Курс kiev-clrs – Лекция 18. Кратчайшие пути: алгоритм Беллмана-Форда, линейное программирование

Олег Смирнов

18 июля 2009 г.

Содержание

1 Цель лекции

- Задача кратчайшего пути в графе с отрицательными дугами
- Линейное программирование

2 Алгоритм Беллмана-Форда

Алгоритм решает ту же задачу, что алгоритм Дейкстры, но работает в графах с отрицательными дугами и позволяет обнаруживать отрицательные циклы. Алгоритм проще в реализации, но хуже в производительности.

Bellman_Ford(G, w, s)

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$
    do $d[v] \leftarrow \infty$
for $i \leftarrow 1$ **to** $|V| - 1$
    do for each $(u, v) \in E$
        do if $d[v] > d[u] + w(u, v)$ //релаксация дуги
            then $d[v] \leftarrow d[u] + w(u, v)$
for each $(u, v) \in E$ //проверка наличия отрицательных циклов
    do if $d[v] > d[u] + w(u, v)$ //релаксация возможна?
        then return False //есть отрицательный цикл
return d

Алгоритм вычисляет путь с кратчайшим весом $\delta(s, v)$ из истока $s \in V$ во все вершины $v \in V$. Время выполнения $T = O(V \cdot E)$ как минимум квадратично от количества вершин, если граф связный.

Для иллюстрации работы на примере нужно задать некоторый (произвольный) порядок для дуг графа:

Первая итерация цикла даст следующий результат:

	A	**B**	**C**	**D**	**E**
	0	∞	∞	∞	∞
	0	-1	∞	∞	∞
	0	-1	4	∞	∞
	0	-1	2	∞	∞

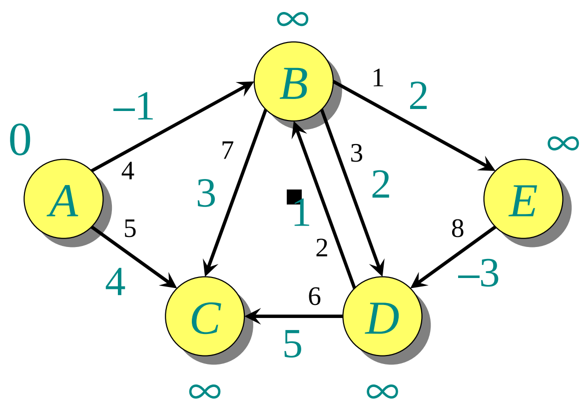

Рис. 1: Шаг 1

На второй итерации:

	A	**B**	**C**	**D**	**E**
	0	-1	2	∞	1
	0	-1	2	1	1
	0	-1	2	-2	1

Для оптимизации алгоритм можно остановить, если на очередной итерации веса вершин не изменились. Для доказательства корректности необходимо показать, что алгоритм сходится, т.е. что за $|V|-1$ итераций будут гарантировано получены кратчайшие пути до всех вершин. Алгоритм выполняет только релаксацию, т.е. можно использовать леммы, доказанные для алгоритма Дейкстры.

У алгоритма Беллмана-Форда есть интересная особенность: его можно выполнять на многопроцессорной системе. Шаг релаксации можно выполнять независимо на каждом из узлов.

Лемма корректности: если в графе $G = (V, E)$ нет циклов с отрицательным весом, тогда алгоритм Беллмана-Форда завершается с $d[v] = \delta(s, v)$.

Необходимо доказать, что $d[v] \leftarrow \delta(s, v)$ в какой-то момент времени, т.к. по лемме о релаксации, $d[v]$ может только уменьшаться и никогда не станет меньше необходимого значения.

Пусть $p = v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_k$, где $v_0 = s$, $v_k = v$ – кратчайший путь из s в v, который содержит минимальное количество вершин. Тогда p – это простой путь, т.е. на нём нет повторяющихся вершин.

По лемме об оптимальной подструктуре: $\delta(s, v_i) = \delta(s, v_{i-1}) + w(v_{i-1}, v_i))$.

Для доказательства по индукции необходимо проверить базовый случай: $d[v_0] = 0 = \delta(s, v_0) = \delta(s, s)$. База индукции выполняется, т.к. в графе нет отрицательных циклов.

Пусть $d[v_j] = \delta(s, v_j)$ после j-го шага итерации алгоритма, где $j < i$. Тогда на i-м шаге после релаксации дуги (v_{i-1}, v_i), $d[v_i]$ станет равным $\delta(s, v_{i-1}) + w(v_{i-1}, v_i) = \delta(s, v_i)$. Таким образом, после k итераций будет получено правильное значение пути для $d[k]$. Так как количество вершин в простом пути в графе не превышает $|V| - 1$, такое же количество итераций необходимо, чтоб получить все пути.

Следствие: если алгоритм Беллмана-Форда не сошелся после $|V| - 1$ итераций (т.е. дальнейшие релаксации возможны), то в графе есть циклы с отрицательным весом.

3 Линейное программирование

Дана матрица A размера $m \times n$ и m-элементный вектор b и n-элементный вектор c. Найти вектор x размера n, такой, что $c^T \cdot x$ (целевая функция) максимально и $A \cdot x \leqslant b$ (выполняются ограничения), если такой вектор существует.

Существует много методов решения задачи линейного программирования. Например:

- симплекс-метод (экспоненциальное время в худшем случае)
- метод эллипсоидов (полиноминальное время)
- метод внутренней точки

Задача разностных ограничений является частным случаем задачи линейного программирования со следующими вариациями:

- ответ – любое допустимое решение (нет ограничения целевой функции)
- в каждой строке матрицы ограничений A один элемент 1, один -1, остальные – нули

Т.е. каждое ограничение имеет форму:

$$x_j - x_i \leqslant w_{ij} \tag{1}$$

Пример задачи:

$$\begin{aligned} x_1 - x_2 &\leqslant 3 \\ x_2 - x_3 &\leqslant -2 \\ x_1 - x_3 &\leqslant 2 \end{aligned}$$

Ответ:

$$\begin{aligned} x_1 &= 3 \\ x_2 &= 0 \\ x_3 &= 2 \end{aligned}$$

Задачу разностных ограничений можно представить в виде графа (ограничений) $G = (V, E)$, где $|V| = n$, $|E| = m$. Каждое неравенство вида $x_j - x_i \leqslant w_{ij}$ представляется в виде дуги между вершинами: Связь меж-

Рис. 2: Дуга ограничений

ду графом и ограничениями можно увидеть, переписав неравенство (1) в виде:

$$\begin{aligned} x_j &\leqslant x_i + w_{ij} \\ d[j] &\leqslant d[i] + w(i, j) \end{aligned}$$

Теорема: если в графе ограничений есть циклы отрицательного веса, то соответствующая задача разностных ограничений не имеет допустимого решения.

Доказательство: пусть $v_1 \to v_2 \to \ldots \to v_k \to v_1$ – путь весом строго меньше нуля. Тогда ему соответствуют следующие ограничения:

$$\begin{aligned} x_2 - x_1 &\leqslant w_{12} \\ x_3 - x_2 &\leqslant w_{23} \\ &\ldots \\ x_k - x_{k-1} &\leqslant w_{k-1k} \\ x_1 - x_k &\leqslant w_{k1} \end{aligned}$$

Если сложить обе неравенства, получим:

$$0 \leqslant w(\text{пути}) < 0$$

Т.о. система неравенств не имеет допустимого решения.

Теорема: если в графе G нет циклов отрицательного веса, то у задачи разностных ограничений есть решение.

Доказательство: добавить в граф G вершину s соединенную с каждой вершиной $v \in V$ дугой веса 0. В этом графе нет циклов отрицательного

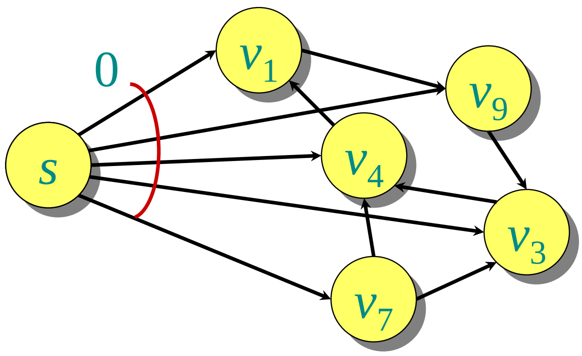

Рис. 3: Граф ограничение

веса и есть кратчайшие пути из s в каждую вершину весом не более 0. Обозначим $x_i = \delta(s, v_i)$.

Неравенство $x_j - x_i \leqslant w_{ij}$ эквивалентно

$$\delta(s, v_i) - \delta(s, v_j) \leqslant w_{ij} \iff$$
$$\iff \delta(s, v_i) \leqslant \delta(s, v_j) + w_{ij}$$

Последнее неравенство истино, т.к. это неравенство треугольника. Следовательно исходное неравенство также выполняется.

Таким образом, задачу разностных ограничений из m неравенств с n переменными можно преобразовать в граф, а затем решить с помощью алгоритма Беллмана-Форда за время $T = O(mn)$.

4 Заключительные замечания

Алгоритм Беллмана-Форда используется в протоколах маршрутизации семейста “distance-vector routing”, например в протоколе RIP версий 1 и 2.