Сложность вычислений
Алгоритмический проект

Феофанова Мария, 697 группа

25 декабря 2018 г.

1 Постановка задачи.

Поиск наименее плотного разреза. Требуется разбить вершины графа на 2 группы, так чтобы отношения числа рёбер между ними к размеру меньшей группы было как можно меньше. Имплементируйте алгоритм Лейтона–Рао приближённого решения с точностью $O(\log n)$.

2 Общий случай. Задачи с несколькими товарами.

В своей статье Лейтон и Рао определяют и доказывают результаты для более общего случая, нежели в постановке нашей задачи. Заметим, что, как известно из частного случая, минимальный разрез напрямую связан с максимальным потоком. При решении обобщенной задачи, как выяснится далее, подобная связь сохранится, пусть и не в таком сильном виде. Далее мы рассмотрим, какие же более общие задачи были изучены в статье, результаты для них и связь максимального потока с наименее плотным разрезом.

Задачи с несколькими товарами (MFP). В отличие от привычного определения задачи нахождения максимального потока, рассматриваются задачи с несколькими ($k \geq 1$) товарами, которые необходимо доставить из истока s_i в сток t_i. Также, у каждого товара есть спрос D_i, который означает, что необходимо доставить D_i единиц товара.

В связи с определением возникает проблема: а как же определять максимальный поток, учитывая, что источников может быть больше одного? Авторы остановились на определении, старающемся максимизировать общую долю f каждого товара.

Определение. *Максимальный поток* в MFP - это такое максимальное число f, что для каждого i из истока в сток может быть доставлено fD_i единиц

товара одновременно со всеми остальными товарами, без превышения пропускных способностей ребер.

Определение. *Минимальный разрез* в MFP - это такое Ф для графа с множеством вершин V, что:

$$\Phi = \min_{U \subseteq V} \frac{C(U, \overline{U})}{D(U, \overline{U})}, \tag{1}$$

где

$$C(U, \overline{U}) = \sum_{e \in <U, \overline{U}>} C(e), \tag{2}$$

$$D(U, \overline{U}) = \sum_{\{i \,|\, s_i \in U \text{ and } t_i \in \overline{U} \text{ or } t_i \in U \text{ and } s_i \in \overline{U}\}} D_i, \tag{3}$$

То есть, это минимальное отношение пропускной способности разреза к сумме спроса тех товаров, исток и сток которых лежат по разные стороны от разреза. Нетрудно заметить, что такое определение является обобщением определения максимального потока для случая, когда товар и спрос один.

Теперь покажем, что максимальный поток всегда не больше минимального разреза. Допустим, $i_1, ..., i_r$ - множество индексов тех товаров, исток и сток которых разделены некоторым разрезом. По определению максимального потока, мы знаем, что

$$\sum_k f D_{i_k} \leq C(U, \overline{U}), \tag{4}$$

$$\sum_k D_{i_k} = D(U, \overline{U}), \tag{5}$$

Откуда следует, что

$$f \leq \frac{C(U, \overline{U})}{D(U, \overline{U})}, \tag{6}$$

3 Общий случай. Равномерный вариант задачи. Результат.

Для того, чтобы добиться логарифимческого приближенного решения, авторы сужают постановленную ранее задачу.

Определение. *UMFP* - сужение задачи MFP, в которой для любой пары вершин существует товар, и спрос D_i всех товаров одинаков. Б.о.о $D_i = 1$.

Для такой задачи был получен результат: минимальный разрез в UMFP не больше чем в $O(\log n)$ раз больше максимального потока. Также существует пример, на котором достигается такая оценка.

Заметим, что при такой постановке задачи D - это просто произведение мощностей множеств, полученных разрезом (для каждого элемента-истока из одного множества все элементы из другого множества являются стоками):

$$D(U, \overline{U}) = |U||\overline{U}|, \tag{7}$$

Отсюда, минимальный разрез для UMFP:

$$\Phi = \min_{U \subseteq V} \frac{C(U, \overline{U})}{|U||\overline{U}|}, \tag{8}$$

4 Теорема о минимальном разрезе в UMFP.

Далее мы изложем доказательство того факта, что минимальный разрез максимум в порядка $\log n$ раз больше, чем максимальный поток. То есть, будет доказано, что:

$$\frac{C(U, \overline{U})}{|U||\overline{U}|} \leq O(f \log n), \tag{9}$$

где f - максимальный поток в графе.
Учитывая предыдущие результаты, сформулируем главную теорему:

Теорема 1. Для любого UMFP, верно следующее:

$$\Omega(\frac{\Phi}{\log n}) \leq f \leq \Phi \tag{10}$$

Авторы статьи при доказательстве теоремы ссылаются на другой результат: на задачу, двойственную к MFP. А именно, это задача линейного программирования, где требуется отыскать такие расстояния $d(u, v)$, что

$$\sum_i D_i d(s_i, t_i) \geq 1 \tag{11}$$

И при этом следующая сумма минимизирована:

$$\sum_e C(e) d(e) \tag{12}$$

В частном случае UMFP:

$$\sum_{u,v} d(u, v) \geq 1 \tag{13}$$

$$W = \sum_e C(e) d(e) \tag{14}$$

Оказывается, из теории линейного программирования следует, что $f = W$. Таким образом, мы свели часть задачи к решению задачи линейного программирования. Как известно, существуют полиномиальные решения таких задач. Дальнейшие выкладки помогут построить минимальный разрез с помощью W.

Лемма 1. Для любого графа G, с произвольными пропускными способностями ребер и функцией расстояния (с суммарным весом W), а также для любого $\delta > 0$, существует такое разбиение графа на компоненты радиуса не

больше δ, так что пропускная способность ребер, соединяющих разные компоненты, будет не больше $4W \log n/\delta$.

Доказательство Леммы 1. Пусть $C = \sum_{e \in E} C(e)$ - *суммарная пропускная способность* графа G. Если $\delta \leq 4W \log n/C$, мы можем выделить каждую вершину в отдельную компоненту. Тогда пропускная способность между разными компонентами $\leq C \leq 4W \log n/\delta$.

Рассмотрим случай $\delta > 4W \log n/C$. Для дальнейших рассуждений построим новый граф G^+ следующим образом: заменим каждое ребро в G на цепочку из $\lceil Cd(e)/W \rceil$ ребер. Пропускную способность каждого такого ребра сделаем равной $C(e)$, а расстояние 1. Далее будет показано, как сформировать нужные нам компоненты в G^+ и выделить их в исходном графе.

Алгоритм формирования компонент: Сначала выбираем произвольную вершину v, соответствующую какой-то вершине из G. Определим G_i^+ - подграф G^+, содержащий все вершины и ребра на расстоянии не больше i от вершины v. Нетрудно заметить, что, так как длины ребер в G^+ равны 1, то расстояние между двумя вершинами в нем это просто количество ребер в кратчайшем пути между ними. Также обозначим за $C_0 = 2C/n$, а для $i > 0$ обозначим за $C_i = \sum_{e \in G_i^+} C(e)$. Найдем такое наименьшее j, что $C_{j+1} < (1+\epsilon)C_j,\ \epsilon = W \log n/(\delta C)$. Такое j точно найдется, поскольку с какого-то момента $C_i = C_{i+1}$, так как граф конечен. Найденный подграф G_j^+ формирует первую компоненту. Остальные компоненты найдем рекурсивно, удаляя ребра и вершины от предыдущей компоненты из графа G^+, пока не исчерпаем все такие v, которые лежат и в исходном графе G.

Пусть $C^+ = \sum_{e \in G^+} C(e)$. Тогда, согласно определению G^+:

$$C^+ = \sum_{e \in E} C(e) \lceil \frac{Cd(e)}{W} \rceil \leq \sum_{e \in E} C(e)(1 + \frac{Cd(e)}{W}) = C + C/W \sum_{e \in E} C(e)d(e) = C + C = 2C \tag{15}$$

Пропускная способность ребер, выходящих из любой компоненты C_i, согласно неравенству $C_{j+1} < (1+\epsilon)C_j$:

$$C_{i+1} - C_i < \epsilon C_i \tag{16}$$

Посчитаем суммарную пропускную способность ребер, выходящих из компонент. У нас есть 2 вида компонент: C_i, когда $i \neq 0$, и C_0. Во втором случае компонента просто состоит из 1 вершины, значит, таких может быть максимум n. В первом - суммарно они не больше C^+ по определению:

$$\epsilon(\sum_{i \geq 1} C_i + nC_0) \leq \epsilon C^+ + \epsilon n C_0 \leq \epsilon 2C + \epsilon 2C = \epsilon 4C \tag{17}$$

Теперь построим компоненты в G. Поместим все вершины из G, лежащие в одной компоненте в G^+, в одну компоненту в G. Тогда, заметим, что пропускная способность ребер, выходящих из компонент одинакова для обоих графов (и там и там разрезается ребро e, причем в G^+ разрезается "подребро"на нем,

которое по построению имеет ту же пропускную способность $C(e)$). Значит, суммарная пропускная способность в G ребер, выходящих из компонент, такая же: $4\epsilon C \leq 4W \log n/\delta$. Это доказывает первую часть теоремы.

Осталось показать, что все компоненты радиуса не больше δ.

Зафиксируем j и рассмотрим компоненту с этим радиусом в G^+. Исходя из способа построения компонент, понятно, что пропускная способность ребер в компоненте будет не меньше $(1+\epsilon)^j C_0 = (1+\epsilon)^j (2C/n)$ Поскольку пропускная способность всего графа, как мы доказали выше, не больше $2C$, можно составить и решить следующее неравенство (в последнем пользуемся неравенством на логарифм с тем, что $\epsilon \leq 1/4$):

$$(1+\epsilon)^j (2C/n) \leq 2C \tag{18}$$

$$j \log(1+\epsilon) \leq \log n \tag{19}$$

$$j \leq \frac{\log n}{\log(1+\epsilon)} \leq \frac{\log n}{\epsilon} \tag{20}$$

Любой путь в G^+ длиной l соответствует пути в G длиной не больше Wl/C. Объединяя оба неравенства, получим, что радиус любой компоненты не больше

$$\frac{W \log n}{C\epsilon} = \delta \tag{21}$$

, ч.т.д.

Далее сформулируем несколько утверждений, которые приведут нас к решению задачи.

Следствие 1. Для любого графа G и функции расстояния с суммарным весом W, возможно одно из двух:

- Найдется компонента радиуса не больше $1/(2n^2)$, которая содержит $2/3$ вершин графа;
- Найдется разрез размера $O(W \log n)$.

Лемма 2. Для любого графа G и функции расстояния с суммарным весом W и множества вершин T, содержащим не менее $2/3$ вершин графа, такого, что

$$\sum_{u \in V-T} d(T,u) \geq 1/(2n) \tag{22}$$

Существует разрез размером $O(W)$.

Лемма 3. Для любого графа G и функции расстояния с суммарным весом W, существует разрез размера $O(W \log n)$

Доказательство Теоремы 1 напрямую следует из Леммы 3 и того факта, что $W = f$.

Доказательства оставшихся лемм и следствия было решено не включать в работу, так как они менее содержательны, чем Лемма 1, основываются на похожих заключениях, и, к тому же, их всегда можно найти в источнике, указанном в Литературе.

Поскольку в доказательствах утверждений используются явные построения, можно имплементировать явный полиномиальный алгоритм который находит ответ с заданными свойствами.

5 Сведение полученных результатов к задаче из условия.

Заметим, что все это время решалась немного не та задача: нахождение минимума (8). Тогда как в условии нашей задачи стояло нахождение минимума

$$\Phi_1 = \min_{U \subseteq V} \frac{C(U, \overline{U})}{\min(|U|, |\overline{U}|)}, \tag{23}$$

Покажем, что, на самом деле, результаты применимы и к такой постановке. Заметим, что:

$$\Phi_1 = \max(|U|, |\overline{U}|)\Phi \tag{24}$$

И, так как $n/2 \leq \max(|U|, |\overline{U}|) \leq n$, то:

$$\frac{n}{2}\Phi \leq \Phi_1 \leq n\Phi \tag{25}$$

Откуда

$$\Omega(\frac{\Phi_1}{n \log n}) \leq f \leq \frac{2}{n}\Phi_1 \tag{26}$$

6 Описание алгоритма, построенного по теоремам.

Соединив все вместе, получится следующий алгоритм.

1. Решаем задачу линейного программирования нахождения функции расстояний для $C(e) = 1$;
2. Находим с помощью Леммы 1 компоненты для $\delta = 1/(2n^2)$;
3. По Следствию 1, находим либо нужный разрез (тогда мы закончили), либо большую компоненту, содержащую $2/3$ узлов;
4. Если разрез еще не найден, по Лемме 2 строим разрез $O(W)$;

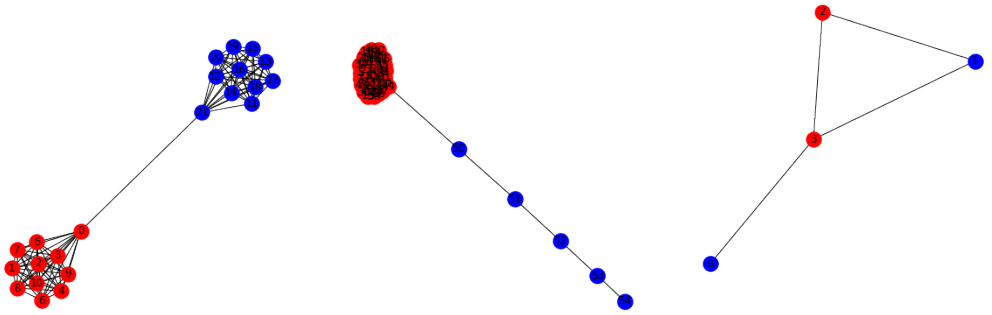
7 Анализ, результаты для частных примеров.

Я реализовала получившийся алгоритм на **Python3**, использовав библиотеки **networkx** для графов и **pulp** для линейного программирования.

К сожалению, как выяснилось, алгоритм очень часто падает в частный случай на простых, небольших графах, выделяя каждую вершину в компоненту и затем равномерно распределяя их по 2 кучи примерно равного размера. Ему может часто везти при разделении, но так происходит не всегда. Можно сделать вывод, что такой алгоритм не подходит, если требуется решать задачи для небольших графов, да и еще достаточно точно.

Однако, нашлись примеры, на которых алгоритм заходит и в нетривиальные ветви. Например, граф **Гантелька** хорошо насыщен ребрами при относительно малом количестве вершин, из-за чего неравенство из Леммы 1 все-таки выполнилось. Хорошо видно, как алгоритм сумел отделить две основные части графа друг от друга.

Еще один пример - это **Леденец**. Там алгоритм еще и, ко всему прочему, нашел большую ($>2/3$) компоненту. Как видим, снова получился оптимум.



(a) Гантелька. (b) Леденец. (c) Плохой пример на простом графе.

Рис. 1: Найденные разрезы.

8 Литература.

- Tom Leighton and Satish Rao. 1999. Multicommodity Max-Flow Min-Cut Theorems and Their Use in Designing Approximation Algorithms **[link]**