

# Проект по теме

## "Задача о наименьшем мультиразрезе"

Надежда Виденеева

### 1 Введение

В рамках данного проекта будет доказана **NP**-трудность задачи о наименьшем мультиразрезе и приведен один из приближенных алгоритмов ее решение, работающий за полиномиальное время. Реализацию алгоритма и таблицы с данными (время работы и точность результата) можно найти здесь: <https://github.com/VNVid/ComplexityTheory> Список использованной литературы располагается в конце документа, картинки заимствованы из [1], [2].

### 2 Постановка задачи

#### Задача о наименьшем мультиразрезе

Пусть задан взвешенный неориентированный граф  $(G = (V, E), w : E \rightarrow \mathbb{R}_+)$ , в котором отмечены вершины  $s_1, \dots, s_k$ . Требуется найти наименьший мультиразрез (minimum multicut), т. е. множество рёбер  $E_0$  наименьшего суммарного веса, такое что отмеченные вершины находятся в разных компонентах  $(V, E \setminus E_0)$ .

#### Задание

1. Докажите, что задача поиска наименьшего мультиразреза является **NP**-трудной при  $k \geq 3$ ;
2. Докажите, что задача поиска мультиразреза, имеющего вес не больше удвоенного наименьшего, решается за полиномиальное время (есть и более точные алгоритмы);
3. Имплементируйте получившийся алгоритм и проанализируйте его работу.

### 3 NP-трудность задачи

Докажем, **NP**-трудность для частного случая: пусть теперь граф  $G$  планарен и его вершины имеют степень не больше 3. Пусть также дано некоторое число  $B$ ,

и требуется найти мультиразрез суммарного веса не больше  $B$ . Будем сводить к нашей задаче модифицированный язык PLANAR 3-SAT. Рассмотрим двудольный граф, вершины одной доли которого соответствуют переменным в формуле, а другой - скобкам-дизъюнкциям, ребра соединяют вершины-переменные с вершинами-скобками, в которых есть соответствующие литералы. PLANAR 3-SAT состоит из выполнимых формул в 3-кнф, для которых описанный граф планарен. Мы же позволим скобкам содержать не только 3, но и 2 литерала, но потребуем, чтобы каждая переменная содержалась ровно в 3х скобках, причем оба литерала для переменной должны присутствовать (т.е. и  $x$ , и  $\bar{x}$ ).

Для сведения нам понадобятся два гаджета: для переменной и для скобки с дизъюнкцией.

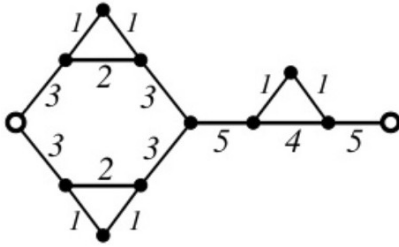


Рис. 1: Гаджет для переменной.

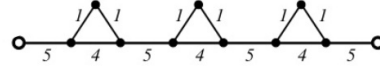


Рис. 2: Гаджет для скобки.

Левый конец гаджета для переменной на рис.1 соответствует тому литералу, который встречается в двух скобках, а правый - тому, что в одной (т.е. если для некоторой переменной  $x$  литерал  $x$  присутствует в двух скобках, а  $\bar{x}$  в одной, то левая вершина будет отвечать за  $x$ , а правая - за  $\bar{x}$ ). В гаджете для скобок каждый треугольник с весами на ребрах (1, 1, 4) отвечает за свой литерал (значит, если в скобке только 2 литерала, то в гаджете будет 2 треугольника).

Посмотрим на рис.3, чтобы понять, как мы будем соединять гаджеты для скобок с гаджетами для переменных.

Синим контуром выделен гаджет для скобки  $(x_3 \vee \bar{x}_4 \vee x_7)$ , причем литералы  $x_3, \bar{x}_4, x_7$  встречаются во всей формуле два раза (т.к. соединяются в своих гаджетах с двумя треугольниками с ребрами весом (1, 1, 2)). Таким образом, гаджеты соединяются ребром веса 2 через вершины маленьких треугольников, которым инцидентны ребра с весом 1.

В построенном графе вершинами  $s_i$ , которые мы хотим отделить (назовем их терминальными), будут крайние вершины гаджетов (левые и правые концы на рис.1 и 2). Все вершины имеют степень не больше 3, и по построению граф получился планарным. В качестве числа  $B$  возьмем число  $10n + 4m$ , где  $n$  - количество переменных,  $m$  - количество скобок в формуле.

Пусть теперь у формулы есть некоторый выполняющий набор, пойдем какие

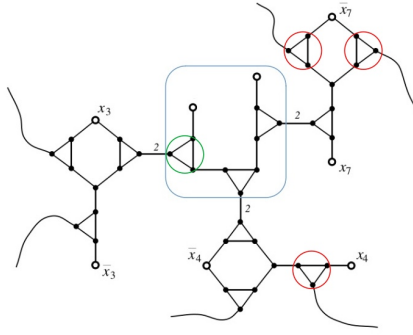


Рис. 3: Связь переменных и скобок.

ребра войдут в искомый разрез. Из каждого гаджета для переменной возьмем треугольник(и), отвечающий(е) за литерал, НЕ соответствующий значению в выполняющем наборе. Например, обратимся к рис.3, если  $x_4 = 0$ ,  $x_7 = 1$ , то в разрез войдут ребра треугольников, выделенных красным. В каждом гаджете для скобки удалим треугольник, соответствующий какому-нибудь литералу, принимающему значение 1. На рис.3. зеленым выделен треугольник, который можно включить в разрез, если  $x_3 = 1$  в выполняющем наборе. Таким образом мы уже убрали пути между концами каджетов по самим гаджетам. Разделим еще гаджеты между собой, включив в разрез соединяющие их ребра (веса 2), которые еще прикреплены к обоим гаджетам (т.е. не удалили ни один треугольник из гаджетов, вершины которых соединяет ребро). Итого, из каждого гаджета будет удалено либо одно ребро веса 4, либо два веса 2, и из каждого соединения гаджетов (рис.4) либо соединяющее ребро веса 2, либо два ребра из треугольника веса 1. Причем соединений всего  $3n$ , так как каждый треугольник гаджета для переменной соединен с каким-то гаджетом для скобки (следует из предположения, что каждая переменная встречается ровно в трех скобках).

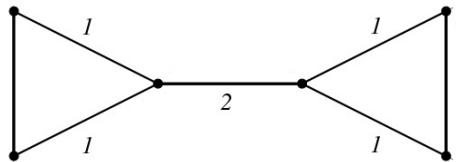


Рис. 4: Место соединения гаджетов.

Таким образом, получаем, что если формула имеет выполняющий набор, то

найдем нужный разрез. В другую сторону тоже верно.

## 4 Приближенный полиномиальный алгоритм

Введем понятие изолирующего разреза для одной терминальной вершины: это такой разрез, после удаления которого данная вершина будет недостижима из всех оставшихся терминальных вершин. Минимальный изолирующий разрез - это изолирующий разрез с минимальным суммарным весом ребер. Алгоритм заключается в поиске минимального изолирующего разреза для каждой терминальной вершины. Эти разрезы вместе, кроме самого тяжелого, образуют искомый мультиразрез.

### Точность алгоритма

Суммарный вес ребер найденного разреза не превосходит вес лучшего больше, чем в 2 раза.

*Док-во.*

Пусть  $E$  - лучший мультиразрез, удаление которого из графа  $G$  разбивает его на компоненты  $V_1, \dots, V_k$ , где  $V_i$  - компонента содержащая одну терминальную вершину  $s_i$ . Пусть  $E_i$  - те ребра разреза  $E$ , которые ведут из  $V_i$  в  $V \setminus V_i$ . Таким образом каждый  $E_i$  является изолирующим разрезом для  $s_i$  и  $E = \cup E_i$ .

Выполняется следующее равенство  $\sum_{i=1}^k w(E_i) = 2w(E)$ , так как каждое ребро разреза в сумме слева считается дважды. Б.о.о. предположим, что  $E_k$  является самым "тяжелым" изолирующим разрезом, тогда:  $\sum_{i=1}^{k-1} w(E_i) \leq 2 \frac{k-1}{k} w(E) < 2w(E)$ .

Наш алгоритм нашел минимальные изолирующие разрезы  $C_i$  для терминальных вершин, значит, для всех вершин  $w(C_i) \leq w(E_i)$ . И тогда  $\sum_{i=1}^{k-1} w(C_i) \leq \sum_{i=1}^{k-1} w(E_i) < 2w(E)$ , ч.т.д.

### Реализация и анализ

Чтобы найти минимальный изолирующий разрез для одной терминальной вершины, можно перестроить граф, объединив все остальные терминальные вершины в одну новую, и запускать на нем какой-нибудь алгоритм поиска минимального разреза между данной терминальной вершиной и новой. Я использовала алгоритм Диница ([3]).

Было проведено три группы тестов. Первый состоял из маленьких графов с произвольным количеством ребер и произвольными весами. Как и ожидалось, алгоритм работает значительно быстрее наивного. В подавляющем большинстве случаев он дает ответ очень близкий к правильному (с точностью  $>90\%$ ).

В книге [1] был представлен пример графов, являющихся узким местом для данного алгоритма. Данные графы представляют собой цикл из  $k$  вершин, к которым

присоединены еще  $k$  терминальных вершин (см. рис.5). Ребра в цикле имеют вес 1, а ребра, ведущие из цикла в терминальные вершины - чуть меньше двух. Правильным ответом является  $k$  (минимальный мультиразрез состоит из ребер цикла). Реализованный мной алгоритм показал 100% точность.

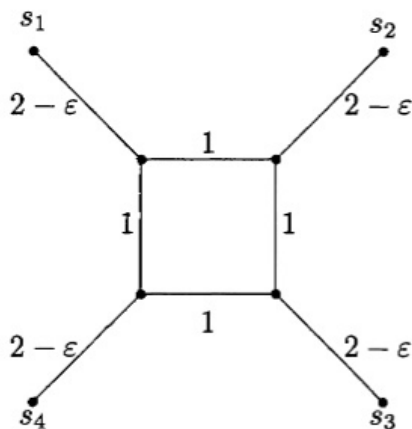


Рис. 5:

И еще одна группа тестов состояла из циклов с произвольным весом ребер. Здесь были случаи, в которых на практике алгоритм давал результат, отличающийся от лучшего больше, чем в 2 раза. Причем в этих случаях терминальных вершин было либо очень мало, либо все вершины графа являлись терминальными. Также можно заметить, что чем больше вершин в графе, тем точнее результаты.

## Список литературы

- [1] V. V. Vazirani. Approximation Algorithms
- [2] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts.
- [3] Yefim Dinitz. Dinitz' Algorithm: The Original Version and Even's Version