Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС»

Комбинаторика и теория графов

Задача построения максимального потока в сети. Алгоритм Эдмондса-Карпа

Авад Фатхи Абделмонем Мохамед Ахмед

<https://github.com/FATHEY12352/alg_cm3>

**Содержание**

1. **Введение**
2. **Формальная постановка задачи**
3. **Теоретическое описание алгоритма**  
   3.1. Основы алгоритма Эдмондса-Карпа  
   3.2. Временная сложность
4. **Сравнительный анализ алгоритма**
5. **Перечень используемых инструментов**
6. **Описание реализации**  
   6.1. Основные компоненты реализации  
   6.2. Пример кода на C#
7. **Тестирование и результаты**  
   7.1. Пример графа и ход выполнения  
   7.2. Итоговые результаты
8. **Заключение**
9. **Список литературы**

**Введение**

В теории графов задача максимального потока является одной из центральных. Она имеет множество приложений, начиная от оптимизации транспортных сетей и заканчивая распределением ресурсов в компьютерных системах. Алгоритм Эдмондса-Карпа, модификация алгоритма Форда-Фалкерсона, позволяет эффективно решать эту задачу, используя стратегию поиска в ширину для нахождения увеличивающих путей.

**1. Формальная постановка задачи**

Рассматривается сеть G=(V,E)G = (V, E)G=(V,E), где:

* VVV — множество вершин графа,
* EEE — множество рёбер,
* c(u,v)c(u, v)c(u,v) — пропускная способность ребра (u,v)(u, v)(u,v).

**Задача**: Найти максимальный поток f(s,t)f(s, t)f(s,t) из истока sss в сток ttt, удовлетворяющий следующим условиям:

1. Поток через любое ребро f(u,v)f(u, v)f(u,v) не превышает его пропускную способность: 0≤f(u,v)≤c(u,v)0 \leq f(u, v) \leq c(u, v)0≤f(u,v)≤c(u,v), (u,v)∈E(u, v) \in E(u,v)∈E.
2. Закон сохранения потока: сумма входящих потоков в вершину равна сумме выходящих потоков, за исключением истока sss и стока ttt: ∑u∈Vf(u,v)=∑w∈Vf(v,w),  ∀v∈V∖{s,t}.\sum\_{u \in V} f(u, v) = \sum\_{w \in V} f(v, w), \; \forall v \in V \setminus \{s, t\}.u∈V∑​f(u,v)=w∈V∑​f(v,w),∀v∈V∖{s,t}.

**Цель**: Максимизировать поток ∑u∈Vf(s,u)\sum\_{u \in V} f(s, u)∑u∈V​f(s,u).

**2. Теоретическое описание алгоритма**

**Алгоритм Эдмондса-Карпа**

Алгоритм решает задачу максимального потока, используя поиск в ширину (BFS) для нахождения увеличивающего пути в остаточной сети.

**Этапы работы алгоритма:**

1. **Инициализация**:
   * Поток через все рёбра устанавливается равным нулю: f(u,v)=0, (u,v)∈E
   * Создаётся остаточная сеть G\_f​, где остаточная пропускная способность определяется как: c\_f(u, v) = c(u, v) - f(u, v).
2. **Поиск увеличивающего пути**:
   * Используется BFS для нахождения пути из s в **t** в **Gf​**.
   * Если путь найден, определяется минимальная остаточная пропускная способность вдоль пути (бутылочное горлышко): δ=min⁡(u,v)∈Pcf(u,v).
3. **Обновление потоков**:
   * Для каждого ребра в пути увеличивается поток: f(u,v)=f(u,v)+δ.
   * Для обратных рёбер уменьшается поток: f(v,u)=f(v,u)−δ
4. **Повтор**:
   * Алгоритм повторяет шаги 2–3, пока существует увеличивающий путь.

**Временная сложность:**

Алгоритм имеет временную сложность O(VE^2), где V — число вершин, E — число рёбер. Это объясняется тем, что каждый BFS выполняется за O(E), а количество итераций ограничено O(E).

**3. Сравнительный анализ алгоритма**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Алгоритм** | **Временная сложность** | **Подход** | | Форда-Фалкерсона | Экспоненциальная | DFS | | Эдмондса-Карпа | O(VE^2) | BFS | | Алгоритм Диница | O(V^2.E) | Уровневые графы | | Алгоритм Каргера | O(V^3) | Случайное сокращение | |

Эдмондс-Карп является более предсказуемым и легко реализуемым, чем классический Форда-Фалкерсона, хотя уступает в производительности алгоритму Диница.

**4. Перечень используемых инструментов**

* **Язык программирования**: C#.
* **Среда разработки**: Microsoft Visual Studio.
* **Дополнительные библиотеки**: System.Collections.Generic для работы с очередями.

**5. Описание реализации**

**Основные компоненты реализации:**

1. **Класс EdmondsKarp**:
   * Содержит матрицы пропускной способности и потока.
   * Реализует BFS для поиска пути.
   * Обновляет потоки.
2. **Методы**:
   * AddEdge(int from, int to, int capacity) — добавление ребра.
   * MaxFlow(int source, int sink) — вычисление максимального потока.

**Код на C#A screen shot of a computer screen

Description automatically generated**

A screenshot of a computer program

Description automatically generated

## 6. Тестирование и результаты

Для проверки работы алгоритма использовался следующий граф с шестью вершинами:

* Ребро из 0 в 1 с пропускной способностью 16.
* Ребро из 0 в 2 с пропускной способностью 13.
* Ребро из 1 в 2 с пропускной способностью 10.
* Ребро из 1 в 3 с пропускной способностью 12.
* Ребро из 2 в 4 с пропускной способностью 14.
* Ребро из 3 в 2 с пропускной способностью 9.
* Ребро из 3 в 5 с пропускной способностью 20.
* Ребро из 4 в 3 с пропускной способностью 7.
* Ребро из 4 в 5 с пропускной способностью 4.

**Результат работы программы**:  
Максимальный поток из вершины 0 в вершину 5 составляет **23**.

A screenshot of a computer

Description automatically generated

**Журнал выполнения алгоритма**

1. Первый увеличивающий путь: 0→1→3→5 бутылочным горлышком δ=12.
2. Второй увеличивающий путь: 0→2→4→ бутылочным горлышком δ=4.
3. Третий увеличивающий путь: 0→2→1→3→5 бутылочным горлышком δ=7.

**Сумма потоков**: 12+4+7=23.

## 7. Заключение

Алгоритм Эдмондса-Карпа продемонстрировал эффективность при решении задачи максимального потока в сети. Преимущества алгоритма:

1. Простота реализации благодаря использованию поиска в ширину.
2. Предсказуемая временная сложность O(VE^2), что позволяет решать задачи среднего масштаба.

Основной недостаток алгоритма — сравнительно высокая временная сложность по сравнению с алгоритмом Диница, особенно для графов с большим числом рёбер. Тем не менее, алгоритм остаётся популярным благодаря своей универсальности и ясности.

## 8. Список литературы

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. "Алгоритмы: Построение и анализ".
2. Edmonds J., Karp R.M. "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems", 1972.
3. Официальная документация C#: [docs.microsoft.com](https://docs.microsoft.com).