



O Algoritmo de Bellman-Ford

Notas de Aula nº 10
13 de dezembro de 2012

Ciência da Computação
Grafos
Prof. Leandro Zatesko

1	Ciclos negativos em grafos com pesos nas arestas	1
2	O Algoritmo de Bellman-Ford	2
3	Análise do Algoritmo de Bellman-Ford	2
3.1	Demonstração da parada	2
3.2	Demonstração da corretude	2
3.3	Análise da complexidade de tempo	3
3.4	Análise da complexidade de espaço	4
4	Uma vantagem no uso do Algoritmo de Bellman-Ford	4
5	Exercícios	4

O Algoritmo de Dijkstra, proposto em 1959, apresenta complexidade de tempo $O(|E(G)| + |V(G)| \log |V(G)|)$ quando implementado com uma *heap* de Fibonacci ao invés de uma *heap* binária. Em termos assintóticos, esta é a melhor complexidade de tempo conhecida mesmo nos dias de hoje para o *Problema do Caminho Mínimo*. O Algoritmo de Dijkstra, entretanto, não funciona se houver arestas com pesos negativos. Nestas Notas apresentamos um algoritmo que, embora mais ineficiente que o Algoritmo de Dijkstra, funciona mesmo havendo arestas com pesos negativos. Trata-se do *Algoritmo de Bellman-Ford*, desenvolvido por Richard Bellman e por Lester Ford Jr. entre 1958 e 1962.

1 Ciclos negativos em grafos com pesos nas arestas

DEFINIÇÃO 1.1. Um *ciclo negativo* num grafo com pesos nas arestas (V, E, ρ) é um ciclo com no mínimo 3 vértices tal que a soma dos pesos das arestas do ciclo é menor que zero. ciclo negativo

No grafo da Figura 1, um ciclo negativo é o ciclo definido por $C = 1, 2, 3, 4, 7, 6, 1$, pois a soma dos pesos das arestas deste ciclo é -3 . O Algoritmo de Bellman-Ford não funciona em grafos com ciclos negativos, a menos que esses ciclos estejam numa componenta conexa distinta daquela em que se encontra a origem s sobre a qual se executará o Algoritmo, ou seja, a menos que esses ciclos não sejam *alcançáveis* a partir de s .

Como era de se esperar, a desigualdade triangular continua valendo mesmo que haja arestas com pesos negativos.

TEOREMA 1.2. Para todo grafo G com peso nas arestas e toda tripla de vértices de G distintos s , u e w tais que $w \in N_G(u)$ vale que

$$\text{dist}(s, w) \leq \text{dist}(s, u) + \rho(\{u, w\}),$$

ainda que haja arestas com pesos negativos e ainda que haja ciclos negativos.

Se um ciclo negativo for inalcançável a partir de s , não há problema, já que a distância de qualquer vértice do ciclo a partir de s vai ser ∞ de qualquer jeito.

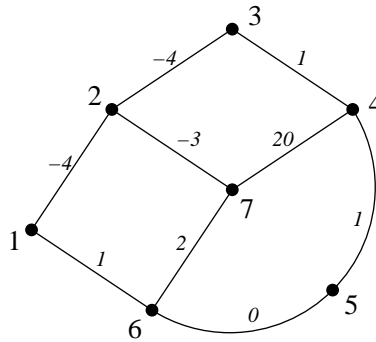


Figura 1

DEMONSTRAÇÃO. Exercício 4. ♦

COROLÁRIO 1.3. Se $P = v_0 v_1 \dots v_k$ é um caminho mínimo num grafo com pesos nas arestas, ainda que haja arestas com pesos negativos e ainda que haja ciclos negativos, o subcaminho $P^i = v_0 v_1 \dots v_i$ é também mínimo para todo $i \in [0..k]$.

DEMONSTRAÇÃO. Exercício 5. ♦

2 O Algoritmo de Bellman-Ford

```

BELLMAN-FORD( $G, s$ ):
1  para todo  $v \in V(G)$ , faça:
2     $dist[v] \leftarrow \infty$ ;
3     $pai[v] \leftarrow v$ ;
4   $dist[s] \leftarrow 0$ ;
5  para  $j$  de 1 até  $|V(G)| - 1$ , faça:
6    para todo  $u \in V(G)$ , faça:
7      para todo  $w \in N_G(u) \setminus \{pai[u]\}$ , faça:
8        se  $dist[w] > dist[u] + \rho(\{u, w\})$ , então,
9           $dist[w] \leftarrow dist[u] + \rho(\{u, w\})$ ;
10          $pai[w] \leftarrow u$ ;
11  para todo  $u \in V(G)$ , faça:
12    para todo  $w \in N_G(u) \setminus \{pai[u]\}$ , faça:
13      se  $dist[w] > dist[u] + \rho(\{u, w\})$ , então,
14        devolva "ERRO! Há ciclos negativos alcançáveis a partir da origem!"
15  devolva  $dist, pai$ .

```

Algoritmo 1

3 Análise do Algoritmo de Bellman-Ford

3.1 Demonstração da parada

Todos os laços são executados um número finito de vezes, então, o Algoritmo para.

3.2 Demonstração da corretude

Note que este Lema não vale se G possui ciclos negativos alcançáveis a partir de s . Tente achar um contraexemplo!

A demonstração é mais ou menos a que fizemos na aula do Algoritmo de Dijkstra, mas você precisa tomar cuidado com os ciclos negativos!

LEMA 3.1. Se G não possui ciclos negativos alcançáveis a partir de s , então, em qualquer momento da execução de BELLMAN-FORD(G, s) vale para todo $u \in V(G)$ que $dist[u] \geq dist(s, u)$ e, uma vez que $dist[u] \leftarrow dist(s, u)$, $dist[u]$ nunca mais é alterado.

DEMONSTRAÇÃO. Exercício 6. ♦

LEMA 3.2. Se $\text{dist}(s, v) = \infty$, então, $\mathbf{dist}[v] = \infty$ durante toda a execução do Algoritmo.

DEMONSTRAÇÃO. Exercício 7. ◆

LEMA 3.3. Em qualquer momento da execução do Algoritmo vale para todo vértice v que $\mathbf{dist}[v] = \mathbf{dist}[\mathbf{pai}[v]] + \rho(\mathbf{pai}[v], v)$.

DEMONSTRAÇÃO. Exercício 8. ◆

LEMA 3.4. Num grafo G com pesos possivelmente negativos nas arestas mas sem ciclos negativos alcançáveis a partir de s , se $P = v_0 v_1 v_2 \dots v_{k-1} v_k$ é um caminho mínimo entre $s = v_0$ e $w = v_k$, então, para todo $i \in [0..k]$, após i iterações do laço da linha 5 de $\text{BELLMAN-FORD}(G, s)$ vale que $\mathbf{dist}[v_j] = \text{dist}(s, v_j)$ para todo $j \in [0..i]$.

DEMONSTRAÇÃO. Para $i = 0$ após 0 iterações do laço da linha 5 vale que $\mathbf{dist}[v_0] = \mathbf{dist}[s] = 0 = \text{dist}(s, s)$. Se i , entretanto, é maior que 0, suponhamos por indução que para todo $i' < i$ valha, após i' iterações do laço da linha 5, que $\mathbf{dist}[v_j] = \text{dist}(s, v_j)$ para todo $j \in [0..i']$. Então, da hipótese da indução, após $i - 1$ iterações, $\mathbf{dist}[v_{i-1}] = \text{dist}(s, v_{i-1})$. Assim, na i -ésima iteração do laço da linha 7, quando u for tomado $u = v_{i-1}$:

- Se $\mathbf{pai}[u] \neq v_i$, então, quando w for tomado $w = v_i$ a relaxação fará valer que $\mathbf{dist}[v_i] \leq \mathbf{dist}[v_{i-1}] + \rho(\{u, w\})$. Porém, como $\mathbf{dist}[v_{i-1}] = \text{dist}(s, v_{i-1})$, e como $\text{dist}(s, v_{i-1}) + \rho(\{u, w\}) = c(P^i)$ sendo $P^i = v_0 \dots v_i$, e como do Corolário 1.3 P^i é mínimo, temos que $c(P^i) = \text{dist}(s, v_i)$ e portanto que

$$\mathbf{dist}[v_i] \leq \text{dist}(s, v_i).$$

Por outro lado, já que o grafo não possui ciclos negativos alcançáveis a partir de s , vale o Lema 3.1, o que nos traz que

$$\mathbf{dist}[v_i] \geq \text{dist}(s, v_i)$$

e por conseguinte que $\mathbf{dist}[v_i] = \text{dist}(s, v_i)$, como queríamos.

- Se $\mathbf{pai}[u] = v_i$, então, w não poderá ser tomado v_i para acertar $\mathbf{dist}[v_i]$, mas isso nem será necessário. Vejamos: como $\mathbf{dist}[v_{i-1}] = \text{dist}(s, v_{i-1})$ e como, do Lema 3.3, $\mathbf{dist}[v_{i-1}] = \mathbf{dist}[v_i] + \rho(v_i, v_{i-1})$, temos que

$$\text{dist}(s, v_{i-1}) = \mathbf{dist}[v_i] + \rho(v_i, v_{i-1}). \quad (1)$$

Do Lema 3.1, uma vez que o grafo não possui ciclos negativos a partir de s , $\mathbf{dist}[v_i] \geq \text{dist}(s, v_i)$, e, portanto, $\text{dist}(s, v_{i-1}) \geq \text{dist}(s, v_i) + \rho(v_i, v_{i-1})$. Por outro lado, do Teorema 1.2, $\text{dist}(s, v_{i-1}) \leq \text{dist}(s, v_i) + \rho(v_i, v_{i-1})$. Concluimos dessarte que $\text{dist}(s, v_{i-1}) = \text{dist}(s, v_i) + \rho(v_i, v_{i-1})$, o que pela Equação 1 significa que $\mathbf{dist}[v_i] = \text{dist}(s, v_i)$, como desejávamos. ◆

TEOREMA 3.5. Se G não possui ciclos negativos alcançáveis a partir da origem s , após a execução de $\text{BELLMAN-FORD}(G, s)$ vale que $\mathbf{dist}[u] = \text{dist}(s, u)$ para todo u .

DEMONSTRAÇÃO. Seja u um vértice qualquer, seja P um caminho mínimo de s a u e seja k o número de arestas de P . Como P é um caminho, não pode repetir vértices, o que nos traz que $k \leq |V(G)| - 1$. Então, do Lema 3.4, após k iterações do laço da linha 5 vale que $\mathbf{dist}[u] = \text{dist}(s, u)$. Ainda, do Lema 3.1, $\mathbf{dist}[u]$ nunca mais é alterado depois que atinge $\text{dist}(s, u)$.

Por fim, quando o laço da linha 5 terminar, a execução das linhas 11–14 não encontrará nenhuma aresta $\{u, w\}$ passível de relação, pois todos os vértices u já terão seu valor $\mathbf{dist}[u]$ certo. A execução cairá portanto na linha 15 e o Algoritmo devolverá o valor correto. ◆

3.3 Análise da complexidade de tempo

Esboçemos:

- o laço da linha 1 é iterado exatas $|V(G)|$ vezes, sendo que cada iteração custa tempo $O(1)$, já que podemos assumir que a estrutura `dist` é de acesso randômico e já que fazemos a análise da complexidade de tempo de um algoritmo em função do tamanho da entrada;
- a linha 4 também custa tempo $O(1)$;
- o laço da linha 5 é iterado exatas $|V(G)| - 1$ vezes, sendo que cada iteração custa:
 - o laço da linha 6 é iterado para todo vértice u , sendo que cada iteração custa $d_G(u)O(1)$, já que podemos assumir que $O(1)$ é o tempo das linhas 8–10;
- o laço da linha 11 é iterado para todo vértice u , sendo que cada iteração custa $d_G(u)O(1)$, já que podemos assumir que $O(1)$ é o tempo das linhas 13–14.

Totalizando:

$$\begin{aligned} T_{\text{BELLMAN-FORD}}(|V(G)| + |E(G)|) &\leq |V(G)|O(1) + O(1) + \left((|V(G)| - 1) \sum_{u \in V(G)} d_G(u)O(1) \right) + \sum_{u \in V(G)} d_G(u)O(1) \\ &= O(|V(G)|) + |V(G)|(2|E(G)|O(1)) \\ &= O(|V(G)||E(G)|). \end{aligned}$$

3.4 Análise da complexidade de espaço

Não é difícil perceber que `dist` assume todos os vértices do grafo G , o que faz com que o Algoritmo de Bellman-Ford tenha uma indesejada complexidade de espaço $O(|V(G)|)$.

4 Uma vantagem no uso do Algoritmo de Bellman-Ford

A principal vantagem do Algoritmo de Bellman-Ford, que faz com que ele seja bastante utilizado em roteamento de redes, é que pode ser facilmente distribuído e facilmente aplicado num contexto dinâmico, em que o grafo está mudando o tempo todo. Numa rede, por exemplo, cada nó da rede armazena uma tabela com as distâncias conhecidas daquele nó até os demais nós da rede. Quando ocorre alguma mudança, o nó que verificou essa mudança dispara a informação para seus vizinhos, que vai ser propagada para toda a rede de ciclo em ciclo.

Computers are like air conditioners — they stop working properly if you open Windows.

Linus Torvalds?

5 Exercícios

EXERCÍCIO 1. Execute o Algoritmo de Bellman-Ford para o grafo da Figura 2 a partir da origem 1 e descubra um caminho mínimo de 1 a 5.

EXERCÍCIO 2. O grafo da Figura 3 possui ciclos negativos? O Algoritmo de Bellman-Ford funciona se executado a partir da origem 5? E se for executado a partir da origem 10?

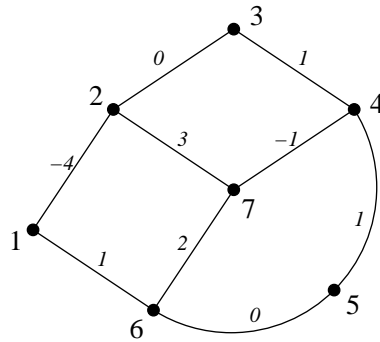


Figura 2

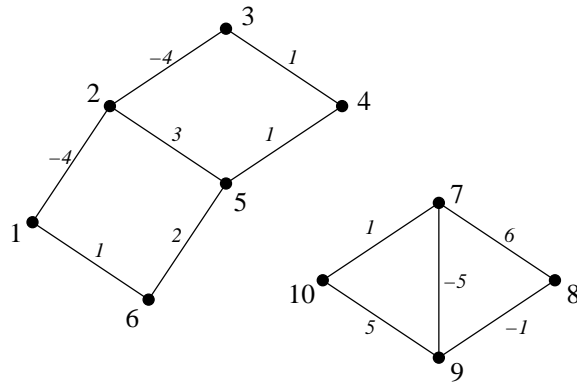


Figura 3

EXERCÍCIO 3. Execute o Algoritmo de Bellman-Ford para o grafo da Figura 1 (a partir de qualquer origem) e tente entender por que o Algoritmo falha quando há ciclos negativos alcançáveis a partir da origem.

EXERCÍCIO 4. Mostre o Teorema 1.2.

EXERCÍCIO 5. Mostre o Corolário 1.3.

EXERCÍCIO 6. Mostre o Lema 3.1.

EXERCÍCIO 7. Mostre o Lema 3.2.

EXERCÍCIO 8. Mostre o Lema 3.3.