Комбинаторные алгоритмы

Пути в сетях. Алгоритм Дейстры

Гальперин Александр Леонидович

2018 г.

Пути в сетях

Случай неотрицательных весов

Алгоритм Дейкстры

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Алгоритм Дейкстры позволяет вычислять расстояние от фиксированной вершины s до всех остальных вершин и находить кратчайшие пути более эффективно, чем алгоритм Форда–Баллмана.

Алгоритм был предложен Эдсгером Дейкстрой в 1959 году.

В основе алгоритма Дейкстры лежит принцип «жадности», заключающийся в последовательном вычислении расстояний сначала до ближайшей к s вершине, потом — до следующей ближайшей и т.д.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Для удобства изложения обозначим через $d(v)$ расстояние от s до v, т.е. длину кратчайшего (s, v)–пути в графе G.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Первая ближайшая к вершине s вершина v находится просто: это сама вершина s, находящаяся на нулевом расстоянии от s:

$$d(s) = 0.$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Первая ближайшая к вершине s вершина v находится просто: это сама вершина s, находящаяся на нулевом расстоянии от s:

$$d(s) = 0.$$

Пусть ближайшие k вершин к вершине s определены, и для всех них вычислены расстояния $d(v)$, т.е. определено множество

$$S = \{v_1 = s, v_2, \ldots, v_k\},$$

причем выполнены неравенства:

$$0 = d(v_1) \leqslant d(v_2) \leqslant \ldots \leqslant d(v_k); \tag{1}$$

$$d(v_k) \leqslant d(v), \text{для всех } v \in F, \text{где } F = V \backslash S. \tag{2}$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

$\mathcal{NB}$

Неравенство (2) имеет «потенциальный» характер: мы считаем известными значения расстояний лишь для вершин $v_1, \ldots, v_k$, а для всех остальных вершин расстояние еще не вычислены, но для них известно, что неравенства (2) будут выполняться.

Найдем следующую ближайшую вершину к s в сети G. Для каждого $w \in F$ положим

$$D[w] = \min\{d(v) + c(v, w) | v \in S\}.$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

$\mathcal{NB}$

$D[w]$ определяет длину минимального (s, w)–пути среди всех (s, w)–путей, все вершины в которых, кроме вершины w, принадлежат S.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Выберем теперь $w^* \in F$ такую, что

$$D[w^*] = \min\{D[w] | w \in F\}.$$

Выберем теперь $w^* \in F$ такую, что

$$D[w^*] = \min\{D[w] | w \in F\}.$$

Оказывается, что вершина w^* является самой близкой к s среди всех вершин, не входящих в S (она является $(k+1)$–ой ближайшей к s вершиной), и, более того, расстояние от вершины s до вершины w^* в точности равно $D[w*]$, т.е. справедливо равенство

$$d(w^*) = D[w^*].$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Обоснуем сначала равенство $d(w^*) = D[w^*]$.
Пусть $P : s = w_0, w_1, \ldots, w_r = w^*$ — произвольный (s, w^*)–путь в сети G. Достаточно доказать неравенство $D[w^*] \leqslant c(P)$.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Обоснуем сначала равенство $d(w^*) = D[w^*]$.
Пусть $P : s = w_0, w_1, \ldots, w_r = w^*$ — произвольный (s, w^*)–путь в сети G. Достаточно доказать неравенство $D[w^*] \leqslant c(P)$.

Среди всех вершин пути P выберем вершину w_j с наименьшим номером среди тех, которые не входят во множество S.Т.к. начальная вершина входит в S, а конечная — нет, то такой номер найдется.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Итак,

$$w_{j-1} \in S, w_j \in F.$$

Тогда из определение $D[w]$, где $w \in F$, и определения стоимости пути вытекают неравенства

$$D[w_j] \leqslant d(w_{j-1}) + c(w_{j-1}, w_j) \leqslant c(w_{j-1}, w_1) + \ldots + c(w_{j-1}, w_j) = c(Q),$$

где через Q обозначен (s, w_j)–подпуть пути P.

Итак,

$$w_{j-1} \in S, w_j \in F.$$

Тогда из определение $D[w]$, где $w \in F$, и определения стоимости пути вытекают неравенства

$$D[w_j] \leqslant d(w_{j-1}) + c(w_{j-1}, w_j) \leqslant c(w_{j-1}, w_1) + \ldots + c(w_{j-1}, w_j) = c(Q),$$

где через Q обозначен (s, w_j)–подпуть пути P.

Из условия неотрицательности весов дуг следует, что

$$c(Q) \leqslant c(P).$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Кроме того, в силу выбора w^*, имеем:

$$D[w^*] \leqslant D[w_j].$$

С учетом этих неравенств получаем:

$$D[w^*] \leqslant c(P).$$

А, следовательно, $d(w^*) = D[w^*]$. $\square$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Осталось убедиться, что вершина w^* — $(k+1)$–ая ближайшая к s вершина.

Для этого достаточно доказать, что $d(w^*) \leqslant d(w)$ для всех $w \in F$.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Зафиксируем (s, w^*)–путь P_1 и (s, w)–путь P_2, для которых

$$c(P_1) = d(w^*) \quad \text{и} \quad c(P_2) = d(w).$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Зафиксируем (s, w^*)–путь P_1 и (s, w)–путь P_2, для которых

$$c(P_1) = d(w^*) \quad \text{и} \quad c(P_2) = d(w).$$

В силу доказанного равенства $d(w^*) = D[w^*]$ можно считать, Что в пути P_1 все вершины, кроме w^*, лежат в S.

Тогда справедливо неравенство

$$D[w^*] \leqslant D[w]$$

и, повторяя вышеприведенные утверждения, приходим к неравенству

$$D[w^*] \leqslant c(P).$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

А отсюда следует, что

$$c(P_1) = d(w^*) = D[w^*] \leqslant c(P_2) = d(w).$$

$\square$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Итак, выбирая вершину $w \in F$ с минимальным значением $D[w]$ и добавляя в S, мы расширяем множество вершин, до которых вычислено расстояние, на один элемент. Следовательно, повторяя $n-1$ раз процесс расширения S, мы вычислим расстояние до всех вершин сети G.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

При формальной записи алгоритма Дейкстры ход вычисления расстояний от s до всех вершин v будет отражать в массиве D.

По окончании работы алгоритма равенства $D[v] = d(v)$ будут выполнены для всех $v \in V$.

Без формального описания используется функция $Min(F)$, которая возвращает вершину $w \in F$:

$$D[w] = \min \{D[v] | v \in F\}.$$

Иначе говоря, она находит тот самый элемент , который следует добавить к S и удалить из F.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Мы ограничимся только вычислением расстояний и меток *Previous*, с помощью которых кратчайшие пути строятся точно так же, как в алгоритме Форда–Беллмана.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

(∗ нахождение расстояний от фиксированной вершины до всех остальных в сети с неотрицательными весами ∗)
Вход: сеть $G = (V, E, c)$, заданная матрицей весов A порядка n; выделенная вершина s.

Выход: расстояния $D[v]$ от s до всех вершин $v \in V$, $Previous[v]$ — предпоследняя вершина в кратчайшем (s, v)-пути.

```
begin
  D[s] := 0; Previous[s] := 0; F := V \ {s};
  for v ∈ F do
    begin D[v] := A[s, v]; Previous[v] := s; end;
  for k := 1 to n − 1 do
    begin
      w := Min(F); F := F \ {w};
      for v ∈ F do
        if D[w] + A[w, v] < D[v] then
          begin
            D[v] := D[w] + A[w, v];
            Previous[v] := w;
          end
    end
end.
```

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Для доказательства корректности алгоритма заметим, что при входе в очередную k–ую итерацию цикла 5–13 уже определены k ближайших к s вершин и вычислены расстояния от s до каждой из них.

```
5.   for k := 1 to n − 1 do
6.     begin
7.       w := Min(F); F := F \ {w};
8.       for v ∈ F  do
9.         if D[w] + A[w, v] < D[v] then
10.          begin
11.            D[v] := D[w] + A[w, v];
12.            Previous[v] := w;
13.          end
```

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Пр этом первоначальные значения расстояний вычислены в строке 4.

4. **begin** $D[v] := A[s, v]$; $Previous[v] := s$; **end;**

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Как показано выше, выполнение строки 7 правильно определяет $(k+1)$–ую ближайшую к s вершину.

$$7. \qquad w := Min(F);\ F := F \setminus \{w\};$$

Удаление w из F влечет, что $D[w]$ больше меняться не будет, а по доказанному выше пересчитывать его нет нужды, т.к. выполняется условие $D[w] = d(w)$.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Осталось заметить, что проверка в строке 9 в каждой итерации цикла 5–13 позволяет правильно пересчитать значения $D[v]$ и отслеживать предпоследнюю вершину в кратчайшем пути.

```
5.    for k := 1 to n − 1 do
6.      begin
7.        w := Min(F); F := F \ {w};
8.        for v ∈ F  do
9.          if D[w] + A[w, v] < D[v] then
10.           begin
11.             D[v] := D[w] + A[w, v];
12.             Previous[v] := w;
13.           end
```

Следовательно, алгоритм правильно вычисляет расстояния и кратчайшие пути от фиксированной вершины до всех остальных вершин сети.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Теорема

Алгоритм Дейкстры имеет сложность $O(n^2)$.

Доказательство. Пусть найдены расстояния до k ближайших к s вершин. Определение расстояния до $(k+1)$–ой вершины требует в строке 7 числа операций пропорционального $(n-k)$, т.к. именно столько вершин находится во множестве F.

$$7. \qquad w := Min(F);\ F := F \setminus \{w\};$$

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

Проверка условия в строке 9, и, если понадобится, пересчет $D[v]$ и $Previous[v]$ в строках 11 и 12, требует числа операций, пропорционального $n - k$,

```
9.     if D[w] + A[w, v] < D[v] then
10.      begin
11.        D[v] := D[w] + A[w, v];
12.        Previous[v] := w;
```

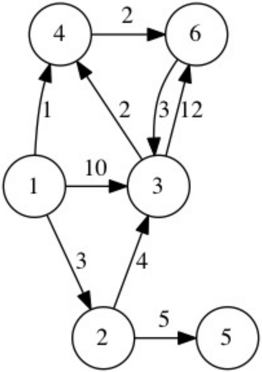
Окончательно, общее число операций в алгоритме Дейкстры пропорционально

$$n + (n-1) + \ldots + 1 = \frac{n(n-1)}{2},$$

т.е. равно $O(n^2)$. ∎

Пути в сетях

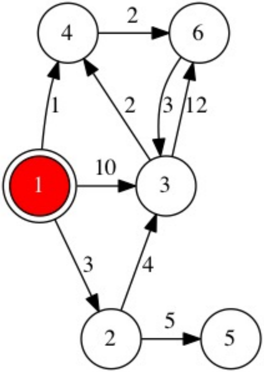
Случай неотрицательных весов. Алгоритм Дейкстры



Рассмотрим работу алгоритма Дейкстры на примере следующего графа.

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

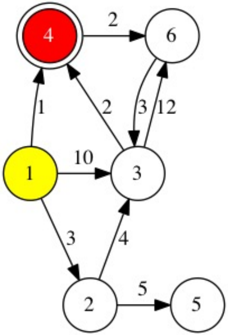


Просмотренные вершины будем помечать красным цветом. Проведем инициализацию. Уже вычисленные расстояния выделяем в таблице.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
$1 = s$	0	3	10	1	∞	∞
ПРЕДШ	0	1	1	1	1	1

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

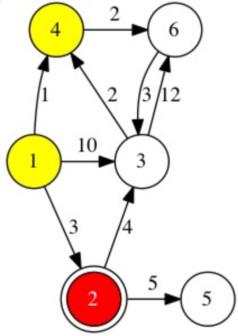


Выбираем «минимальную» вершину из F — вершину 4. Пересчитываем расстояния.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
стар	0	3	10	1	∞	∞
$1, 4$	0	3	10	1	∞	3
ПРЕДШ	0	1	1	1	1	4

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

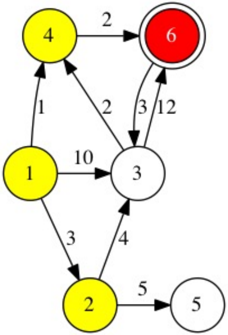


Выбираем «минимальную» вершину из F — вершину 2. Пересчитываем расстояния.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
стар	0	3	10	1	∞	3
$1, 4, 2$	0	3	7	1	8	3
ПРЕДШ	0	1	2	1	2	4

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

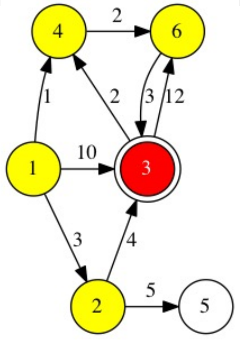


Выбираем «минимальную» вершину из F — вершину 6. Пересчитываем расстояния.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
стар	0	3	7	1	8	3
$1, 4, 2, 6$	0	3	6	1	8	3
ПРЕДШ	0	1	6	1	2	4

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры

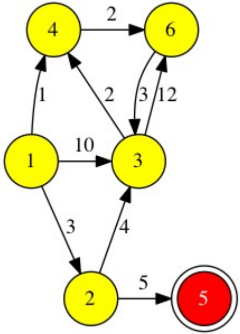


Выбираем «минимальную» вершину из F — вершину 3. Пересчитываем расстояния.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
стар	0	3	6	1	8	3
$1, 4, 2, 6, 3$	0	3	6	1	8	3
ПРЕДШ	0	1	6	1	2	4

Пути в сетях

Случай неотрицательных весов. Алгоритм Дейкстры



Выбираем «минимальную» вершину из F — вершину 5. Пересчитываем расстояния. Кратчайшие пути построены.

$S = V \setminus F$	$D[1]$	$D[2]$	$D[3]$	$D[4]$	$D[5]$	$D[6]$
стар	0	3	6	1	8	3
$1, 4, 2, 6, 3, 5$	0	3	6	1	8	3
ПРЕДШ	0	1	6	1	2	4