Максимальный поток

Артем Васильев

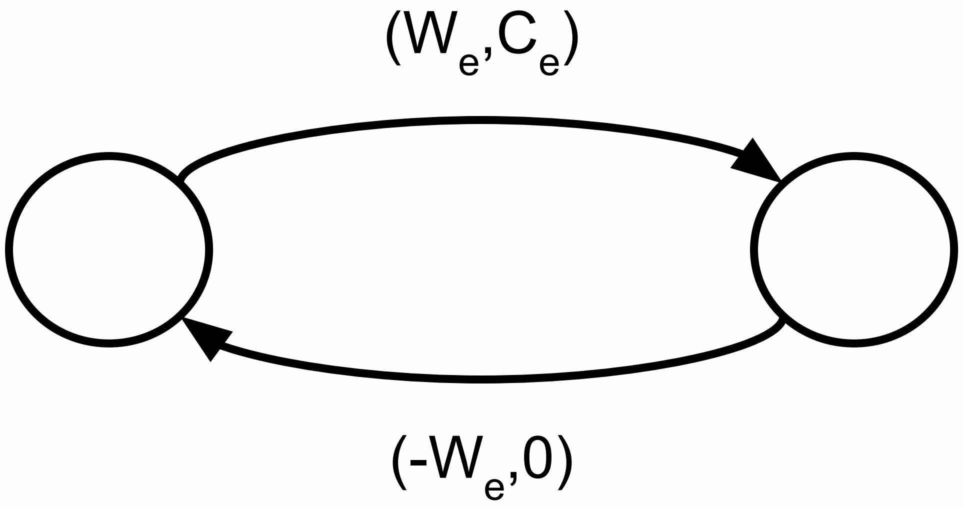
Алгоритмы и структуры данных Advanced

Отличие от обычного потока

- Взвешенная сеть G(V, E)
- Каждое ребро имеет *вес* w(e)
- Каждая единица потока по ребру стоит w(e)
- Вес может быть отрицательным
- Стоимость потока по ребру e: w(e) * f(e)

Обратные ребра

- Остаточная сеть строится аналогично: $c_f(e) = c(e) - f(e)$
- Какой вес у обратного ребра?
- $-c(e)$



Задачи

- Максимальный поток минимальной стоимости
- Поток минимальной стоимости заданной величины
- Поток минимальной стоимости (независимо от величины)
- Задача о назначениях

Теорема про оптимальность

- Зафиксируем стоимость потока
- Пусть f_1 не минимальный, f_2 - минимальный
- Рассмотрим “разность потоков”: $f_1 - f_2$
- Это поток величины 0 - “циркуляция”
- Циркуляция состоит из циклов
- Хотя бы один имеет стоимость < 0
- Можно добавить к f_1 и получить поток меньшей стоимости

Алгоритм отмены отрицательных циклов

- Поток минимальный тогда и только тогда, когда в остаточной сети нет отрицательных циклов
- Алгоритм: найдем любой поток (максимальный или фиксированной стоимости)
- Затем, пока существует отрицательный цикл в остаточной сети, найти и добавить его к потоку

Поиск отрицательного цикла

- Как найти отрицательный цикл в графе?
- Алгоритм Флойда: $O(V^3)$
- Найдем dist[u][v] для всех пар (u, v)
- Проверим, есть ли такая вершина, что dist[u][u] < 0
- Восстановим отрицательный цикл u -> … -> u

Поиск отрицательного цикла

- Алгоритм Форда-Беллмана: O(VE)
- Обычно запускаем V - 1 итерацию обновления ДП
- Запустим еще один раз
- Если что-то обновилось - существует отрицательный цикл
- Запустим восстановление пути из обновленной вершины

Другой подход

- Имея f_k - поток минимальной стоимости величины k, попробуем построить f_{k+1}
- $f_{k+1} - f_k$ это поток величины 1
- Найдем кратчайший путь P в остаточной сети G_f
- $f_{k+1} = f_k + P$
- Повторить пока не перестанет существовать путь
- Или не наберем поток величины k

Другой подход

- Время работы O(VE * |f|)
- Хочется использовать быстрый алгоритм поиска пути: алгоритм Дейкстры
- Но, в остаточной сети есть обратные отрицательные ребра!

Потенциалы Джонсона

- φ(u) - произвольное число
- Новый вес ребра w’(u, v) = w(u, v) + φ(u) - φ(v)
- Пусть могут быть отрицательные веса, но не отрицательных циклов
- Тогда существует φ: w’(u, v) >= 0 для всех ребер
- φ(u) = dist(s, u)

Улучшенный алгоритм

- Пусть в исходной сети нет отрицательных циклов
- Найдем φ(u) = dist(s, u) за O(VE) с помощью алгоритма Ф-Б
- Каждый следующий путь можно найти Дейкстрой за $O(V^2)$ или O(E log V) (или даже за O(V log V + E))
- Суммарное время O(VE + E log V * |f|)

And beyond!

- Все сегодняшние алгоритмы - неполиномиальные (зависят от |f|)
- Существуют строго пониномиальные
- Если находить не любой цикл, а цикл *минимального среднего веса*, то будет полином