Второй курс, осенний семестр 2017/18
Практика по алгоритмам #4

Потоки побыстрее, mincost

29 сентября

Собрано 21 сентября 2018 г. в 23:11

Содержание

Потоки побыстрее, mincost

1. **Многосочетание**
   Дан двудольный граф. Каждой вершине сопоставлено число a_i.
   Выбрать max число рёбер так, чтобы степени вершин были $\leqslant a_i$.

   За сколько будет работать алгоритм Диница в данном случае?

2. **Глобальный разрез**
   В неорграфе без кратных рёбер удалить min число рёбер так, чтобы увеличилось число компонент связности. $\mathcal{O}(V \cdot Flow)$. За сколько работает Форд-Фалкерсон?

3. **Давайте придумаем LR-поток**
   a) Несколько истоков, стоков. Пустить max поток.
   b) А если мы хотим k путей, причём из s_i именно в t_i?
   c) Даны несколько заводов (производит a_i товара) и магазинов (нуждается в b_j товара) и дорожная сеть. Придумать план перевозок, который удовлетворит все магазины.
   d) Найти любую LR-циркуляцию ($l_e \leqslant f_e \leqslant c_e$).
   e) Найти любой LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).
   f) Найти максимальный LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).

4. **Нарушение связности**
   Дан орграф. У каждого ребра есть неотрицательная стоимость удаления.
   Удалить рёбра минимальной суммарной стоимости так, чтобы из s не было пути в t.

5. **Оптимизированный поток**
   Научитесь поток размера $|f|$ искать за $\mathcal{O}(|f| \cdot V^2/w)$.

6. **LR через mincost**
   Найти LR-поток, используя mincost flow, за время $\mathcal{O}(\texttt{mincost flow})$.

7. **Mincost + Диниц**
   Рассмотрим такой алгоритм для поиска mincost потока: пока $\exists$ дополняющий путь в G_f, выделяем сеть кратчайших путей в G_f, ищем максимальный поток в этой сети.
   a) Докажите корректность алгоритма.
   b) За сколько работает такой алгоритм?
   c) К каким графам его разумно применять?

8. **Задача про автоматы**
   Есть k автоматов и n заданий. Про каждое задание известно, во сколько его нужно начать делать, во сколько закончить, а также его стоимость. Каждый автомат может выполнять только одно задание в каждый момент времени. Нужно выполнить задания максимальной суммарной стоимости.

   a) Решение за $\mathcal{O}(kn^2)$.
   b) Решение за $\mathcal{O}(kn \log n)$.

9. **(*) Малхотра-Кумар-Махешвари**
   Идея для оптимизации Диница: давайте научимся за $\approx \mathcal{O}(V)$ насыщать самую "узкую" вершину. Придумайте поток за $\mathcal{O}(V^3)$.

10. **(*) Непрерывная цена, мультипаросочетание**
Дан двудольный граф. Нужно найти мультипаросочетание – выбрать подмножество рёбер, у каждой вершины ограничена степень. Запишем задачу формально.
f_{ij} – поток между i-й вершиной первой доли и j-й вершиной второй доли.
$0 \leqslant f_{ij} \leqslant 1$, $s_i = \sum_j f_{ij} \leqslant a_i$, $t_i = \sum_j f_{ij} \leqslant b_j$.
Стоимость мультипаросочетания в этой задаче равна $\sum_i cost_i s_i^2$ (в обычном mincost потоке $\sum_i cost_i s_i$). Минимизировать стоимость максимального по $\sum_i s_i$ (размеру) паросочетания.

Разбор задач практики

1. **Многосочетание**
   Добавляем рёбра из истока в левую долю, из правой доли в сток. На добавленных рёбрах пропускные способности a_i, на рёбрах исходного графа 1. Ищем максимальный поток, ответ – насыщенные ребра исходного графа.

   Диниц за $\mathcal{O}(E\sqrt{E})$ по первой теореме Карзанова.

2. **Глобальный разрез**
   Вершина 1 лежит в какой-то половине разреза, нужно найти вершину, лежащую в другой. Фиксируем любой исток и перебираем сток. Пустили поток, нашли разрез. Время $\mathcal{O}(V \cdot Flow)$. Размер i-го потока не более $\deg_i \Rightarrow$ суммарный размер всех потоков не более $E \Rightarrow$ даже обычный Форд-Фалкерсон даст время $\mathcal{O}(E^2)$.

3. **Давайте придумаем LR-поток**
   a) **Несколько истоков, стоков. Пустить max поток.**
   Добавим общий исток и общий сток. Пропускные способности = $+\infty$.
   b) k **путей, из** s_i **именно в** t_i
   Это NP-трудно.
   c) **Транспортная задача (избытки и недостатки).**
   Добавим общий исток и общий сток. Пропускные способности = a_i, b_j.
   d) **Найти любую LR-циркуляцию ($l_e \leqslant f_e \leqslant c_e$).**
   1. $\forall$ ребра e пустим l_e единиц потока.
   Для ребра $e\colon a \to b$ образуется недостаток l_e в a и избыток l_e в b.
   2. В каждой вершине сложили избытки и недостатки.
   Получили предыдущую задачу на графе с пропускными способностями $c_e - l_e$.
   e) **Найти любой LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).**
   Добавим ещё и ребро $t \to s$ пропускной способности $+\infty$.
   f) **Найти максимальный LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).**
   Максимальный LR-поток – f^*, мы уже нашли какой-то LR-поток f.
   Заметим, что $f^* - f$ – максимальный (уже не LR, а обычный) поток в G_f.

4. **Нарушение связности**
   Нужно просто найти разрез min стоимости.

5. **Оптимизированный поток**
   Мы посетим всего V вершин. Задача: научиться ходить в непосещённую вершину за $\mathcal{O}(V/w)$. `unused` – `bitset` непосещённых вершин. Стоим в вершине `v`, переходим в младший бит `g[v] & unused`.

6. **LR через mincost**
   Вместо LR-ребра делаем L-ребро стоимости $-\infty$ и $(R-L)$-ребро стоимости 0. Если в полученном графе есть отрицательные циклы – ничего страшного, задачу mincost circulation мы тоже умеем решать.

7. **Mincost + Диниц**

1. В отличии от Диница сеть кратчайших путей – не слоистый граф, а произвольный. Даже может содержать циклы веса ноль.
2. Любой путь из s в t в сети кратчайших путей – кратчайший из s в t.
3. Мы уже знаем, что расстояние от s до t неуменьшается.

4. Пусть перед очередной фазой наш поток – f, G_f – остаточная сеть, $H(G_f)$ – оставили только рёбра на кратчайших путях. d – длина кратчайшего пути от s до t в G_f.
Δf – max поток в $H(G_f)$. Тогда, если в $G_{f+\Delta f}$ есть путь p длины d, то рассмотрим поток $(\Delta f)+p$ в G_f, декомпозируем его на пути. Все получившиеся пути имеют длину ровно $d \Rightarrow$ $(\Delta f)+p$ – поток в $H(G_d) \Rightarrow \Delta f$ не был максимальным.

5. Время работы $\mathcal{O}(D(VE + \texttt{maxFlow}))$, где D – число различных кратчайших расстояний.
6. Если все цены рёбер в графе равны, различных расстояний не более V, получается $\mathcal{O}(V(VE + \texttt{maxFlow}))$. Если все цены равный $\mathcal{O}(1)$, выйдет такая же асимптотика.

8. **Задача про автоматы**

Можно решать так: проведём ребро между заданиями, если одно можно начать после конца другого. Получится граф из $\Theta(n^2)$ ребер.
Раздваиваем вершины, на ребре между копиями v ставим цену $-c_v$.
У исходных ребер цена 0, у всех ребер пропускная способность 1.
Ищем mincost поток размера k с помощью нахождения кратчайшего пути в остаточной сети k раз. На первой итерации ищем кратчайший путь за $\mathcal{O}(E)$ dfs-ом, так как граф ацикличен. Дальше Дейкстрой, каждый путь за $\mathcal{O}(V^2)$. Итого $\mathcal{O}(E + kV^2)$.
Быстрое решение. Вершины графа – моменты времени: начала и концы заданий. Ребро цены $-cost_i$ и пропускной способности 1 между началом и концом i-го задания. Также ребро из каждого момента времени в следующий момент, цена 0, пропускная способность $+\infty$. То есть, свободный автомат либо берет задание и освобождается к концу задания, либо переходит в следующий момент времени, оставаясь свободным. Получилось $2n$ вершин и $3n$ ребер. Mincost поток размера k ищем за $\mathcal{O}(E + k \cdot Dijkstra) = \mathcal{O}(kE \log V) = \mathcal{O}(kn \log n)$.

Для графа с циклами не работает. Для начала заметим, что задача для $k = 1$ – уже поиск гамильтонова пути. Что же сломается? Только то, что поток размера 1 – это уже не путь, а "путь + нетривиальная циркуляция отрицательного веса". Поэтому ломается сопоставление mincost потоку величины k каких-то k непересекающихся путей *такого же* суммарного веса.

9. **(*) Малхотра-Кумар-Махешвари**

Оставим в графе только вершины $v\colon \exists s \rightsquigarrow v \rightsquigarrow t$. На оставшемся графе определим: $c[v] = \min(in[v], out[v])$, $in[v] = \sum_{e \in in[v]} c_e, out[v] = \sum_{e \in out[v]} c_e$.

Выберем за $\mathcal{O}(V)$ среди $v\colon c[v] > 0$ вершину $x\colon c[x] = \min$. Заметим, что из v в t поток можно толкать жадно – ему всегда есть куда утечь. Аналогично из s в v (толкаем из v по обратным рёбрам). Время на проталкивания $c[v]$ единиц потока из v в s и t равно $V + k_i$, где k_i – количество рёбер, по которым произошло насыщающее проталкивание. Если ребро насытилось, его сразу можно удалить из графа $\Rightarrow \sum k_i \leqslant E \Rightarrow$ суммарное время работы алгоритма $\mathcal{O}(V^2 + E)$.

10. **(*) Непрерывная цена, мультипаросочетание**
Когда мы уже пустили f потока в i-ю вершину левой доли, мы платим $f^2 cost_i$. Пустим ещё $+\varepsilon$ потока, получим стоимость $(f+\varepsilon)^2 cost_i$, прирост равен $(2f\varepsilon + \varepsilon^2) cost_i = 2f\varepsilon cost_i + o(\ldots)$. Получаем, что удельная стоимость единицы потока равна $2f \cdot cost_i$.
Можно представить, что у нас не одно ребро, а бесконечно много узких рёбер. j-е ребра ребра в i-ю вершину левой доли стоит $2(\varepsilon \cdot j) \cdot cost_i$. В одной пачке рёбра упорядочены по возрастанию веса ($cost_i > 0$). Стоимость кратчайшего пути – стоимость его первого ребра. Алгоритм поиска mincost потока говорит, что корректно $+\varepsilon$ потока толкнуть по пути, который начинается с ребра минимального веса.
Решение: n раз бинпоиском находим такое максимальное x, что можно пустить по всем рёбрам поток $\frac{x}{2cost_i}$, таким образом, заплатив за каждое $\leqslant x$, $cost_i$ пересчитано в соответствии с уже пущенным потоком. Проверка внутри бинпоиска через max flow. После конца бинпоиска по одному из n рёбер в левую долю нельзя пустить даже $x + \varepsilon$ потока, его можно просто удалить. Всего из истока исходит n рёбер, в худшем случае они будут умирать по одному.
`Time` $= \mathcal{O}(n \log MAX \cdot MaxFlow)$.