Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Потоки на графах

КУРСОВАЯ РАБОТА

студента 4 курса 431 группы
специальности 10.05.01 «Компьютерная безопасность»
факультета компьютерных наук и информационных технологий
Енца Михаила Владимировича

Научный руководитель
Ассистент ___________ Е.Н.Новокшонова
подпись, дата

Заведующий кафедрой
д.ф.-м.н., доцент ___________ М. Б. Абросимов
подпись, дата

Саратов 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В общем потоки на графах это очень полезная штука, расскажу вам об этом немного ниже. Вот.

1. Потоки на графах

Различные алгоритмы и их время работы.

1.1 Транспортные сети

Транспортная сеть $G = (V, E)$ представляет собой ориентированный граф, в котором каждое ребро $(u, v) \in E$ имеет неотрицательную пропускную способность (capacity) $c(u, v) > 0$. Если $(u, v) \notin E$, предполагается, что $c(u, v) = 0$. В транспортной сети выделяются две вершины: источник s и сток t. Для удобства предполагается, что каждая вершина лежит на неком пути из источника к стоку, т.е. для любой вершины $v \in V$ существует путь s → v → t. Таким образом, граф являелтся связным и $|E| > |V| - 1$. На рисунке __ показан пример транспортной сети __.

Пусть $G = (V, E)$ – транспортная сеть с функцией пропускной способности c. Пусть s – источник, а t – сток. Потоком (flow) в G является действительная функция $f: V \: x \: V \rightarrow R$, удовлетворяющая следующим трем условиям.

Ограничение пропускной способности (capacity constraint): $f(u, v) \leq c(u, v)$ для всех $u, v \in V$.

Антисиммутричность (skew symmetry): $f(u, v) = -f(u, v)$ для всех $u, v \in V$.

Сохранение потока (flow conservation): для всех $u \in V - \{s, t\}$

$$\sum_{v \in V} f(u, v) = 0$$

Количество $f(u, v)$, которое может быть положительным, нудевым или отрицательным, называется потоком (flow) из вершины u в вершину v. Величина (value) потока f определяется как

$$|f| = \sum_{v \in V} f(s, v)$$

Т.е. как суммарный поток, выходящий из источника. В задаче и максимальном потоке (maximum flow problem) дана некоторая транспортная суть G с источником s и стоком t, и необходимо найти поток максимальной величины.

Ограничение пропускной способности предполагает, чтобы поток из одной вершины в другую не превышал заданную пропускную способность ребра. Антисимметричность введена для удобства обозначения и заключается в том, что поток из вершины u в вершину v противоположен потоку в обратном направлении. Свойство сохранения потока утверждает, что суммарный поток, выходящий из вершины, не являющийся источником или стоком, равен нулю. Используя антисимметричность, можно записать свойство сохранения потока как

$$\sum_{u \in V} f(u, v) = 0$$

Для всех $v \in V - \{s, t\}$, т.е. суммарный поток, входящий в вершину, отличную от источника и стока равен 0.

Если в E не присутствуют ни (u, v), ни (v, u), между вершинами u и v нет потока, и $f(u, v) = f(v, u) = 0$.

Суммарный положительный поток (total positive flow), входящий в вершину v, задается выражением

$$\sum_{\substack{u \in V \\ f(u,v)>0}} f(u,v)$$

Суммарный положительный поток, выходящий из некоторой вершины, определяется симметрично. Суммарный чистый поток (total net flow) в некоторой вершине равен разности суммарного положительного потока, выходящего из данной вершины, и суммарного положительного потока, входящего в нее. Одна из интерпретаций свойства сохранения потока состоит в том, что для отличной от источника и стока вершины, входящей в нее суммарный положительный поток должен быть равен выходящему суммарному положительному потоку. Свойство, что суммарный чистый поток в транзитной вершине должен быть равен 0, часто нестрого формулируют как «входящий поток равен выходящему потоку».

1.1.1 Сети с несколькими источниками

?? надо ли

2. Алгоритм Диницы

Раздел с алгоритмами?? А конкретные сделать подпунктами??
Бла бла бла

3. Алгоритм проталкивания предпотока

В настоящее время многие наиболее асимптотически быстрые алгоритмы поиска максимального потока принадлежат данному классу, и на этом методе основаны реальные реализации алгоритма поиска максимального потока. С помощью методов проталкивания предпотока можно решать и другие связанные с потоками задачи, например, задачу поиска потока с минимальными затратами. Время работы простой реализации $O(V^2E)$, также существует усовершенствованная версия алгоритма, работающая за время $O(V^3)$.

Алгоритм проталкивания предпотока обрабатывают вершины по одной, рассматривая только соседей данной вершины в остаточной сети. Алгоритмы проталкивания предпотока не обеспечивают в ходе своего выполнения свойство сохранения потока. При этом они поддерживают предпоток (preflow), который представляет собой функцию $f: VxV \rightarrow R$, обладающую свойством антисимметричности, удовлетворяющую ограничениям пропускной способности и следующему ослабленному условию сохранения потока: $f(V,u) \geq 0$ для всех вершин $u \in V - \{s\}$. Это количество называется избыточным потоком (excess flow), входящим в вершину u, и обозначается

$$e(u) = f(V,u)$$

Вершина $u \in V - \{s,t\}$, называется переполненной (overflowing), если $e(u) > 0$.
Основные операции:
А алгоритме проталкивая предпотока выполняются две основные операции: проталкивание избытка потока от вершины к одной из соседних с ней вершин и подъем вершины.

Пусть $G = (V, E)$ транспортная сеть с источником s и стоком t, а f – некоторый предпоток в G. Функция $h(s) = |V|, h(t) = 0$ и

$$h(u) \leq h(v) + 1$$

Для любого остаточного ребра $(u, v) \in E_f$. Сразу же можно сформулировать следующую лемму.

Лемма 3.1. Пусть $G = (V, E)$ – транспортная сеть, а f – некоторый препоток в G, и пусть h – функция высоты, заданная на множестве V. Для любых двух вершин $u, v \in V$ справедливо следующее утверждение: $h(u) > h(v) + 1$, то (u,v) не является ребром остаточного графа.

ОПЕРАЦИЯ ПРОТАЛКИВАНИЯ:

Основная операция $PUSH(u, v)$ может применяться тогда, когда u является переполненной вершиной, $c_f(u, v) > 0$ и $h(u) = h(v) + 1$. Предполагается, что остаточные пропускные способности при заданных f и c можно вычислить за фиксированное время. Излишний поток, хранящийся в вершине u, поддерживается в виде атрибута $e[u]$, а высота вершины u – в виде атрибута $h[u]$. Выражение $d_f(u, v)$ – это временная переменная, в которой хранится количество потока, которое можно протолкнуть из u в v.

Процедура $PUSH$ работает следующим образом. Предполагается, что вершина u имеет положительный избыток $e[u]$ и остаточная пропускная способность ребра (u,v) положительна. Тогда можно увеличить поток из u в v на величину $d_f(u, v) = \min(e[u], c_f(u, v))$, при этом избыток $e[u]$ не становится отрицательным и не будет превышена пропускная способность c(u,v). Если функция f является предпотоком перед вызовом процедуры PUSH, она останется предпотоком и после ее выполения.

Процедура PUSH(u,v) называется проталкивание из u к v. Если операция проталкивания применяется к некоторому ребру (u,v), выходящему из вершины u, будем говорить, что операция проталкивания применяется к u. Если в результаттае ребро (u,v) становится насыщенным (после проталкивания $c_f(u, v) = 0$), то это насыщающее проталкивание, в противном случае это ненасыщающее проталкивание. Если ребро насыщено, оно не входит в остаточную сеть. Один из результатов ненасыщающего проталкивания характеризует следующая лемма.

Лемма 3.2. После ненасыщающего проталкивания из u в v вершина u более не является переполненной.

Нужно доказательство??

ОПЕРАЦИЯ ПОДЪЕМА 766 страница

Основная операция RELABEL(u) применяется, если вершина u переполнена и $h[u] \leq h[v]$ для всех ребер $(u, v) \in E_f$. Иными словами, переполненную вершину u можно подвергнуть подъему, если все вершины v, для которых имеется остаточная пропускная способность от u к v, расположены не ниже u, так что протолкнуть поток из u нельзя. Ни источник s, ни сток t нельзя подвергать подъему.

RELABEL(u):

Условие применения: u переполнена и для всех $v \in V$, таких что $(u, v) \in E_f, h[u] \leq h[v]$.

Pass

Когда вызывается операция RELABEL(u), мы говорим, что вершина u подвергается подъему (relabeled). Заметим, что когда производится подъем u, остаточная сеть E_f должна содержать хотя бы одно ребро, выходящее из u, чтобы минимизация в коде операции производилась по непустому множеству. Это свойство вытекает из предположения, что вершина u переполнена. Поскольку $e[u] > 0$, имеем $e[u] = f(V, u) > 0$ и, следовательно, должна существовать по крайней мере одна вершина v, такая что $f[v, u] > 0$. Но тогда

$$c_f(u, v) = c(u, v) - f[u, v] = c(u, v) + f[v, u] > 0,$$

Откуда вытекает, что $(u, v) \in E_f$. Таким образом, операция RELABEL(u) дает u наибольшую высоту, допускаемую наложенными на функцию высоты ограничениями.

Универсальный алгоритм.

Универсальный алгоритм проталкивания предпотока использует следующую процедуру для создания начального предпотока в транспорной сети:

INITIALIZE_PREFLOW(G,s)

for (для) каждой вершины $u \in V[G]$
    $do\ h[u] \leftarrow 0$
        $e[u] \leftarrow 0$
for (для) каждого ребра $(u, v) \in E|G|$
    $do\ f[u, v] \leftarrow 0$
        $f[v, u] \leftarrow 0$
$h[s] \leftarrow |V[G]|$
for (для) каждой вершины $u \in Adj[s]$
    $do\ f[s, u] \leftarrow c(s, u)$
        $f[u, s] \leftarrow -c(s, u)$
        $e[u] \leftarrow c(s, u)$
        $e[s] \leftarrow e[s] - c(s, u)$

Процедура INITIALIZE_PREFLOW(G,s) создает начальный предпоток f, определяемый формулой

$$f[u, v] = \begin{cases} c(u, v), \text{если } u = s \\ -c(v, u), \text{если } v = s \\ 0, \text{в противном случае.} \end{cases}$$

Это действительно функция высоты, поскольку единственным ребром (u, v), для которых $h[u] > h[v] + 1$, являются ребра, для которых u = s, и эти ребра заполнены, а это значит, что их нет в остаточной сети.

Инициализация, за которой следует ряд операций проталкивания и подъема, выполняемых без определенного порядка, образует алгоритм Generic_push_relabel:

Generic_push_relabel:

INITIALIZE_PREFLOW(G,s)
while существует применимая операция проталкивания или подъема

3   do выбрать операцию проталкивания или подъема и выполнить её
Следующая лемма утверждает, что до тех пор пока существует хотя бы одна переполненная вершина, применима хотя бы одна из этих операций.
Лемма (Для переполненной вершины можно выполнить либо проталкивание, либо подъем).

Пусть $G = (V, E)$ – транспортная сеть с источником s и стоком t, f – предпоток, а h – некоторая функция высоты для f. Если u – некоторая переполненная вершина, то к ней можно применить или операцию проталкивания, или операцию подъема.
** Доказательство???
Корректность метода проталкивания предпотока

Чтобы показать, что универсальный алгоритм проталкивания предпотока возволяет решить задачу максимального потока, сначала докажем, что после его завершения предпоток f является максимальным потоком. Затем докажем, что алгоритм завершается. Начнем с рассмотрения некоторых свойств функции высоты h.
Лемма (Высота вершины никогда не уменьшается)

При выполнении процедуры GENERIC_PUSH_RELABEL над ьранстпортной сетью $G = (V, E)$, для любой вершины $u \in V$ ее высота $h[u]$ никогда не уменьшается. Более того, всякий раз, когда к вершине u применяется операция подъема, ее высота $h[u]$ увеличивается как минимум на 1.
** Доказательство???
Лемма

Пусть $G = (V, E)$ – транспортная суть с источником s и стоком t. Во время выполнения процедуры GENERIC_PUSH_RELANEL над сетью G атрибут h сохраняет свойство функции высоты.
** Доказательство???
Следующая лемма характеризует выжное свойство функции высоты.
Лемма

Пусть $G = (V, E)$ – транспортная суть с источником s и стоком t, f – предпоток в G, а h – функция высоты, определенная на множестве V. Тогда не существует пути из источника s к стоку t в остаточной сети G_f.
** Доказательство?? 770

Теперь покажем, что после завершения универсального алгоритма проталкивания предпотока вычисленный алгоритмом предпоток является максимальным потоком.
Теорема (О корректности универсального алгоритма проталкивания предпотока).

Если алгоритм GENERIC_PUSH_RELABEL, выпоняемый над сетью $G = (V < E)$ с источником s и стоком t, завершается, то вычисленный им предпоток f является масимальным потоком в G.
** Доказательство??? 770

4. Практическая часть

Сравнение времени работы алгоритмов.

ЗАКЛЮЧЕНИЕ

Кек

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Лол

ПРИЛОЖЕНИЕ 1

да