Модел. и анализ информ. систем. Т. **21**, № **4** (2014) 54–63

УДК 519.174.1

Устойчивость в задаче поиска минимального разреза в графе

Козлов И. В.[1]

Московский физико-технический институт
141700, Московская область, г. Долгопрудный, Институтский пер., 9.

e-mail: volokno@inbox.ru

получена 2 мая 2014

Ключевые слова: устойчивость, минимальный разрез

Задача комбинаторной оптимизации называется устойчивой, если ее решение сохраняется при возмущении входных параметров, не превышающих некоторого порогового значения – радиуса устойчивости. В работах [1–3], в предположении об устойчивости входа, построены точные полиномиальные алгоритмы для некоторых NP-трудных задач о разрезах.

В настоящей работе показано, как строить ускоренные алгоритмы для достаточно устойчивых полиномиальных задач. Подход иллюстрируется на примере известной задачи о минимальном разрезе (MINCUT). Построен $O(n^2)$ точный алгоритм решения n-устойчивой задачи MINCUT. Кроме того, построен полиномиальный алгоритм вычисления радиуса устойчивости задачи MINCUT и получен простой критерий n-устойчивости.

Постановка задачи. Обзор существующих подходов

Исследование устойчивости в задачах дискретной оптимизации имеет большое значение: если вход задачи является устойчивым, то ее решение совпадает с решением всех задач, входы которых незначительно отличаются от исходного. В данной работе рассмотрим устойчивость в задаче поиска минимального разреза во взвешенном неориентированном графе.

Задача поиска минимального разреза (MINCUT). Задан граф $G = (V, E)$, $|V| = n,\ |E| = m$ и симметрическая функция весов $w\colon V \times V \to \mathbb{R}^+,\ \ w(x, x) = 0$. Требуется найти разрез $(S, \overline{S}),\ S \subset V$ минимального веса $f(S) = \sum_{a\in S, b\in \overline{S}} w(a, b) \to \min$.

В этом определении разрез задается множеством вершин S, но далее в тексте будет встречаться понятие разреза как множества ребер между S и $\overline{S}$. Эти определения разреза эквивалентны и будут употребляться в тексте в зависимости от контекста. Задавая функцию весов не на множестве ребер E, а на всей матрице

[1]Работа выполнена при частичной поддержке гранта РФФИ №11-01-00398.

$V \times V$ мы получаем возможность задавать весь вход задачи через функцию весов w. Если какие-то 2 вершины u и v в графе $G = (V, E)$ не соединены ребром, то $w(u, v) = 0$.

Задача MINCUT также может быть рассмотрена на ориентированном графе, но в данной работе мы не будем подробно останавливаться на этом. Заметим лишь, что следует отличать задачу MINCUT в ориентированном графе от другой известной задачи поиска минимального разреза в сети. Последнюю также будем называть поиском минимального s-t разреза, т.к. в ней имеется дополнительное условие, что пара заданных вершин s и t должны находиться по разные стороны от разреза. В задаче MINCUT мы не ставим такого ограничения.

Впервые рассматриваемая задача была решена за полиномиальное время в работе Форда и Фалкерсона [4], где была доказана дуальность между задачей поиска минимального s-t разреза и задачей поиска максимального потока в сети. Для того чтобы найти минимальный разрез в графе, можно построить s-t разрезы для всех возможных пар s и t. Таким образом, задачу MINCUT можно решить за время $O(n^2 \cdot F)$, где F — время нахождения максимального потока в сети. В 1961 году Гомори и Ху в своей работе [5] показали, как находить все возможные s-t разрезы за время $O(n \cdot F)$. В работе Карзанова и Тимофеева [7] показано, как найти все минимальные разрезы в графе за время $O(\lambda \cdot n^2)$, где λ — величина минимального разреза. Позднее Хао и Орлин [6] показали, как произвести $n-1$ вычисление максимального потока в графе, необходимое для нахождения минимального разреза, за время, сравнимое с вычислением одного максимального потока, т. е. решить задачу MINCUT за время $O(F)$.

Параллельно с этим были разработаны методы, не опирающиеся на решение задачи о максимальном потоке и позволяющие решать задачу MINCUT быстрее, чем за $O(F)$. Одним из наиболее перспективных подходов оказалось использование процедуры слияния вершин. Впервые она была использован в работе Поддерюгина [8], где минимальный разрез в графе находится за время $O(mn)$. Позднее данный метод получил развитие в работах Каргера [9], где представлен вероятностный алгоритм, который находит точное решение задачи MINCUT с высокой вероятностью за время $O(n^2 \log^3 n)$, и Штор–Вагнера [10], в которой изложен детерминированный алгоритм с трудоемкостью $O(n(m + n \log n))$. Алгоритм, который будет представлен в следующем разделе, превосходит по скорости эти алгоритмы, которые в свою очередь быстрее, чем алгоритм Карзанова–Тимофеева, т.к. λ в его оценке трудоемкости может быть сколь угодно велика для взвешенного графа, о котором идет речь в данной работе.

Алгоритм, который будет представлен в данной работе, также основан на принципе слияния вершин и превосходит по скорости все вышеперечисленные алгоритмы. Он находит точное решение задачи MINCUT за время $O(n^2)$, если вход задачи является n-устойчивым. Покажем, что понимается под устойчивостью в задаче поиска минимального разреза в графе.

Для пары непересекающихся подмножеств вершин $A, B \subset V$ обозначим множество ребер между ними как $E(A, B) = \{ab \mid a \in A, b \in B\}$ и суммарный вес ребер этого множества как $w(A, B) = \sum_{uv \in E(A,B)} w(u, v)$. Также обозначим $\tau_w(A) = \tau(A) = w\left(A, \overline{A}\right)$ и $\mu_w(A) = \mu(A) = w(A, V)$. Пусть зафиксирован разрез $\left(S, \overline{S}\right)$, тогда через $\xi(A, S)$ обозначим суммарный вес ребер разреза, исходящих из множества

A, т. е. $\xi(A,S) = \sum_{uv\in E(A,\overline{A})\cap E(S,\overline{S})} w(u,v)$, а через $\iota(A,S) = \tau(A) - \xi(A,S)$ — суммарный вес ребер, исходящих из A и не лежащих в разрезе. Для вершины $A = \{v\}$ и ребра $e = uv$ будем писать $\tau(v)$ и $\tau(e)$ соответственно. Если минимальный разрез единственный, будем писать вместо $\xi(A,S)$ и $\iota(A,S)$ просто $\xi(A)$ и $\iota(A)$.

Пусть $w\colon V\times V \to \mathbb{R}^+$ вход задачи и $\gamma \geq 1$. Вход $w'\colon V\times V \to \mathbb{R}^+$ назовем γ-возмущением w, если $\forall u,v \in V,\ w(u,v) \leq w'(u,v) \leq \gamma w(u,v)$. Вход w назовем γ-устойчивым, если существует разрез, остающийся минимальным при всех γ-возмущениях w' входа w.

Определение. Радиусом устойчивости задачи MINCUT назовем максимальное $\gamma \geq 1$, при котором вход задачи является γ-устойчивым.

Это определение радиуса устойчивости было сформулировано в недавних работах Билу и Линиала [1, 2] и является частным случаем более общего определения, предложенного В.К. Леонтьевым [11, 12]. Данные определения отличаются в случае, когда в задаче есть несколько минимальных разрезов. Согласно определению выше, если в задаче MINCUT вход w_0 таков, что имеется несколько оптимальных решений, то мы можем для каждой пары минимальных разрезов увеличить вес ребер, принадлежащих одному из них, но не принадлежащих другому, оставив, таким образом, минимальным только один из них. Поэтому радиус устойчивости задачи для входа w_0 равен 1. В определении Леонтьева вход w_0 является γ-устойчивым, если при всех γ-возмущениях не возникает новых оптимальных решений и описанное выше возмущение, в общем случае, не нарушает его, поэтому радиус устойчивости может быть и больше единицы. В дальнейшем, устойчивый вход задачи с $\gamma > 1$ означает что минимальный разрез единственный.

Подходы, описанные в работах [1,2], позволяют, в случае когда вход задачи является устойчивым с коэффициентом γ порядка $\Omega(\sqrt{n})$, решать NP-трудную задачу максимального разреза за полиномиальное время. В недавней работе Макарычевых [3] этот результат улучшен и получен алгоритм, возвращающий точное решение для $\gamma \geq c\sqrt{\log n}\log\log n$ или выдающий ответ о том, что вход задачи не является γ-устойчивым с достаточно большим параметром γ.

В данной работе показано, что и для задач из класса P дополнительное предположение об устойчивости входа дает возможность получить более быстрые алгоритмы, чем в общем случае.

План статьи:

В разделе 1 представлен квадратичный по числу вершин алгоритм решения n-устойчивой задачи MINCUT. В разделе 2 построен полиномиальный алгоритм вычисления радиуса устойчивости задачи MINCUT и получен быстро проверяемый критерий устойчивости для достаточно больших коэффициентов γ, сравнимых с числом вершин в графе. Некоторые открытые задачи перечислены в разделе 3.

1. Алгоритм поиска минимального разреза

Вначале опишем процедуру слияния вершин, которая будет использоваться нашим алгоритмом. Вначале выбирается ребро $e = uv$. Затем оно удаляется, а инцидентные ему вершины u и v сливаются в новую вершину w. Ребра, инцидентные w — все ребра графа, инцидентные u или v. Получившийся мультиграф будем обозначать G/uv. Алгоритм, который будет изложен далее, является детерминированным и находит точное решение задачи MINCUT, в случае, когда веса ребер графа положительные (не обязательно целые) и граф обладает n-устойчивостью. Теперь опишем алгоритм, а потом докажем его корректность и оценим трудоемкость.

Алгоритм 1

Вход: n-устойчивый граф $G = (V, E, w)$.
Выход: минимальный разрез в графе G и его вес.

1. Пока $|V_G| \geq 2$
   (a) Взять 2 произвольные вершины v и z, выбрать максимальное по весу ребро e, инцидентное хотя бы одной из них;
   (b) $G \leftarrow G/e$;
2. Вернуть минимальный разрез и его вес.

Доказательство корректности алгоритма будет основано на следующих утверждениях, близких к утверждениям из работы [2], для задачи поиска максимального разреза.

Утверждение 1. Пусть $w : V \times V \to \mathbb{R}^+$ вход задачи MINCUT и $\gamma \geq 1$. Тогда вход w является γ-устойчивым с минимальным разрезом $(S, \overline{S})$ тогда и только тогда, когда выполнено $\gamma \cdot \xi\,(A, S) \leq \iota\,(A, S)$ для всех $A \subset V,\ A \neq S$.

Доказательство. Докажем от противного. Предположим, что существует такое множество вершин A, для которого $\gamma \cdot \xi\,(A, S) > \iota\,(A, S)$. Пусть $A \cap S = B,\ A \cap \overline{S} = C$, тогда $S = B \cup (S \backslash B)\,,\ \overline{S} = C \cup (\overline{S} \backslash C)$. Изменим разрез следующим образом: те ребра, которые исходят из A ($e \in E(A, \overline{A})$) и не включены в разрез — добавим в него, а те из них, что входят в разрез, исключим из него. После такого преобразования в графе будет разрез $\big((S \backslash B) \cup C,\ (\overline{S} \backslash C) \cup B\big)$. Любое изменение разреза мы можем записать подобным образом для некоторого множества A. Заметим, что поскольку для данного A справедливо условие $\gamma \cdot \xi\,(A, S) > \iota\,(A, S)$, в случае если вес ребер, входящих в $E(A, \overline{A}) \backslash E(S, \overline{S})$ оставить прежним, а веса остальных ребер увеличить в γ раз, то получившийся разрез будет иметь вес меньше исходного. Таким образом, мы получили противоречие с тем, что вход задачи являлся γ-устойчивым. ■

Утверждение 2. Пусть w — γ-устойчивый вход задачи MINCUT, $\gamma > 1$, а w' получается из w слиянием двух вершин, лежащих по одну сторону от минимального разреза в w. Тогда w' также является γ-устойчивым и его минимальный разрез состоит из ребер, входящих в минимальный разрез в w.

Доказательство. От противного. Пусть это не так, тогда после слияния каких-то двух вершин образуется разрез, вес которого может быть меньше исходного при увеличении веса определенных ребер. Но заметим, что если получившийся разрез не содержит новых ребер, образовавшихся после слияния, то он мог быть получен и до слияния, что противоречит тому, что вход w является γ-устойчивым. Если же в новый разрез входят новые ребра, то мы можем в старом графе построить разрез, в который входит пара ребер, сумма весов которых равна весу нового ребра. Таким образом, мы снова получили противоречие с γ-устойчивостью входа. ■

Покажем теперь корректность представленного алгоритма, т. е. тот факт, что две вершины, которые мы сливаем, всегда находятся по одну сторону от разреза.

Вначале выберем одну произвольную вершину $v \in V$ и рассмотрим ребро vu, имеющее наибольший вес среди ребер инцидентных v. Тогда $w(v,u) \geq \frac{1}{n-1}\tau(v)$. Если вершина v не совпадает с множеством S, то из утверждения 1 следует неравенство $\xi(v) \leq \frac{1}{n+1}\tau(v)$ и, как следствие, получаем $w(v,u) > \xi(v)$. Поэтому вершины v и u находятся по одну сторону от разреза. Но поскольку мы не знаем заранее минимальный разрез, мы не можем сказать, совпадает вершина v с множеством S или нет.

Для того чтобы выбрать вершину, не совпадающую с множеством S, возьмем сразу две произвольные вершины v и z. Рассмотрим все инцидентные им ребра и выберем среди них ребро с максимальным весом. Без ограничения общности, пусть это будет ребро vu. Покажем тогда, что вершина v не совпадает с множеством S и, как следствие, вершины v и u находятся по одну сторону от разреза. Пусть это не так, тогда ребро vu лежит в минимальном разрезе, а поскольку вход задачи n-устойчивый, то $w(v,u) \leq \tau(v) = f(v) \leq \tau(z)/n$. Получаем противоречие с тем, что ребро vu максимальное по весу, среди инцидентных вершинам v и z, т.к. это эквивалентно $w(v,u) \geq \frac{\tau(z)}{n-1}$. В конце работы алгоритма останется две вершины x и y, каждая из которых была получена в результате слияния какого-то множества вершин исходного графа S и $\overline{S}$ соответственно. Минимальный разрез будет представлять собой множество $(S,\overline{S})$ ребер между ними. Вес минимального разреза будет равен сумме весов ребер между x и y. Таким образом, доказана корректность предложенного алгоритма.

Оценим трудоемкость алгоритма. Минимальный разрез находится за $O(n^2)$ операций: поиск ребра с максимальным весом, инцидентного одной из двух вершин, происходит за $O(n)$, слияние двух вершин также требует $O(n)$ и все это повторяется $O(n)$ раз, пока количество вершин в графе не станет равным двум. Эта оценка сравнима с числом ребер в плотном графе, а потому в общем случае не может быть уменьшена.

Заметим, что представленный алгоритм основывается на жадной процедуре, на каждом шаге выбирается локальный максимум. В этом заключается его сходство с алгоритмом Штор–Вагнера [10], где также используется жадная процедура: MA-ordering, после которой происходит слияние двух последних вершин в очереди. Алгоритм Штор–Вагнера находит минимальный разрез в любом взвешенном неориентированном графе, без предположения об устойчивости входа. Но его трудоемкость $O(n(m+n\log n))$ больше, чем у алгоритма описанного выше, за счет того, что процедура MA-ordering выстраивает в очередь все вершины графа, в то время как в случае n-устойчивого входа нам достаточно рассмотреть соседей пары вершин.

2. Нахождение радиуса устойчивости задачи. Критерий устойчивости

Данный раздел посвящен нахождению радиуса устойчивости задачи, т.е. минимального значения γ, при котором вход γ-устойчив. Основное утверждение, которое здесь доказывается, следующее:

Теорема 1. Пусть задан $w: V \times V \to \mathbb{R}^+$ вход задачи MINCUT, тогда радиус устойчивости может быть найден за время $O(n^2m)$.

Доказательство. Сначала опишем дерево минимальных разрезов Гомори–Ху.

Пусть T — дерево на множестве вершин V графа G, причем ребра дерева не обязательно содержатся в E. Множество ребер исходного графа будем далее называть E_G, а множество ребер дерева — E_T. При удалении любого ребра $(u, v) \in E_T$ дерево распадается на две компоненты связности, т. е. существует дуальность между ребрами дерева и разрезами в исходном графе G. Будем говорить, что разрез в G, разделяющий его вершины на компоненты связности, образующиеся при удалении ребра (u, v) из дерева T, *соответствует* ребру (u, v).

При заданном графе G деревом Гомори–Ху будем называть дерево $T(V, E_T, w_T)$, удовлетворяющее следующим условиям:

1) для каждого ребра $e \in E_T$ вес ребра $w_T(e)$ равен весу разреза, соответствующего ребру e в графе G;

2) для каждой пары вершин $s, t \in V$ вес минимального s-t разреза в G равен весу минимального s-t разреза в T.

Очевидно, что минимальный s-t разрез в T соответствует ребру e с минимальным весом на единственном пути P_{st} между вершинами s и t, и разрез, соответствующий e, является и s-t разрезом в G. Каждое ребро дерева Гомори–Ху соответствует разрезу и его вес равен пропускной способности соответствующего разреза. Таким образом, в $n-1$ ребре дерева Гомори–Ху содержится информация о величинах всех $\binom{n}{2}$ минимальных s-t разрезов в G.

Как было сказано ранее, алгоритм построения дерева Гомори–Ху имеет трудоемкость $O(nf)$, где f — трудоемкость решения задачи поиска максимального потока в сети. В работе [5] для поиска максимального потока в сети использовался алгоритм Форда–Фалкерсона. Но можно применить более быстрые алгоритмы [13, 14]. Первый из них используется, когда число ребер сравнимо с числом вершин в графе, а второй — когда число ребер в графе достаточно велико. Тогда трудоемкость поиска максимального потока в сети получится равной $O(nm)$, а трудоемкость построения дерева Гомори–Ху составит $O(n^2m)$.

Утверждение 3. Для нахождения радиуса устойчивости достаточно перебрать ребра дерева T.

Доказательство. Докажем это от противного. Пусть минимальный разрез s в графе равен $(S, \overline{S})$, а радиус устойчивости задачи равен γ. Тогда существует такой разрез $r = (R, \overline{R})$,что при увеличении в γ раз весов ребер, входящих в разрез s, но не входящих в r ($\Delta_{SR} = E\left(S, \overline{S}\right) \setminus E\left(R, \overline{R}\right)$), веса разрезов сравняются: $w'(S) = w'(R) = w(R)$. Докажем, что либо разрез r порождается одним из ребер дерева Гомори–Ху, либо существует разрез из дерева Гомори–Ху с весом меньшим, чем $w(R)$ после увеличения в γ раз весов ребер из Δ_{SR}.

Рассмотрим разбиение $(R, \overline{R})$ относительно дерева Гомори–Ху T. Между R и $\overline{R}$ находится одно или более ребер T. Если между компонентами связности лежит единственное ребро, то разрез $r = (R, \overline{R})$ порождается данным ребром дерева Гомори–Ху. Если же между компонентами связности находится несколько ребер T, то это означает, что разрез r не порождается никаким ребром дерева Гомори–Ху, но тогда каждому пересеченному ребру соответствует свой разрез, веса которых до увеличения весов ребер меньше веса r. Тогда поскольку в γ раз увеличиваются только веса ребер из Δ_{SR}, то хотя бы для одного из порожденных разрезов $q = (Q, \overline{Q})$ будет выполнено $w'(Q) < w'(R)$, что доказывает предположение. ■

Опишем теперь алгоритм вычисления радиуса устойчивости.

Алгоритм 2
Вход: Граф $G = (V, E, w)$.
Выход: γ — радиус устойчивости задачи MINCUT.

1. Построить по входу w дерево Гомори–Ху T;
2. Найти в T ребра e_{Tmin} с минимальным весом;
3. Если $|e_{Tmin}| > 1$
   (a) Вернуть $\gamma = 1$.
4. Иначе найти разрез $(S, \overline{S})$, соответствующий ребру e_{Tmin}, его вес $\tau(S) = w_T(e_{Tmin}) = A$;
5. Для всех остальных ребер дерева $e \in E_T, e \neq e_{Tmin}$:
   (a) Построить соответствующий разрез
   $$(S', \overline{S'}),\ \tau(S') = B_{S'} > A$$
   (b) Найти множество ребер, входящих в минимальный разрез, но не входящих в $(S', \overline{S'})$:
   $$\Delta_{SS'} = E(S, \overline{S}) \backslash E(S', \overline{S'})$$
   (c) Рассчитать $\gamma_{S'} = \frac{B_{S'} - A + w(e)}{w(\Delta_{SS'})}$;
6. Вернуть $\gamma = \min\limits_{S'} \gamma_{S'}$.

Корректность данного алгоритма следует из утверждения 3: нам достаточно перебрать все ребра дерева T и рассчитать для каждого из них, во сколько раз необходимо увеличить часть ребер минимального разреза так, чтобы он перестал быть минимальным. Оценим теперь его трудоемкость. Имея построенное дерево Гомори–Ху, радиус устойчивости задачи ищется уже за квадратичное по числу вершин время. Построение разреза по ребру дерева осуществляется за $O(n^2)$ времени, разница двух разрезов вычисляется за линейное по числу ребер в разрезах время, т. е. тоже за $O(n^2)$, определение коэффициента γ по множеству ребер происходит за линейное время. Таким образом, трудоемкость вычисления радиуса устойчивости задачи равна трудоемкости построения дерева Гомори–Ху, т.е. $O(n^2 m)$. Теорема 1 доказана. ■

2.1. Критерий устойчивости

Кроме того, оказывается возможным быстро осуществить проверку входа на n-устойчивость (а именно этот случай рассматривался в предыдущем разделе).

Рассмотрим ребра, инцидентные каждой из вершин графа, в порядке возрастания весов. Поскольку для γ-устойчивого графа в задаче MINCUT выполняется условие $\gamma \cdot \xi(A) \leq \iota(A)$ для любого множества вершин $A \subset V$, то это будет справедливо и для каждой вершины по отдельности. Тогда $\xi(v) \leq \frac{1}{\gamma+1}\tau(v)$, следовательно, все ребра с весом, большим $\frac{1}{\gamma+1}\tau(v)$, гарантированно не принадлежат минимальному разрезу. Теперь, для того чтобы по матрице весов проверить граф на γ-устойчивость, выделим для каждой вершины все ребра, вес которых превышает $\frac{1}{\gamma+1}\tau(v)$. Вершины исходного графа и выделенные ребра будем называть тяжелым графом. Если тяжелый граф представляет собой 2 компоненты связности, то исходный граф очевидно является γ-устойчивым. Мы получили достаточное условие, нужно понять, является ли оно необходимым.

В случае если γ достаточно мало, это условие не является необходимым. Для примера рассмотрим $\left(\frac{n}{2}-1\right)$-устойчивый граф, состоящий из двух полных подграфов на $\frac{n}{2}$ вершинах и одного ребра, соединяющего две произвольные вершины из разных подграфов, веса всех ребер которого равны единице. Он, очевидно, является также γ-устойчивым для любого $\gamma < \left(\frac{n}{2}-1\right)$. Но если $\gamma < \left(\frac{n}{2}-2\right)$, то мы должны будем выделить ребра, вес которых превышает $\frac{1}{\gamma+1}\tau(v) > \frac{1}{\left(\frac{n}{2}-2\right)+1}\left(\frac{n}{2}-1\right) = 1$. Веса всех ребер в графе единичные, поэтому не будет выделено вообще ни одного ребра.

Посмотрим теперь, будет ли данное условие необходимым, когда γ достаточно велико, а именно рассмотрим случай $\gamma = n$. Заметим, что если на некотором входе достаточное условие выполняется, то и задачу о минимальном разрезе для него можно решить за время $O(n^2)$. (Поскольку проверка условия требует также $O(n^2)$ операций). Осталось теперь понять, возможна ли ситуация, когда построенный по исходному графу тяжелый граф состоит более чем из двух компонент связности.

Утверждение 4. Граф является n-устойчивым тогда и только тогда, когда тяжелый граф состоит ровно из двух компонент связности.

Доказательство. Достаточность данного условия показана выше. Докажем необходимость методом от противного. Вначале покажем, что если исходный граф n-устойчивый, то тяжелый граф не может быть связным. В самом деле, если тяжелый граф связный, то минимальный разрез исходного графа содержит хотя бы одно ребро тяжелого. Заметим теперь, что ребра тяжелого графа по построению гарантированно не принадлежат разрезу. Получили противоречие.

Теперь осталось доказать, что если исходный граф n-устойчивый, то тяжелый граф не может состоять из трех или более компонент связности. Если такое произошло, то выберем любые три компоненты связности тяжелого графа и посмотрим на ребра исходного графа, лежащие между ними. Данные ребра будут образовывать три различных разреза в исходном графе, которые отличаются друг от друга не более чем в n раз. Выберем такую тройку компонент связности, что один из этих трех разрезов будет минимальным в исходном графе. Тогда если увеличить в n раз вес ребер, входящих в минимальный разрез, но не входящих в остальные, то данный разрез перестанет быть максимальным. Мы получаем противоречие с предположением об n-устойчивости графа. Утверждение доказано. ■

3. Заключение. Открытые вопросы

В данной работе была рассмотрена полиномиальная задача поиска минимального разреза в графе и показано, что при дополнительном предположении о достаточно большой устойчивости входа разрез находится быстрее, чем в общем случае. Кроме того, для рассматриваемой задачи был предложен полиномиальный алгоритм нахождения радиуса устойчивости и критерий устойчивости для одного частного случая.

Дальнейшие направления исследования связаны с получением аналогичных результатов для схожих задач на графах, которые упоминались ранее: поиск минимального разреза в ориентированном графе, поиск минимального разреза и максимального потока в сети, поиск минимального k-разреза и т.д. Также открытым вопросом остается построение ускоренных алгоритмов для рассмотренных полиномиальных задач с качественно меньшим радиусом устойчивости. Кроме того, интересно было бы получить структурное описание для достаточно устойчивых входов.

Список литературы

1. *Bilu Y., Linial N.* Are stable instances easy? // Innov. in Comp. Sci. 2010. P. 332–341.

2. *Linial N. et.al.* On the practically interesting instances of MAXCUT // STACS. 2013. P. 526–537.

3. *Makarychev K., Makarychev Y., Vijayaraghavan A.* Bulu-Linial stable instances of max cut and minimum multiway cut // Proc. of the 25-th ACM-SIAM Symp. on Disc. Alg. 2014. P. 890–906. DOI: 10.1137/1.9781611973402.67

4. *Ford L., Fulkerson D.* Maximal flow through a network // Canadian Journal of Mathematics. 1956. № 8. P. 399–404.

5. *Gomory R., Hu T.* Multi-terminal network flows // Journal of the Society of Industrial and Applied Mathematics. Dec.1961. P. 551–570.

6. *Hao J., Orlin J.* A faster algorithm for finding the minimum cut in a directed graph // Journal of Algorithms. Nov.1994. Vol. 17, № 3. P. 424–446.

7. *Карзанов А.В., Тимофеев Е.А.* Эффективный алгоритм нахождения всех минимальных реберных разрезов неориентированного графа // Кибернетика. 1986. № 2. С. 8–12. (English transl.: *Karzanov A.V., Timofeev E.A.* Efficient algorithm for finding all minimal edge cuts of a nonoriented graph // Kibernetika. 1986. № 2. P. 8–12.)

8. *Поддерюгин В.Д.* Алгоритм для нахождения реберной связности графа // Вопросы кибернетики. 1973. № 2. С. 136. (English transl.: *Podderyugin V.D.* An algorithm for finding the edge connectivity of graphs // Vopr. Kibern. 1973. № 2. P. 136.)

9. *Karger D., Stein C.* A new approach to the minimum cut problem // Journal of the ACM. Jul.1996. Vol. 43, № 4, P. 601–640.

10. *Stoer M., Wagner F.* A simple min-cut algorithm // Journal of the ACM. Jul. 1997. Vol. 44, № 4. P. 585–591.

11. *Леонтьев В.К.* Устойчивость задачи коммивояжера // Журн. вычисл. матем. и матем. физ. 1975. Т. 5, № 15. С. 1298–1309. [*Leont'ev V.K.* Ustoychivost zadachi kommivoyazhera // Zhurn. vychisl. matem. i matem. fiz. 1975. Vol. 5, № 15. S. 1298–1309 (in Russian)].

12. *Гордеев Э.Н., Леонтьев В.К.* Общий подход к исследованию устойчивости решений в задачах дискретной оптимизации // Журн. вычисл. матем. и матем. физ. 1996.

Т. 1, № 36. С. 66–72. [*Gordeev E.N., Leont'ev V.K.* Obshchy podkhod k issledovaniyu ustoychivosti resheny v zadachakh diskretnoy optimizatsii // Zhurn. vychisl. matem. i matem. fiz. 1996. Vol. 1, № 36. S. 66–72. [in Russian].)

13. *Orlin J.* Max flows in O(nm) time, or better // STOC. 2013. P. 765–774.

14. *King V., Rao S., Tarjan R.* A Faster Deterministic Maximum Flow Algorithm // Journal of Algorithms. Nov.1994. Vol. 17, № 3. P. 447—474.

On Stable Instances of MINCUT

Kozlov I. V.

Moscow Institute of Physics and Technology,
Institutskiy pereulok, 9, Dolgoprudny, Moscow region, 141700, Russia

Keywords: stability, mincut

A combinatorial optimization problem is called stable if its solution is preserved under perturbation of the input parameters that do not exceed a certain threshold – the stability radius. In [1–3] exact polynomial algorithms have been built for some NP-hard problems on cuts in the assumption of the entrance stability. In this paper we show how to accelerate some algorithms for sufficiently stable polynomial problems. The approach is illustrated by the well-known problem of the minimum cut (MINCUT). We built an $O(n^2)$ exact algorithm for solving n-stable instance of the MINCUT problem. Moreover, we present a polynomial algorithm for calculating the stability radius and a simple criterion for checking n-stability of the MINCUT problem.

Сведения об авторе:
Козлов Илья Владимирович,
Московский физико-технический институт,
кафедра математических основ управления
аспирант, ассистент