Алгоритм Форда-Фалкерсона нахождения максимального потока в транспортной сети

Выполнил: Рычков А.Е.
Студент 332 группы

Введение

Ориентированный граф можно интерпретировать как некоторую транспортную сеть и использовать для решения задач о потоках вещества в системе трубопроводов. Каждое ориентированное ребро сети можно рассматривать как канал, по которому движется продукт. Каждый канал имеет заданную пропускную способность, которая характеризует максимальную скорость перемещения продукта по каналу. Вершины являются точками пересечения каналов, через вершины, отличные от истока и стока, продукт проходит, не накапливаясь. Иными словами, скорость поступления продукта в вершину должна быть равна скорости его удаления оттуда.

Основные понятия

Транспортная сеть (flow network) $G = (V, E)$ представляет собой ориентированный граф, в котором каждое ребро $(u, v) \in E$ имеет неотрицательную пропускную способность (capacity) $c(u, v) \geq 0$.В случае, если E содержит ребро (u, v), обратного ребра (v, u) быть не должно. Если $(u, v) \notin E$, то $c(u, v) = 0$, петли также запрещены. Также выделяются две вершины: исток (source) s и сток (sink) t. Каждая вершнина лежит на некотором пути от истока к стоку, то есть для любой вершины $v \in V$ транспортная сеть содержит путь $s \to v \to t$. Таким образом, граф является связным, следовательно каждая вершина, отличная от s, содержит как минимум одно входящее ребро ($|E| \geq |V| - 1$).

Потоком(flow) является действительная функция $f : V \to \mathbb{R}$, удовлетворяющая следующим условиям:
Ограничение пропускной способности. Для всех $u, v \in V$ должно выполняться $0 \leq f(u, v) \leq c(u, v)$.
Сохранение потока. Для всех $u \in V - \{s, t\}$ должно выполняться

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

Когда $(u, v) \notin E$, потока из u в v быть не может, так что $f(u, v) = 0$.

Величина потока $|f|$ потока f определяется как

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

Остаточную пропускную способность $c_f(u,v)$ введем как

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v), & \text{если } (u,v) \in E, \\ f(v,u), & \text{если } (v,u) \in E, \\ 0 & \text{в противном случае.} \end{cases}$$

Остаточная сеть для $G = (V, E)$ и потока f представляет собой граф $G_f = (V, E_f)$, где

$$E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$$

Ребра в E_f являются либо ребрами из E, либо обратные им, таким образом, $|E_f| \leq 2|E|$. Остаточная сеть удовлетворяет всем свойствам транспортной сети, кроме того, что она может содержать одновременно и ребро (u,v), и (v,u).

Увеличение потока f в транспортной сети G на f' из остаточной сети G_f определяется как функция, отображающая $V \times V$на $\mathbb{R}$:

$$(f \uparrow f)(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u), & \text{если } (u,v) \in E, \\ 0 & \text{в противном случае} \end{cases}$$

Лемма об увеличении потока

(Здесь и далее леммы и теоремы приводятся без доказательств)

Пусть $G = (V, E)$ является транспортной сетью с истоком s и стоком t и пусть f представляет собой поток в G. Пусть G_f - остаточная сеть G, порожденная f, и пусть f' - поток в G_f. Тогда функция $f \uparrow f'$ представляет собой поток в G с величиной $|f \uparrow f'| = |f| + |f'|$.

Увеличивающим путем p для транспортной сети $G = (V, E)$ и потока f является простой путь из s в t в остаточной сети G_f.

Остаточная пропускная способность - это максимальная величина, на которую можно увеличить поток в каждом ребре увеличивающего пути p, и которая задается формулой

$$c_f(p) = min\{c_f(u,v) : (u,v) \in p\}$$

Лемма об остаточной пропускной способности

Пусть $G = (V, E)$ является транспортной сетью, а f представляет собой поток в G, и пусть p является увеличивающим путем в G_f. Определим функцию $f_p : V \times V \to \mathbb{R}$ как

$$f_p(u,v) = \begin{cases} c_f(p), & \text{если } (u,v) \in p, \\ 0 & \text{в противном случае} \end{cases}$$

Тогда f_p является потоком в G_f с величиной $|f_p| = c_f(p) > 0$.

Следствие

Пусть $G = (V, E)$ представляет собой транспортную сеть, а f является потоком в G , и пусть p представляет собой увеличивающий путь в G_f. Пусть также f_p определен, как в предыдущем уравнении, и предположим, что мы увеличиваем f на f_p. Тогда функция $f \uparrow f_p$ является потоком в G с величиной $|f \uparrow f_p| = |f| + |f_p| > |f|$

Разрезом (S, T) транспортной сети $G = (V, E)$ называется разбиение множества вершин V на множества S и $T = V - S$, такие, что $s \in S$, а $t \in T$.

Чистый поток $f(S, T)$ через разрез (S, T) определяется как

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

Пропускной способностью разреза (S, T) является

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

Минимальным разрезом сети является разрез, пропускная способность которого среди всех разрезов сети минимальна.

Лемма о чистом потоке

Пусть f представляет собой поток в транспортной сети G со стоком s и истоком t и пусть (S, T) - произвольный разрез G. Тогда чистый поток через разрез (S, T) равен $f(S, T) = |f|$.

Следствие

Величина любого потока f в транспортной сети G ограничена сверху пропускной способностью произвольного разреза G.

Теорема о максимальном потоке и минимальном разрезе

Если f представляет собой поток в транспортной сети $G = (V, E)$ с истоком s и стоком t, то следующие утверждения эквивалентны.

1. f является максимальным потоком в G.
2. Остаточная сеть G_f не содержит увеличивающих путей.
3. $|f| = c(S, T)$ для некоторого разреза (S, T) транспортной сети G.

Формулировка задачи о максимальном потоке

Дана некоторая транспортная сеть $G = (V, E)$ с истоком s и стоком t, и необходимо найти поток максимальной величины.

Алгоритм Форда-Фалкерсона

1. **Для** каждого ребра $(u, v) \in G.E$
2. $\quad (u, v).f = 0$
3. **Пока** существует путь p из s в t в остаточной сети G_f
4. $\quad c_f(p) = min\{c_f(u, v) : (u, v) \in p\}$
5. $\quad$ **Для** каждого ребра (u, v) в p
6. $\quad\quad$ **Если** $(u, v) \in E$
7. $\quad\quad\quad (u, v).f = (u, v).f + c_f(p)$
8. $\quad\quad$ **Иначе** $(v, u).f = (v, u).f - c_f(p)$

Примечания к реализации алгоритма

Алгоритм был реализован мною на языке C. Функция, отвечающая за алгоритм – int findMaxFlow(int n, int c[n][n]). В качестве параметров принимает кол-во вершин и матрицу смежности с значениями пропускной способности для каждого ребра, а возвращает величину максимального потока в сети.

В качестве основной структуры данных был использован двумерный статический массив. Поиск увеличивающего пути в остаточной сети ведется с помощью алгоритма поиска в ширину.

Анализ алгоритма

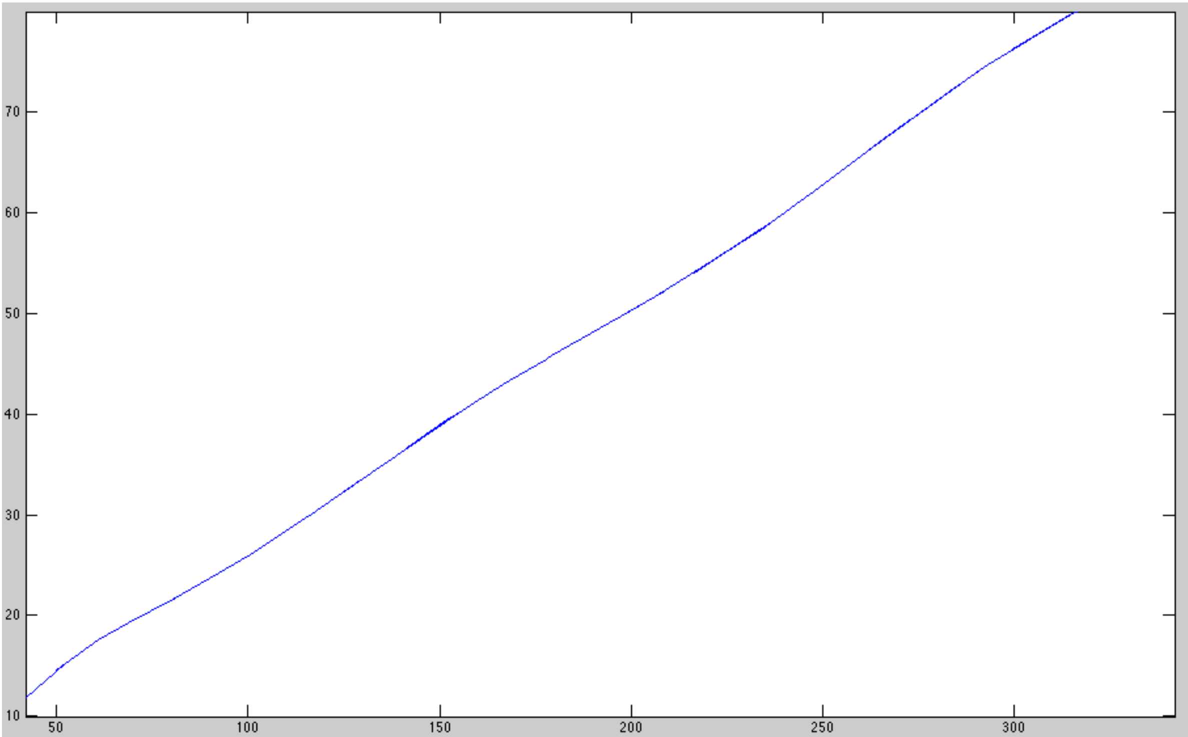
Время выполнения алгоритма зависит от того, как именно выполняется поиск увеличивающего пути в строке 3. Если увеличивающий путь выбирается с использованием алгоритма поиска в ширину, то алгоритм выполняется за полиномиальное время. Если значения пропускной способности ребер являются иррациональными числами, то величина потока не будет сходиться к максимальной величине потока. Если пропускные способности являются рациональными числами, то можно использовать соответствующее масштабирование, чтобы сделать их целыми.

Если обозначить максимальный поток в сети как f^*, то цикл while выполняется не более $|f^*|$ раз, поскольку величина потока за каждую итерацию увеличивается по крайней мере на единицу.

Цикл будет выполняться наиболее эффективно, если реализовать транспортную сеть $G = (V, E)$ с помощью правильно выбранной структуры данных и искать увеличивающий путь с помощью алгоритма с линейным временем работы. Например, если мы поддерживаем структуру данных, соответствующую ориентированному графу $G' = (V, E')$, где $E' = \{(u, v) \in E$ или $(v, u) \in E$. Ребра сети G являются также ребрами графа G', поэтому в такой структуре данных можно легко хранить пропусные способности и потоки. Для данного потока f в G ребра остаточной сети G_f состоят из всех ребер (u, v) графа G', таких, что $c_f(u, v) > 0$. Следовательно, время поиска пути в остаточной сети составляет $\mathrm{O}(V + E') = \mathrm{O}(E)$, если используется поиск в глубину или ширину. Таким образом, итерация цикла while занимает время $\mathrm{O}(E)$, что вместе с инициализацией делает общее время выполнения равным $\mathrm{O}(E|f^*|)$.

Сравнение практической и теоретической части

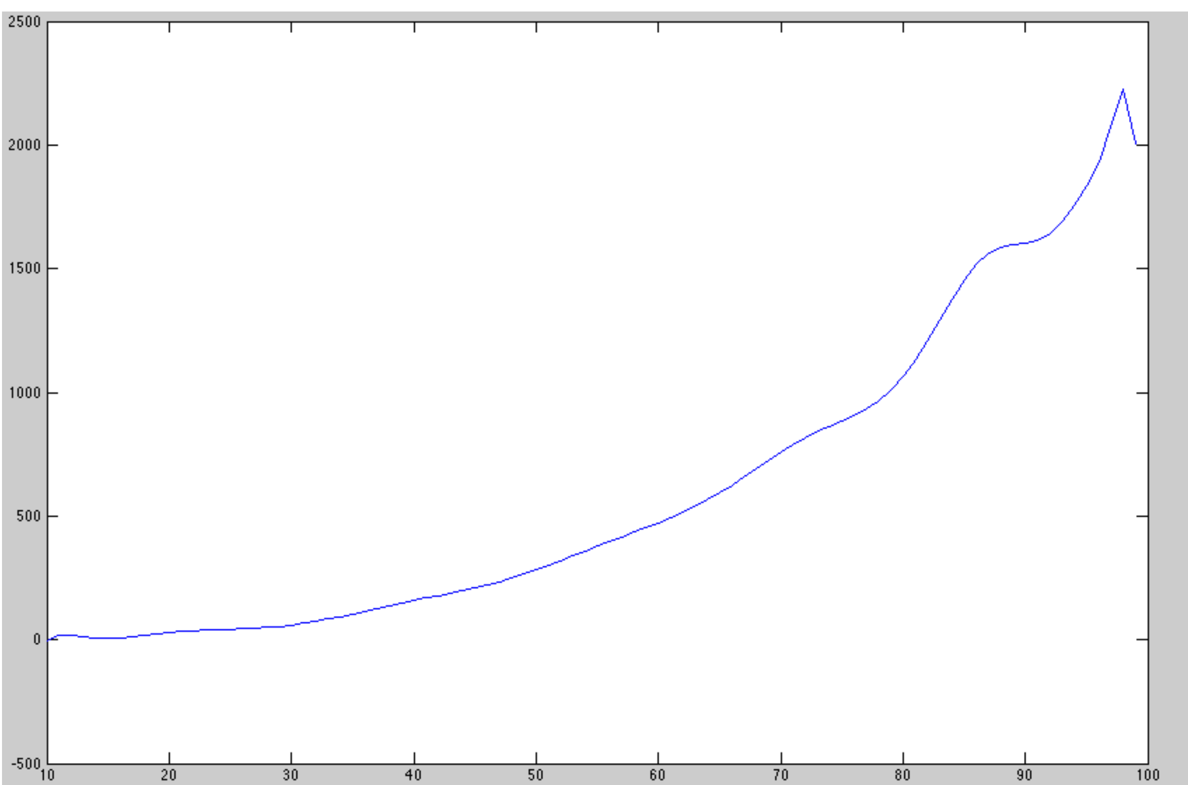
Проверим, действительно ли цикл while будет выполняться не более $|f^*|$ раз. Проведем тест при $|V| = 10 : 100$, при этом пропускная способность каждого ребра (кроме, естественно, петель и обратных ребер) задается случайно в диапазоне $0 : 10$. Для каждого $|V|$ повторим 5 раз, затем аппроксимируем результаты, для получения наиболее приближенной к действительности картины зависимости кол-ва итераций от величины максимального потока. На графике ось абсцисс взята за $|f^*|$, а ось ординат за количество итераций.



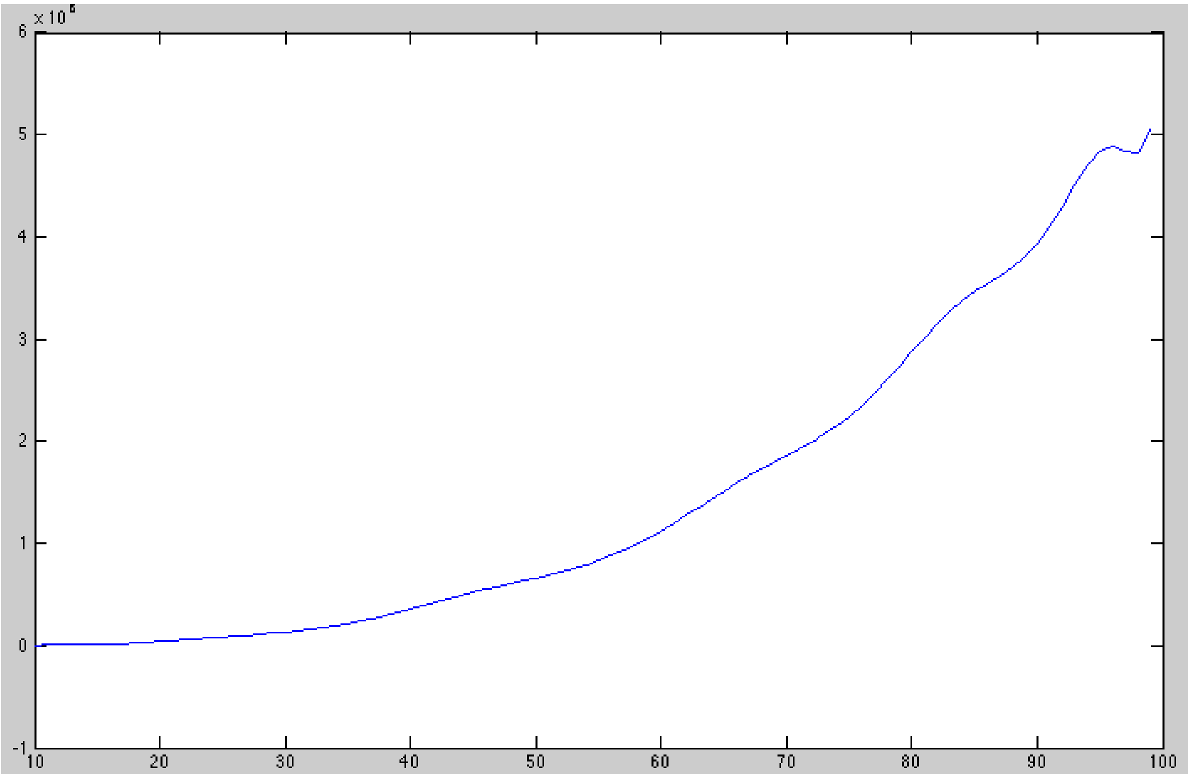
Как видно из графика,теория успешно совпадает с практикой.

Следует заметить, что при увеличении пропускной способности такая зависимость по-прежнему будет соблюдаться, при этом увеличится величина добавляемого потока.

Далее проиллюстрируем влияние рамера задачи на время выполнения алгоритма. Проведем указанные выше операции, но при этом засечем время нахождения (в миллисекундах) каждого максимального потока.



А также покажем зависимость кол-ва элементарных операций от размера входных данных.



Алгоритм выполняет намного больше элементарных операций, чем указано в его теоретической сложности. Но это обусловлено конкретной реализаций, если считать только те элементарные операции, которые обговорены в псевдокоде, то теоретическая сложность действительно будет выполняться.

Вывод

Классический алгоритм поиска максимального потока. Но далеко не самый лучший (если быть более точным, второй с конца по сложности). Хорош для обучения, но в целях практического применения стоит воспользоваться его модификациями или более быстрыми алгоритмами.

Литература

1. Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн "Алгоритмы. Построение и анализ."
2. https://ru.wikipedia.org/
3. http://algolist.manual.ru/maths/graphs/maxflows/
4. http://informatics.mccme.ru/mod/book/view.php?id=448