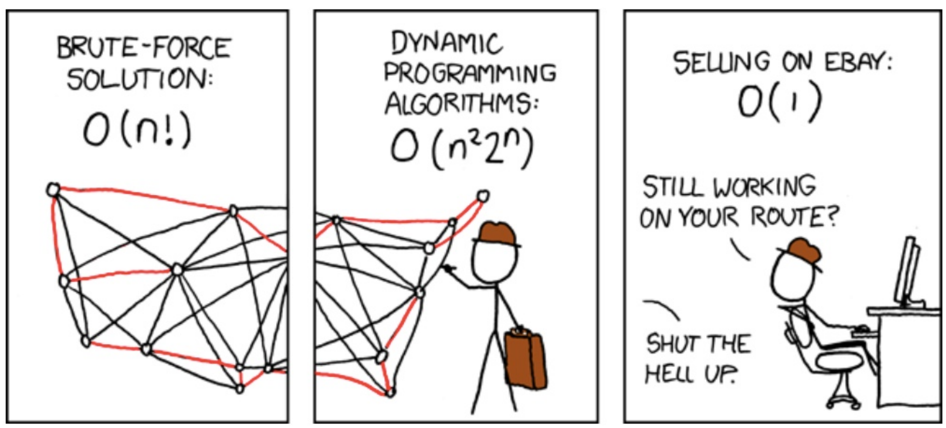
Евклидова задача коммивояжёра

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: $d(x, y) = ||x - y||_2$

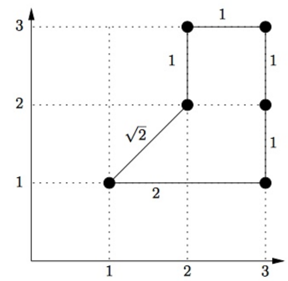


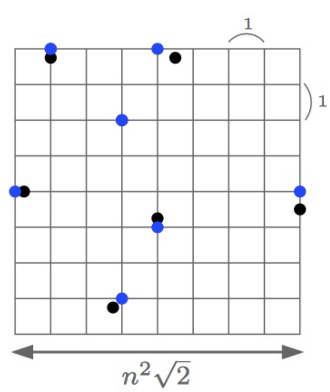
Рис. 1: Стоимость минимального пути составляет $6 + \sqrt(2)$.

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 $\leq OPT(1 + \epsilon)$

Докозательство:

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'.



$h : G \longrightarrow G'$

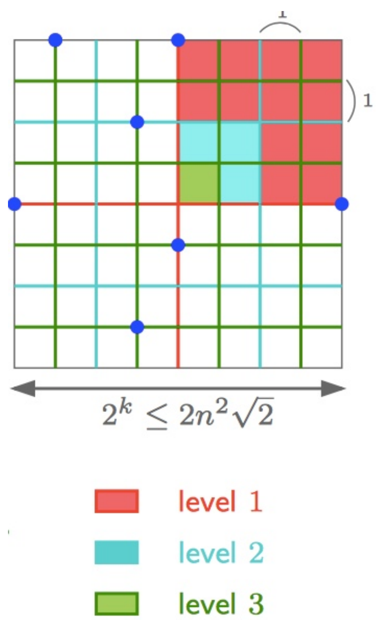
Посмотрим как изменилось оптимальное решение для нового графа:

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

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

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

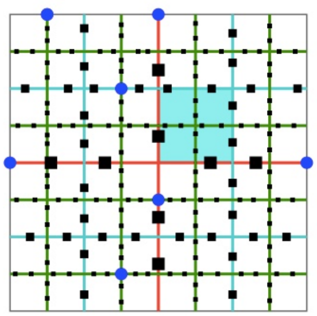
2. **Разбиение на квадраты:**
Начнем разбивать минимальный квадрат, который охватывает наш граф таким образом, чтобы на каждом уровне предыдущий квадрат разбивался на 4 равных. Таких разбиений сделаем k штук, где k находится из $2^{k-1} \leq n^2\sqrt{2} \leq 2^k$.



Размер $O(n^4)$. Получим, что на уровне d будет портал со стороной $\frac{L}{2^d}$. Будем хранить наше разбиение в виде 4-ичного дерева.

3. **Порталы:**
$m := \frac{log(n)}{\epsilon}$. Заметим, что в разбитом квадрате линии можно пересечь только в в определенных точках, назовем их порталами. Всего их $2^i m$, а также по одному порталу 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.



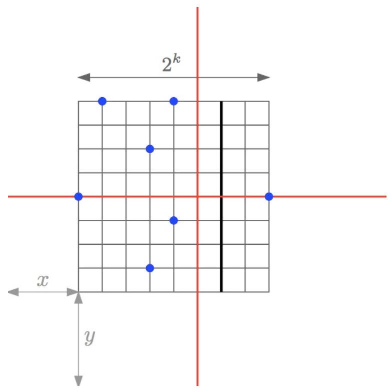
4. **Поиск пути:**
   Наш искомый путь будет проходить через порталы. Понятно, что каждый портал будет посещен не более 2 раз. Наш оптимальный путь для нового графа может пересечь себя только в порталах. Будем искать оптимальный путь с помощью динамического программирования и 4-ичного дерева.
   Для каждого квадрата считаем:

   (a) Кол-во посещений каждого портала, принадлежащего исследуемому квадрату. Из $3^{O(m)} = n^{O(\frac{1}{\epsilon})}$

   (b) Паросочетания между паратлами их будет: $\Omega(m!) = \Omega(n^{logn})$. Понятно, что не будет пересечения внутри квадрата, значит, паросочетания должны образовывать ПСП.

   Оценим размер таблицы: Столбцов будет $O(n^4)$, это кол-во вершин в дереве. Строк будет $n^{O(\frac{1}{\epsilon})}$. Сложность алгоритма булет составлять: $O(n^4 n^{O(\frac{1}{\epsilon})})$

5. **Д-во приближения:** Может получиться так, что OPT' будет намного больше, чем OPT. Для решения этой проблемы рандомизацию.
   Выберем 2 рандомных числа $0 \leq x, y \leq L$, где $L = 2^k$ - длина наибольшего квадрата. И переместим каждую вертикальную линию на x, а горизонтальную на y.



Тогда у каждой лини будет своей уровень.
Оценим вероятность нахождения линии на i-ом уровне.
$P(l$ находится на уровне $i) = \frac{2^{i-2}}{1+2^k}$
$1 + 2^k$ - возможные значения для x,
2^{i-1} - линии $i-$ уровня, половина из которых достигает l (аналогично для y).
X - случайная велечина $=$ уровню линии.

$$E(X) = \sum_{i=1}^{k+1} \frac{2^{i-2}}{1+2^k} \frac{2^{k+1}}{m2^i} \leq \sum_{i=1}^{k+1} \frac{2^{i-1}}{2^k} \frac{2^{k+1}}{m2^i} = \frac{k+1}{2m}$$

OPT пересекает сетку $\leq 2|OPT|$ раз, поэтому

$$|OPT'|-|OPT| \leq \frac{k+1}{m}|OPT| \leq \frac{2logn + \frac{3}{2} + 1}{log\frac{n}{2\epsilon}}|OPT| \leq (4+\frac{5}{logn})\epsilon|OPT| \leq 9\epsilon|OPT|$$

$$2^k < 2n^2\sqrt{2}$$
$$m = \frac{logn}{\epsilon} \geq \frac{logn}{2\epsilon}$$

3 Алгоритм:

Так как алгоритм Ароры имеет больше теоретическое приложение, чем прикладное, то реализуем Генетический алгоритм. Для него надо определить несколько понятий.
Скрещивание: Параметр для приспособленности маршрута. Он равен обратной величене цикла. Чем меньше вес цикла, тем больше параметр приспособленности.

Скрещивание: Пусть у нас есть 2 маршрута. Нам надо по ним построить новый. Для этого к первой половине первого маршрута добавим вторую половину второго, таким образом, чтобы вершины не повторялись.
Мутация: Чтобы получить новый маршрут из данного, достаточно поменять 2 его любые вершины местами.

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