

Санкт-Петербургский государственный университет
Факультет прикладной математики - процессов управления

Работа по эмпирическому анализу алгоритма
построения дерева Гомори-Ху из взвешенного графа

Выполнила:
Кильдякова Ю. А.
группа 18.Б13-пу

Санкт-Петербург
2020

Содержание:

1. Краткое описание алгоритма.....	3
2. Математический анализ алгоритма.....	3
3. Входные данные.....	3
4. Генерация входных данных.....	4
5. Реализация алгоритма.....	4
6. Вычислительный эксперимент.....	4
7. Характеристики использованной вычислительной среды и оборудования.....	5
8. Источники.....	5

1. Краткое описание алгоритма

Задачи разбиения графа — это семейство задач комбинаторной оптимизации, в которых граф должен быть разбит на две или более частей с дополнительными ограничениями.

s - t -разрезом графа называется разбиение вершин графа на два непересекающихся множества S и T , такие что $s \in S$, $t \in T$

Алгоритм, опубликованный Гомори и Ху в 1961 году, применяется для нахождения минимальной пропускной способности s - t -разрезов для всех пар вершин s, t в неориентированном графе G с пропускными способностями ребер $u: E(G) \rightarrow \mathbb{R}^+$. А дерево Гомори-Ху - взвешенный остоновый граф, веса ребер которого содержат информацию о минимальных пропускных способностях s - t -разрезов для всех пар вершин s, t графа.

С помощью этой структуры данных можно решать задачу о наименьшем k -разрезе, которая имеет приложения при разработке сверхбольших интегральных схем, интеллектуальном анализе данных и информационном обмене при параллельных вычислениях.

Сам алгоритм можно найти в источнике [1] стр. 221.

2. Математический анализ алгоритма

n - количество вершин в графе.

m - количество ребер.

Время работы алгоритма доминируется $(n-1)$ -кратным вычислением минимального s - t -разреза (равного максимальному s - t -потoku), которое можно выполнить с помощью алгоритма Диница за время $O(n^2m)$. Все остальные действия можно выполнять за время $O(n^3)$.

Таким образом мы получаем оценку алгоритма $O(n^3m)$.

3. Входные данные

Количество вершин n : возьмем диапазон $[20; 200]$ с шагом 20.

Количество ребер m возьмем равное n . Таким образом на таких входных данных сложность должна быть порядка $O(n^4)$.

Пропускные способности ребер (веса): целые числа в промежутке $(0; 9]$.

Единица измерения трудоемкости: время выполнения алгоритма в секундах, измеренное с помощью библиотеки `time` для языка `python`.

4. Генерация входных данных

На вход подается количество вершин n .

Создается матрица смежности графа, вершины, соединенные ребрами, и веса выбираются случайно.

Матрица смежности преобразуется к классу `Graph` библиотеки `NetworkX`, который используется в алгоритме построения дерева.

5. Реализация алгоритма

Реализация алгоритма взята из библиотеки `NetworkX` (источник [2]). Для поиска минимального разреза используется алгоритм `dinitz`, также реализованный в этой библиотеке.

Код представлен в репозитории на `github`:

<https://github.com/JuliaKil/empirical-analysis>

6. Вычислительный эксперимент

Алгоритм выполняется `repeats=10` раз для каждого значения n из диапазона входных данных. На каждом повторении генерируется новый граф с количеством вершин и ребер n . Вычисляется среднее время работы для каждого значения n .

Результаты эксперимента представлены и описаны вместе с программным кодом в файле `gomory_hu.ipynb`.

7. Характеристики использованной вычислительной среды и оборудования.

Процессор: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz

Тип системы: 64-разрядная

Вычислительная среда: Jupyter Notebook

8. Источники

1. Корте Б., Фиген Й. Комбинаторная оптимизация. Теория и алгоритмы. стр. 218-224
2. Software for Complex Networks — NetworkX 2.5 documentation
<https://networkx.org/documentation/stable/index.html>
3. Gomory-Hu Tree <https://www.geeksforgeeks.org/gomory-hu-tree-introduction/>
4. Левитин А. В. Алгоритмы: введение в разработку и анализ алгоритмов. стр. 127