Сложности вычислений, теорема Ароры и Евклидова TSP в $\mathbb{R}^2$

Мешков Владислав Б05-151б

Постановка

Пусть на плоскости даны n точек: V, и пусть $\|n\|_2$ - Евклидова метрика, нужно найти гамильтонов путь на графе $G = (V, V^2, c)$, где $c(u, v) = \|u - v\|_2 \quad \forall u, v \in V$ (веса ребер - это расстояние между инциедентными ему вершинами)

Основные определения и утверждения

Будем считать, что:

- координаты вершин целые
- расстояния между вершинами не менее 8

Будем на данных вершинах рассматривать покрывающий их квадрат со стороной L и его рекурсивное разбиение:

Каждый квадрат (начиная с исходного) делим на 4 части, и храним в виде дерева с выходной степенью каждой не листовой вершины равной 4 (т.е. 4ричное корневое дерево)

Закончим разбиение на моменте, когда размер квадрата станет меньше 1 и он содержит не более 1 вершины

Кол-во листовых квадратов $= O(L^2)$, а значит глубина дерева $O(logL)$

Данное разбиение квадрата на более мелкие назовем рассечение исходного квадрата

Также назовем OPT - длину максимального пути

Определение

Пусть $a, b \in [0, L)$, (a, b)-смещение рассечения, определяемое как смещение x, y координат всех линий на a, b соответственно, и взяие результата по модулю L

Определение

Пусть m, r - положительные целые числа

(m, r)-регулярное множество порталов для смещения (a, b) - это множество точек на сторонах квадратов в них.

Каждый квадрат имеет портал в угле и m других порталов равноудаленных друг от друга на сторонах

Замечание Будем считать, что наш путь искомый проходит через входные вершины и некоторые порталы, при этом порталы можно посещать несколько раз

Определение : путь назовем (m, r)-простым, относительно смещения (a, b), если пересекает каждое ребро каждого квадрата не более r раз и всегда через портал

Опредление : Пусть между вершинами u, v, проходит через порталы $P_1, P_2, ..., P_k$, т.е. имеет вид $u - P_1 - ... - P_k - v$

Говорим, что ребро (u, v) изогнуто в точках $P_1, ..., P_k$

Замечание : Предполагаем, что вершины не лежат на границах областей в рассечении, чтобы добиться этого, масштабированием расстояния между вершинами, и полагая, что вершины имеют нечетные координаты, а линии рассечения - четные.

Замечание : В будущем получим алгоритм, со стоимостью $(1 + \epsilon)OPT$, с изогнутыми ребрами - их по надобности можно выровнять

Структурная Теорема

Теорема

Пусть $c > 0$. Пусть минимальное расстояние между вершинами не менее 8

Пусть L - размер ограничивающего их квадрата. Пусть также $(a, b) \in [0, L]$ - случайные числа из данного отрезка. Тогда с вероятностью не менее $\frac{1}{2}$, тогда есть решение данной задачи, стоимостью не более $(1 + \frac{1}{c})OPT$ и он является (m, r)-простым со смещением $(a, b$, тут $m = O(clogL)$, $r = O(c)$

Шаги алгоритма:

Преобразуем координаты

Пусть L_0 - размер стороны квадрата, ограничивающего точки V
И пусть OPT - размер оптимального Гамильтонова цикла в графе
Разобьем данный квадрат на подквадраты размера $\frac{L_0}{8c}$ (т.е. сделаем точность сетки равную $\frac{L_0}{8c}$
Теперь переместим каждую вершину в ближайшитй узел данной сетки

Заметим, что сейчас, стоимость любого пути между вершинами, отличается от такого же пусти в исходном графе (такого же по порядку прохода вершин) не более чем в $2n\frac{L_0}{8nc} = \frac{L_0}{4c} \leqslant \frac{OPT}{4c}$
А значит аппрокимация сеткой меняет оптимальное расстоние не более чем в $\frac{OPT}{4c}$ раз
Теперь делим расстояния на $\frac{L_0}{64c}$, то координаты в сетке становятся целыми с минимальным расстонием между в-ми не менее 8
Также постановим, что $L = O(nc)$ - размер ограничивающего квадрата(стороны)
Теперь осталось найти $(1 + \frac{3}{4}c)$-аппрокимацию вместо $(1 + \frac{1}{c})$-аппрокимации
Чтобы не тащить константу, и будем исать $1 + \frac{1}{c}$-аппрокимацию чтобы прийти к $\frac{3}{4}$ достаточно применить рассуждения для $c' = \frac{3}{4}c$

Строим смещенное дерево квадратов

Получаем смещение (a, b) случайно
Считаем дерево с этим смещением
$L = O(n)$ - рзмер покрывающего квадрата, высота смещенного дерева $O(logn)$
Число квадратов: $O(nlogn) = O(n)log(n^2)$ (в дереве квадрантов, в отличии от нашего, кол-во листьев $O(n)$

DP

Пусть $m = O(clonn)$, $r = O(c)$
С помощью динамического программирования будем искать оптимальный $(m, r)-$простой путь с

поощью смещенного дерева квадрантов
Пусть S - квадрат нашего дерева и лучший (m, r)-простой пусть, переекающий границу $2p \leqslant 4r$ раз.
Также пусть $a_1, ..., a_{2p}$ -п-ть порталов, которые он прошел в соотв. порядке
И пусть путь проходит через все вершины из S, и путь (составленный из $a_1, ..., a_{2p}$) - (m, r)-простой
Вход:

- Непустой квадрат смещенного дерева
- мультимножество порталов размера не более r порталов на каждом из 4 сторон квадрата, причем муммарный размер: $2p \leqslant 4r$
- Пары $(a_1, a_2), ..., (a_{2p-1}, a_{2p})$ на данных порталах

Вместе пути соединяют все вершины из квадрата
Строим DP, в которой на (k_m, k_r)-м месте записан оптимальный (k_m, k_r)-простой путь
Тогда на (m, r) записан ответ на нашу задачу в корне при $p = 0$

Число элементов таблицы очевидно равно числу разных случаев, т.е. $O(T(m+4)^{4r}(4r)!)$, где T-число число непустых квадратов

Инициализируем в листах, т.е. там где есть хоть 1 вершина и там же не более $O(r)$ порталов, решается за $O(r)$ (перебираем все $O(r)$ возможностей разместить узел)
Предположим, что алгоритм верно решил задачу для (m, r) для квадратов глубины $> i$, и пусть S - это квадраты глубины i
S_1, S_2, S_3, S_4-дети S
Алгоритм перебирает все варианты, когда (m, r)-простой путь пересечет стороны S_1, S_2, S_3, S_4
Стороны S_1, S_2, S_3, S_4
Итак перебираем:

- мультимножество из $\leqslant r$ порталов на 3 внутренних ребрах S_i-х (про них все знаем т.к. уже считали): $O((m+4)^r)^4$
- порядок в котором порталы из пред пнкта посещались оптимальным (m, r)-протым путем $O((4r)^{4r}(4r)!)$

Замечание: множитель в 2 части $((4r)^{4r})$ возник как оценка на вариантов выбора для каждого порала (один из $\leqslant 4r$ порталов, через которые он идет)

Все перебраные варианты пользуются какими-либо решениями задачи на детях
Проверив все варианты алгоритм скажет ответ: оптимальный путь

Таким образом время работы: $O(T(m+4)^{8r}(4r)^{4r}(4r)!) = O(n(logn)^{O(c)})$

Замечание:

Для наших квадратов с округлением все даже проще (ко-во вариантов меньше)