Транспортная сеть и максимальный поток

Гусев Илья, Булгаков Илья

Московский физико-технический институт

Москва, 2019

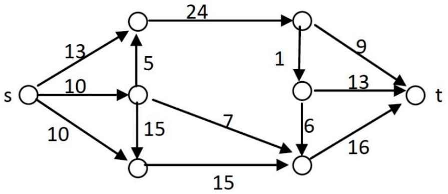
Содержание

Транспортная сеть

Определение. Транспортная сеть — ориентированный граф G=(V,E), в котором

- Каждое ребро $(u, v) \in E$ имеет неотрицательную пропускную способность $c(u, v) \geq 0$.
- Выделяются две вершины: источник s и сток t такие, что любая другая вершина сети лежит на пути из s в t.

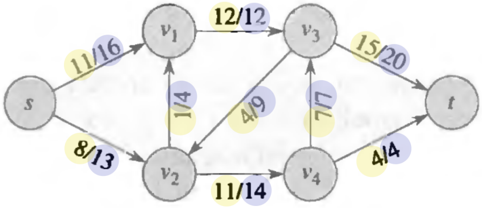
Зачем это нужно? Транспортная сеть может быть использована для моделирования, например, дорожного трафика.



Поток в транспортной сети

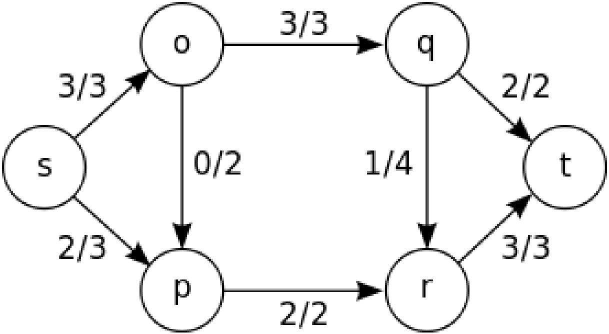
Определение. Потоком f в сети $G = (V, E, c)$ называется функция $f : E \to R$, удовлетворяющая условиям:

- $0 \leq f(e) \leq c(e)$ для всех $e \in E$;
- $f(v') = f(v*)$ для всех $v \in V$, $v \neq s$, $v \neq t$, где $f(v') = \sum_{w \in v'} f(w, v)$, $f(v*) = \sum_{u \in v*} f(v, u)$.



Задача о максимальном потоке

Задача о максимальном потоке заключается в нахождении такого потока по транспортной сети, что сумма потоков из истока, или, что то же самое, сумма потоков в сток максимальна.



Задача о максимальном потоке

Построение максимального потока

- Метод и алгоритм Форда-Фалкерсона
- Алгоритм Эдмондса-Карпа
- Алгоритм Диница

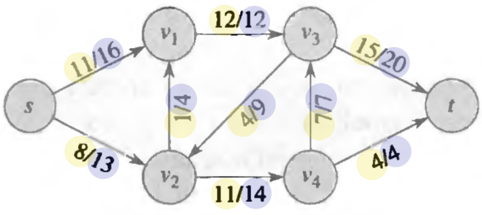
Метод Форда-Фалкерсона

Описание метода Форда Фалкерсона

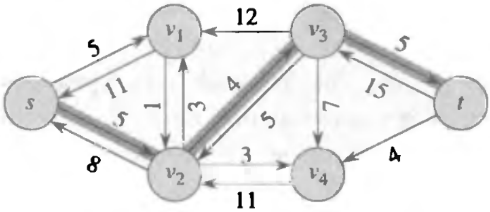
1. Задаём начальное значение потока $f = 0$
2. while (существует увеличивающий путь p)
   - Увеличиваем поток f вдоль пути p
3. Возвращаем f

Метод Форда-Фалкерсона

Сеть

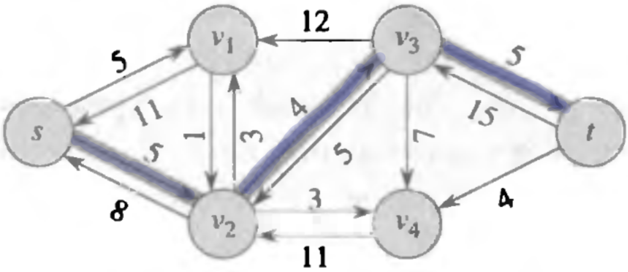


Остаточная сеть



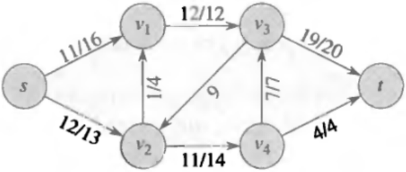
Метод Форда-Фалкерсона

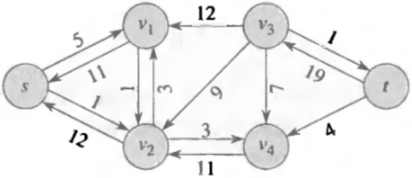
Увеличивающий путь



Метод Форда-Фалкерсона

Остаточная сеть и невозможность построить увеличивающий путь





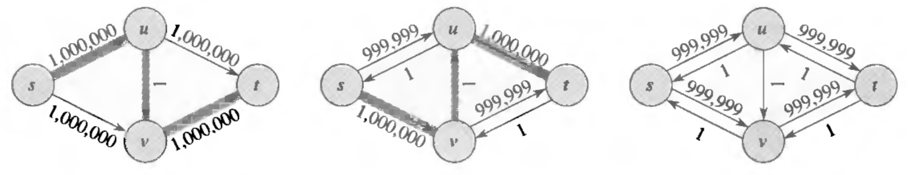
Метод Форда-Фалкерсона

Описание базового алгоритма Форда Фалкерсона

1. Для каждого ребра $(u, v) \in G$ задаём начальное значение потока $f = 0$
2. **while** (существует путь p в остаточной сети G_f)
    - Вычисляем $c_f(p) = min\{c_f(u, v) : (u, v)$ содержится в p $\}$
    - Для каждого ребра (u, v) в p
        - **if** $(u, v) \in E$ содержится в p $(u, v).f+ = c_f(p)$
        - **else** $(u, v).f- = c_f(p)$
3. Возвращаем f

Метод Форда-Фалкерсона

Сложность алгоритма – $O(E|f*|)$
Проблемы алгоритма при большой величине потока. Требуется 2млн итераций для нахождения максимального потока.



Алгоритм Эдмондса-Карпа

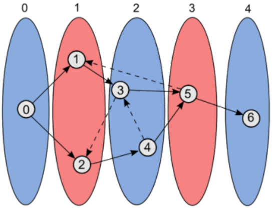
Модификация алгоритма Форда-Фалкерсона, в котором вычисление увеличивающего пути p производится как поиск в ширину и выбирается кратчайший путь из s в t. Вес ребра принимается за 1.
Сложность алгоритма – $O(V * E^2)$

Алгоритм Диница

Блокирующим потоком в данной сети называется такой поток, что любой путь из истока s в сток t содержит насыщенное этим потоком ребро. Иными словами, в данной сети не найдётся такого пути из истока в сток, вдоль которого можно беспрепятственно увеличить поток.
Блокирующий поток не обязательно максимален. Поток будет максимальным тогда и только тогда, когда в остаточной сети не найдётся $s - t$ пути; в блокирующем же потоке ничего не утверждается о существовании пути по рёбрам, появляющимся в остаточной сети.

Алгоритм Диница

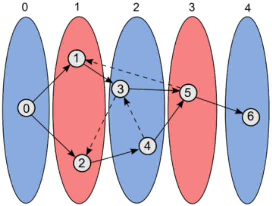
Слоистая сеть (layered network).



- Определяем длины кратчайших путей из истока s до всех остальных вершин. Назовём уровнем $\mathrm{level}[v]$ вершины её расстояние от истока.
- Включаем все те рёбра (u, v) исходной сети, которые ведут с одного уровня на какой-либо другой, более поздний, уровень, т.е. $\mathrm{level}[u] + 1 = \mathrm{level}[v]$
- Удаляем все рёбра, расположенные целиком внутри уровней, а также рёбра, ведущие назад, к предыдущим уровням.

Алгоритм Диница

Слоистая сеть (layered network).



Слоистая сеть ациклична. Кроме того, любой $s - t$ путь в слоистой сети является кратчайшим путём в исходной сети.
Как построить: для этого надо запустить обход в ширину по рёбрам этой сети, посчитав тем самым для каждой вершины величину $\mathrm{level}[]$, и затем внести в слоистую сеть все подходящие рёбра.

Алгоритм Диница

Алгоритм представляет собой несколько фаз.
Фаза алгоритма

- Строим остаточную сеть
- По отношению к ней строим слоистую сеть (обходом в ширину)
- Ищем произвольный блокирующий поток
- Найденный блокирующий поток прибавляется к текущему поток

Алгоритм Диница

Поиск блокирующего потока. Возможные реализации:

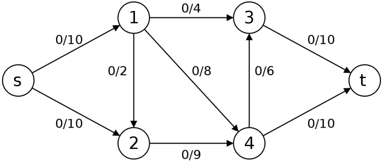
- Ищем пути по одному, пока такие пути находятся. Путь можно найти за O(m) обходом в глубину, а всего таких путей будет O(m) (каждый путь насыщает минимум одно ребро). Получаем $O(E^2)$.
- Аналогично предыдущей идее, однако удалять в процессе обхода в глубину из графа все "лишние"рёбра, т.е. рёбра, вдоль которых не получится дойти до стока. Поддерживать в списке смежности каждой вершины указатель на первое неудалённое ребро, и увеличивать этот указать в цикле внутри обхода в глубину. Худший случай $O(V * E)$

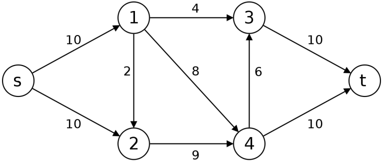
Алгоритм Диница

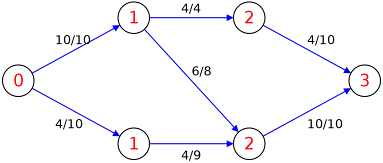
Отличие от алгоритма Эдмондса-Карпа: на каждой итерации поток увеличивается не вдоль одного кратчайшего $s - t$ пути, а вдоль целого набора таких путей (ведь именно такими путями и являются пути в блокирующем потоке слоистой сети).
Сложность алгоритма в худшем случае – $O(E * V^2)$

Алгоритм Диница

Пример. Фаза 1.

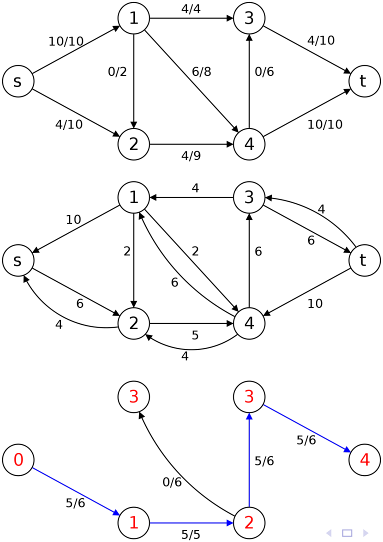






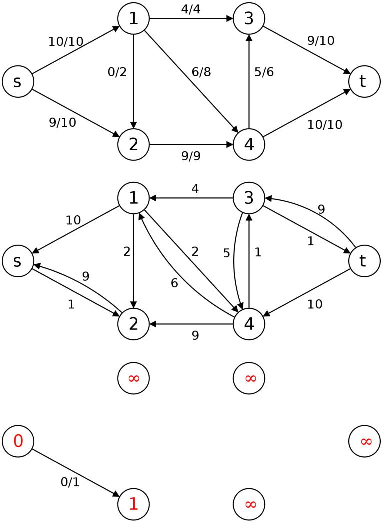
Алгоритм Диница

Пример. Фаза 2.



Алгоритм Диница

Пример. Фаза окончания.



Полезные ссылки I

- E-maxx: описание алгоритма Диница
  http://e-maxx.ru/algo/dinic
- Методы применения алгоритма нахождения максимального потока в сети
  https://habr.com/ru/post/102367/