PCC173/BCC463 - Otimização em Redes

Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

Conteúdo

Aviso

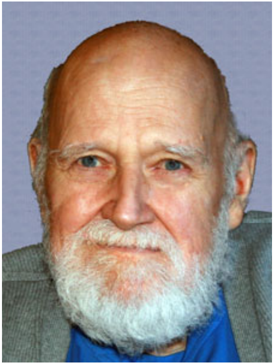
Fonte

Este material é baseado nos livros

- Goldbarg, M., & Goldbarg, E. (2012). *Grafos: conceitos, algoritmos e aplicações*. Elsevier.
- Goldbarg, M. C., & Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier.

Licença

Biografia



Lester Randolph Ford Jr.
⋆ 23/09/1927 † 26/02/2017

- Matemático americano;
- Especialista em fluxo em redes;
- Autor do algoritmo *Bellman-Ford*;
- Autor do teorema de corte mínimo/fluxo máximo;
- Co-autor do algoritmo *Ford-Fulkerson*.

Delbert Ray Fulkerson
⋆ 14/08/1924 † 10/01/1976

- Matemático americano;
- Em sua homenagem, a *Mathematical Programming Society* criou o *Fulkerson Prize*;
- Co-autor do algoritmo *Ford-Fulkerson*;
- Autor do algoritmo *Out-of-Kilter*, para o problema de fluxo de custo mínimo.

Algoritmo de Ford & Fulkerson

- O algoritmo de Ford & Fulkerson (1956) calcula o fluxo máximo em uma rede a partir de um fluxo viável;
- No caso de os limites inferiores de todos os arcos serem iguais a zero, o fluxo zero pode ser adotado – ou seja, $f(i,j) = 0$ para todo arco (i, j);
- O algoritmo rotula os vértices de R buscando encontrar um caminho de aumento de fluxo entre s e t. Caso tal caminho exista, o algoritmo aumenta o fluxo na rede;
- O rótulo de um vértice y qualquer segue o formato $[x, \pm, \xi_y]$
  - x: indica o vértice a partir do qual o vértice y foi rotulado;
  - $\pm$: indica rotulação a partir de um arco direto (+) ou reverso (-);
  - ξ_y: indica o quanto o fluxo pode ser aumentado no caminho de s até o vértice y.
- O processo é repetido até que não seja mais possível encontrar um caminho de aumento de fluxo, situação na qual o fluxo máximo está circulando na rede.

Algoritmo de Ford & Fulkerson

Atenção para os detalhes

- Não é possível selecionar arcos nos quais os dois vértices já foram rotulados;
- Somente os caminhos que terminam em t devem ser considerados;
- Não se esqueça dos arcos reversos;
- Não se esqueça de atualizar o grafo de aumento de fluxo, adicionando e removendo arcos adequadamente;
- Se todos os arcos que partem do vértice s tiverem a capacidade esgotada, a execução do algoritmo terminou.

Algoritmo de Ford & Fulkerson

Sendo f um fluxo viável na rede;
Rotular s com $[-\infty, 0, +\infty]$;
enquanto *existir vértice i rotulado incidente a um arco utilizável* **faça**
    se $a = (i, j)$ **então**
        //arco utilizável: j não rotulado e $f(i,j) < \bar{u}(i,j)$
        Rotular j com $[i, +, \xi_j]$, em que $\xi_j = \min\{\xi_i, \bar{u}(i,j)\text{-}f(i,j)\}$
    fim
    senão
        //arco utilizável $a = (j, i)$ com j não rotulado e $f(i,j) > \underline{u}(i,j)$
        Rotular j com $[i, -, \xi_j]$, em que $\xi_j = \min\{\xi_i, f(i,j)\text{-}\underline{u}(i,j)\}$
    fim
    se t *foi rotulado* **então**
        Construir o caminho P de aumento de fluxo a partir de t;
        Aumentar o fluxo nos arcos de P somando ξ_t nos arcos diretos e subtraindo ξ_t nos arcos reversos;
        Cancelar todos os rótulos (exceto o de s);
        Atualizar o grafo de aumento de fluxo;
    fim
fim

Algoritmo de Ford & Fulkerson

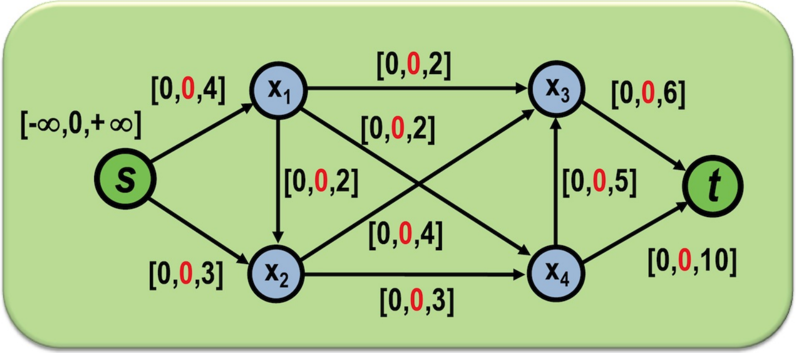
Complexidade

- No caso de a capacidade dos arcos ser inteira, a complexidade será $O(mf_{max})$;
- Uma variação deste algoritmo, o algoritmo de Edmonds–Karp, possui complexidade independente do fluxo: $O(nm^2)$.

Corte Mínimo

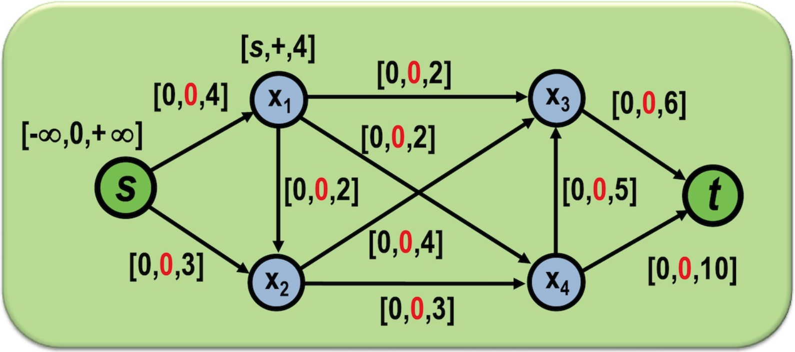
O fluxo determinado é máximo. Um corte s-t de capacidade mínima pode ser obtido colocando-se todos os vértices passíveis de rotulação[a] em X e os restantes em $\bar{X}$.

[a] Na última iteração do algoritmo não será possível rotular todos os vértices em um caminho até t.



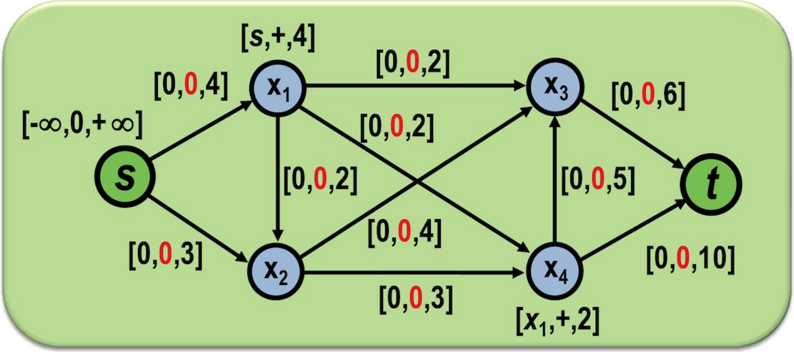
Rede de exemplo.
Fluxo viável f=0, logo, não há arcos reversos.
Vértice s rotulado.

Ford & Fulkerson – Exemplo



Primeira rotulação do vértice x_1.
O arco é direto.
$\xi_{x_1} = min\{+\infty, 4\}$

Ford & Fulkerson – Exemplo

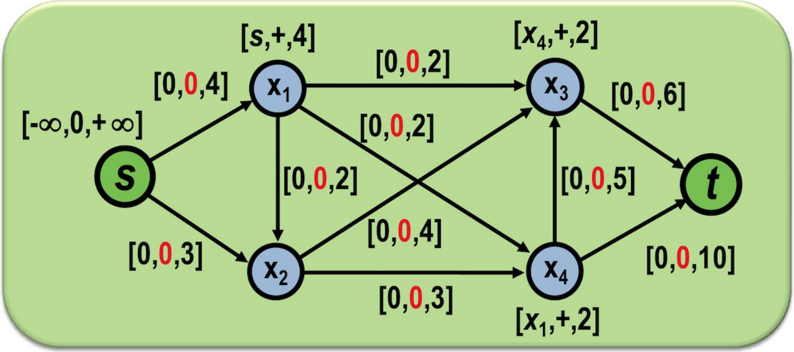


Primeira rotulação do vértice x_4.

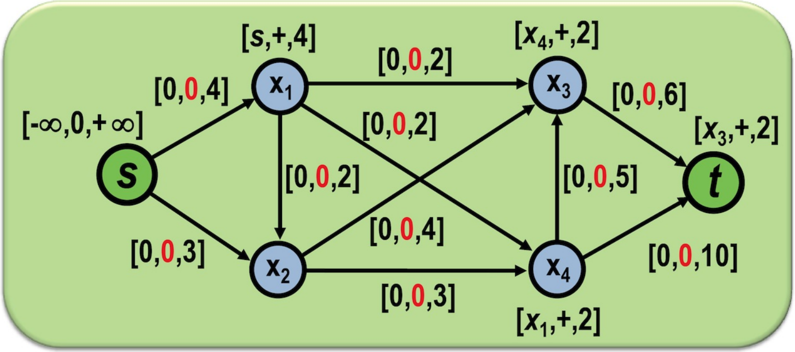
O arco é direto.

$\xi_{x_4} = min\{4, 2\}$

Ford & Fulkerson – Exemplo



Primeira rotulação do vértice x_3.
O arco é direto.
$\xi_{x_3} = min\{2, 5\}$



Primeira rotulação do vértice t.

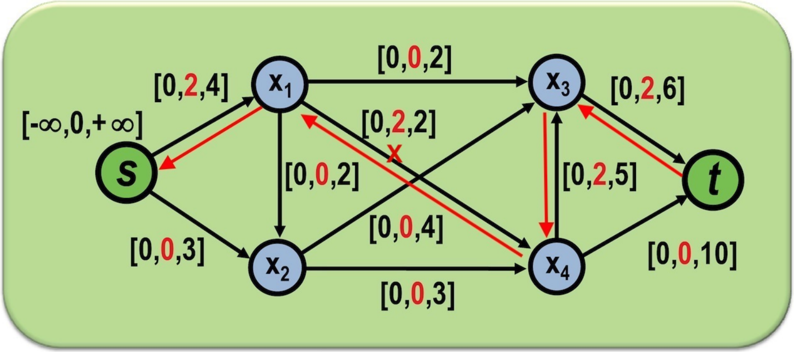
O arco é direto.

$\xi_t = min\{2, 6\}$

Algoritmo de Ford & Fulkerson

Caminho Encontrado

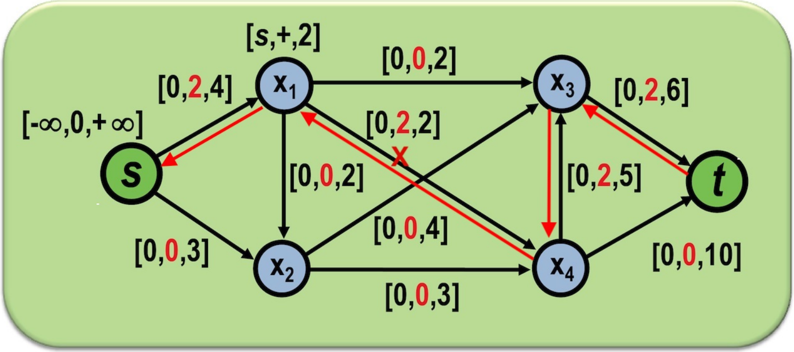
- Como o vértice t foi rotulado, um caminho de aumento de fluxo foi encontrado;
- O número de unidades de fluxo que podem ser aumentadas neste caminho é dado por $\xi_t = 2$;
- Recupera-se o caminho a partir de t, andando para trás: s, x_1, x_4, x_3, t, aumentado o fluxo em 2 nos arcos deste caminho;
- A capacidade do arco (x_1, x_4) foi esgotada;
- O algoritmo remove os rótulos de todos os vértices (exceto o de s) e é reiniciado.



Caminho com aumento de fluxo encontrado.

Rede com fluxo f=2.

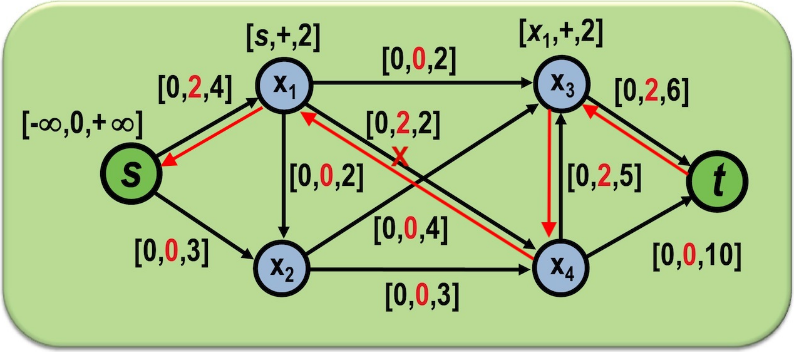
Note a inclusão de arcos reversos!



Segunda rotulação do vértice x_1.

O arco é direto.

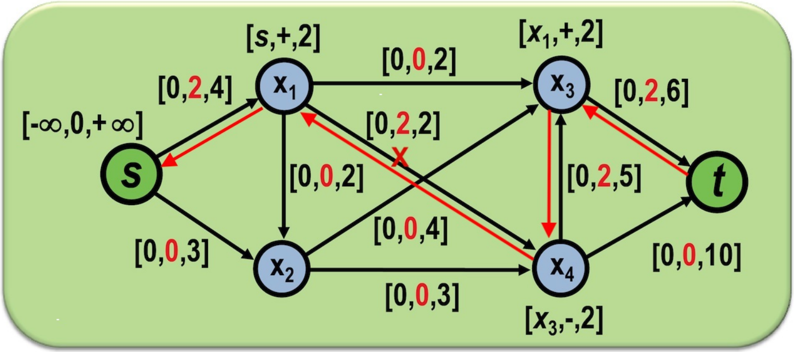
$$\xi_{x_1} = min\{+\infty, 2\}$$



Segunda rotulação do vértice x_3.

O arco é direto.

$$\xi_{x_3} = min\{2, 2\}$$

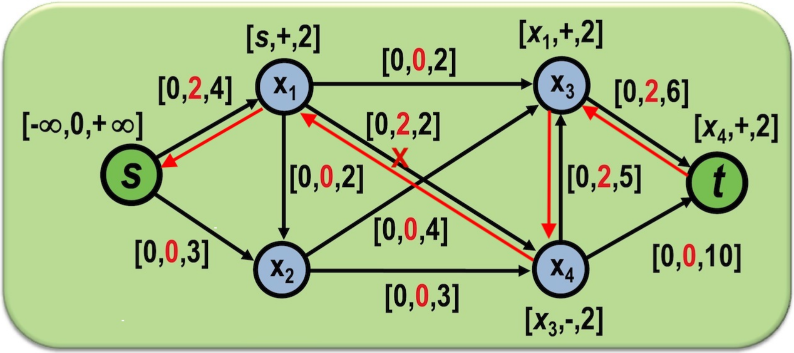


Segunda rotulação do vértice x_4.

O arco é reverso.

$$f(x_4, x_3) - \underline{u}(x_4, x_3) = 2$$

$$\xi_{x_4} = min\{2, 2\}$$



Segunda rotulação do vértice t.

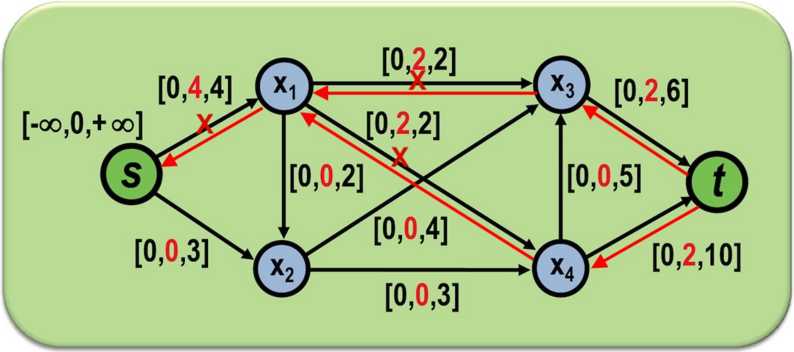
O arco é direto.

$\xi_t = min\{2, 10\}$

Algoritmo de Ford & Fulkerson

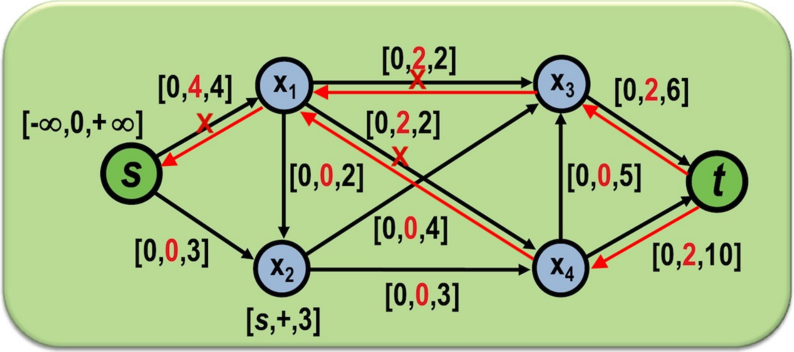
Caminho Encontrado

- Como o vértice t foi rotulado, um caminho de aumento de fluxo foi encontrado;
- O número de unidades de fluxo que podem ser aumentadas neste caminho é dado por $\xi_t = 2$;
- Recupera-se o caminho a partir de t, andando para trás: s, x_1, x_3, x_4, t, aumentado o fluxo em 2 unidades nos arcos deste caminho;
- As capacidades dos arcos (s, x_1) e (x_1, x_3) foram esgotadas;
- O algoritmo remove os rótulos de todos os vértices (exceto o de s) e é reiniciado.



O fluxo dos arcos diretos é aumentado em 2 unidades.
O fluxo do arco reverso é diminuído em 2 unidades.
Rede com fluxo f=4.

Ford & Fulkerson – Exemplo

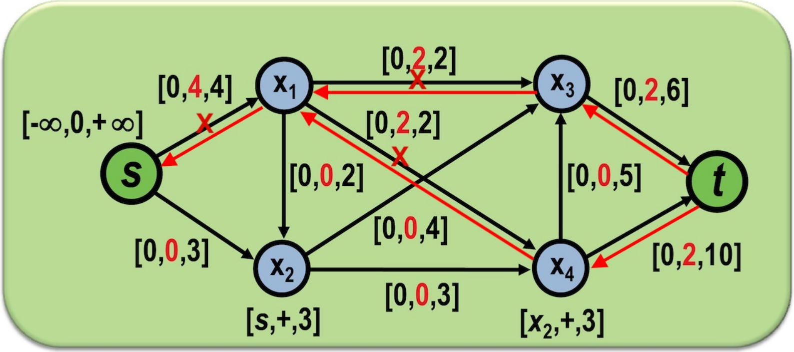


Primeira rotulação do vértice x_2.

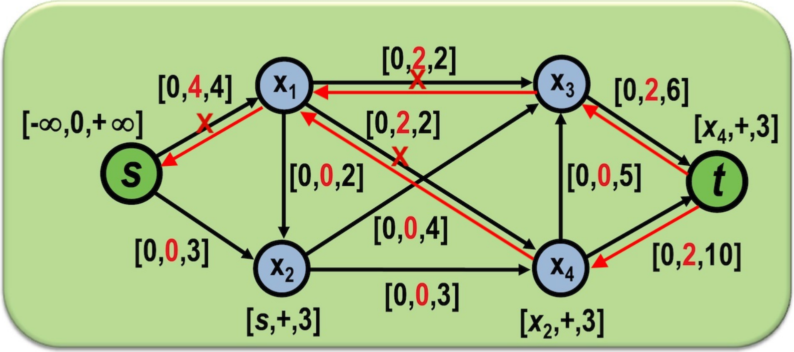
O arco é direto.

$$\xi_2 = min\{+\infty, 3\}$$

Ford & Fulkerson – Exemplo



Terceira rotulação do vértice x_4.
O arco é direto.
$\xi_4 = min\{3, 3\}$



Terceira rotulação do vértice t.

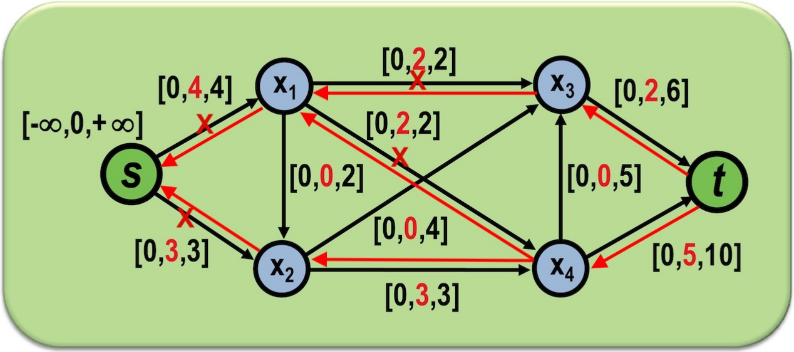
O arco é direto.

$$\xi_t = min\{3, 8\}$$

Caminho Encontrado

- Como o vértice t foi rotulado, um caminho de aumento de fluxo foi encontrado;
- O número de unidades de fluxo que podem ser aumentadas neste caminho é dado por $\xi_t = 3$;
- Recupera-se o caminho a partir de t, andando para trás: s, x_2, x_4, t, aumentado o fluxo em 3 nos arcos deste caminho;
- As capacidades dos arcos (s, x_2) e (x_2, x_4) foram esgotadas;
- O algoritmo remove os rótulos de todos os vértices (exceto o de s) e é reiniciado.

Ford & Fulkerson – Exemplo



O fluxo dos arcos diretos é aumentado em 3 unidades.
Não há arcos diretos.
Rede com fluxo f=7.

Ford & Fulkerson – Exemplo

Final do Algoritmo

- Após a rotulação do vértice t, o algoritmo remove todos os rótulos e reinicia;
- No entanto, não existe arco utilizável a partir de s, e desta forma, não há nenhum outro vértice a ser rotulado – o fluxo é máximo;
- O algoritmo termina quando o vértice t não puder mais ser rotulado – neste caso os vértices rotulados e os não rotulados definem também um corte mínimo em R;
- O único vértice rotulado é s, logo, o corte mínimo do exemplo é $X = \{s\}$ e $\bar{X} = \{x_1, x_2, x_3, x_4, t\}$.

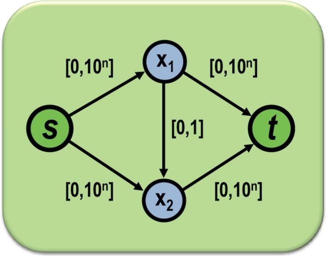
Ford & Fulkerson – Ponto fraco

Ponto Fraco

O algoritmo de Ford & Fulkerson pode apresentar um comportamento ineficiente caso enfrente alguns casos patológicos.

No caso abaixo, é possível que o algoritmo escolha alternadamente os caminhos de aumento de fluxo (s, x_1, x_2, t) e (s, x_2, x_1, t). Serão necessárias 2×10^n operações de aumento de fluxo.

Caso a escolha fosse (s, x_1, t) e (s, x_2, t), seriam necessárias apenas 2 iterações.



Dúvidas?