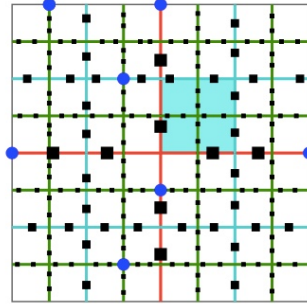


Размер $O(n^4)$

3. Порталы:

$m := \frac{\log(n)}{\epsilon}$. Заметим, что в разбитом квадрате линии можно пересечь только в определенных точках, назовем их порталами. Всего их $2^i m$, а также по одному portalу на каждом узле решетки. На уровне i каждая линия инцидентна 2^i парам квадратов, не считая узлов решетки. Граница каждого квадрата состоит из линий, поэтому учитывая еще и

углы квадратов, получаем, что на каждый квадрат приходится $4m+4$ порталов.



4. Поиск пути:

TODO

3 Алгоритм:

Так как алгоритм Ароры имеет больше теоретическое приложение, чем прикладное, то реализуем муравьиный алгоритм. Реализуем все на питоне, чтобы потом можно было визуализировать.

```

1. Ввод матрицы расстояний D
2. Инициализация параметров алгоритма —  $\alpha, \beta, \kappa, Q$ 
3. Инициализация ребер — присвоение видимости  $\varphi_{ij}$  и начальной концентрации феромона
4. Размещение муравьев в случайно выбранные города без совпадений
5. Выбор начального кратчайшего маршрута и определение  $L^*$ 
//Основной цикл
6. Цикл по времени жизни колонии  $t = 1, t_{\max}$ 
7. Цикл по всем муравьям  $k = 1, m$ 
8. Построить маршрут  $T_k(t)$  по правилу (1) и рассчитать длину  $L_k(t)$ 
9. конец цикла по муравьям
10. Проверка всех  $L_k(t)$  на лучшее решение по сравнению с  $L^*$ 
11. В случае если решение  $L_k(t)$  лучше, обновить  $L^*$  и  $T^*$ 
12. Цикл по всем ребрам графа
13. Обновить следы феромона на ребре по правилам (2) и (3)
14. конец цикла по ребрам
15. конец цикла по времени
16. Вывести кратчайший маршрут  $T^*$  и его длину  $L^*$ 

```

4 Запуски:

1. Сгенерируем набор из 100 случайных точек и посмотрим на время работы.
2. Интересно было бы увеличивать кол-во данных и посмотреть в какой момент мы не получим оптимальное решение. Так же Для неболь-

шого кол-ва точек можно найти оптимальный вес с помощью точных алгоритмов и сравнить с нашим ответом.

3. Воспользуемся данным с [сайта](#). Это координаты городов USA. Данных в [датасете](#) слишком много, поэтому сначала уменьшим их кол-во и посмотрим на результат. В зависимости от времени решим, сможем ли мы запустить на полном наборе.
4. Возможно, что-то еще.

Список литературы

- [1] Sanjeev Arora: Journal of the ACM, 1998
- [2] Sanjeev Arora: Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems, 1996