Aula 27 – Algoritmo de Ford-Fulkerson

Notas de Aula de Teoria dos Grafos

Prof[a.]: Patrícia D. L. Machado

UFCG – Unidade Acadêmica de Sistemas e Computação

Sumário

Nesta aula, apresentamos o algoritmo de Ford-Fulkerson que determina um fluxo máximo e um corte mínimo em uma rede de fluxos.

Definições Preliminares

A seguir, apresentamos alguns conceitos e terminologia utilizados no algoritmo de Ford-Fulkerson.

Tipos de Arco

Seja $N(x,y)$ uma rede de fluxo e seja f um fluxo nesta rede. Um arco a em $N(x,y)$ é:

- ***f-zero*** se $f(a) = 0$
- ***f-positivo*** se $f(a) > 0$
- ***f-não-saturado*** se $f(a) < c(a)$
- ***f-saturado*** se $f(a) = c(a)$

Como exemplo, considere a rede de fluxo na Figura 1, onde o par de valores em cada arco apresenta um valor de fluxo e a capacidade do arco, respectivamente. Os arcos são classificados em cada uma das definições acima da seguinte forma:

- ***f-zero***: (v3, v5), (v3, y)
- ***f-positivo***: todos, exceto os *f-zero*
- ***f-não-saturado***: (x, v1), (v1, v4), (v2, v1), (v2, v5), (v3, v5), (v3, y)

- ***f-saturado***: (x, v3), (v1, v3), (v3, v2), (v4, v5), (v5, y)

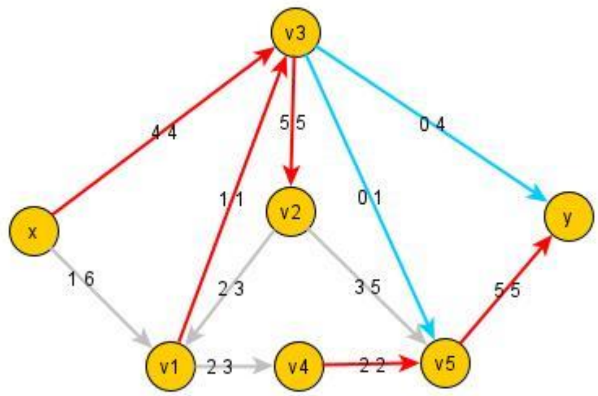


Figura 1

Seja W := ($v_0a_1v_1$, …, $v_{k-1}a_kv_k$) um passeio, não necessariamente dirigido, em um dígrafo.

Um arco a_i é **avante** (*forward*) se v_{i-1} é a cauda de a_i e v_i é sua cabeça.

Um arco a_i é **reverso** (*reverse*) se v_i é a cauda de a_i e v_{i-1} é sua cabeça.

Por exemplo, ignorando as direções dos arcos, considere o caminho T de x para y que está hachurado em laranja na Figura 2. Neste caminho, podemos observar que o arco $(x, v1)$ é avante, pois aponta na direção de x para y. Por outro lado, o arco $(v2, v1)$ é reverso, pois, neste caminho T, o arco tem direção para x e não para y.

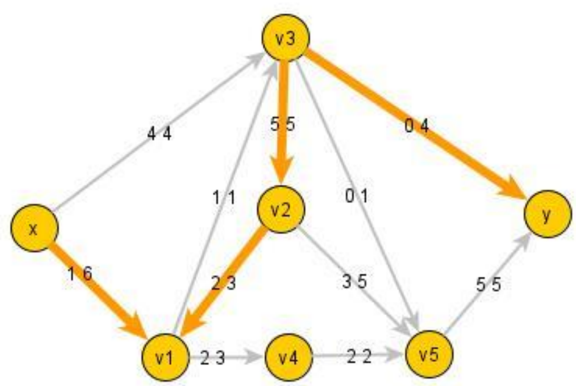


Figura 2

Como vimos na aula anterior, a fim de obter o fluxo máximo em uma rede, é necessário incrementar ou decrementar o valor do fluxo dos arcos. No algoritmo que estamos estudando nesta aula, a estratégia central é *incrementar arcos avante e decrementar arcos reverso*.

Incremento Mínimo

Considerando o grafo base de uma rede de fluxo $N(x, y)$ onde as direções dos arcos são ignoradas, seja T um caminho entre os vértices x e y.

A quantidade mínima de incremento ao valor do fluxo dos arcos de T, $\varepsilon(T)$, pode ser definida como o menor valor entre os valores máximos de incremento ou decremento para todos os arcos.

$\varepsilon(T) = \min\{\varepsilon(a): a \in A(T)\}$, onde:

$$\varepsilon(a) = \begin{cases} c(a) - f(a) & se\ a \text{ é } avante \\ f(a) & se\ a \text{ é } reverso \end{cases}$$

Ou seja, primeiro determinamos o valor máximo de incremento (se arco é avante) ou decremento (se arco é reverso) para cada arco, depois escolhemos o menor destes valores a fim de poder aplicar o mesmo valor (como incremento, se avante; como decremento, se reverso) a todos os arcos sem ultrapassar a capacidade definida para cada um.

O valor máximo de incremento de um arco a é definido como a diferença entre o valor da capacidade e o valor atual de fluxo de a quando o arco é avante, ou seja, considerando o caminho T, o arco é direcionado no sentido de x para y; neste caso, o algoritmo deverá incrementar o valor de fluxo. Caso contrário, quando o arco é reverso, ou seja, considerando o caminho T, o arco é direcionado no sentido de y para x o valor de incremento é o valor do próprio fluxo.

Para os arcos no caminho T apresentado na Figura 2 podemos encontrar os seguintes valores de para ε:

$$\varepsilon\big((x, v1)\big) = 5, \varepsilon\big((v2, v1)\big) = 2, \varepsilon((v3, v1)) = 5, \varepsilon\big((v3, y)\big) = 4$$

Neste caso, $\varepsilon(T) = 2$, o menor dentre os valores. Se aplicarmos este valor para incremento/decremento do valor de fluxo na Figura 2, obtermos o novo fluxo apresentado na Figura 3. Note que o fluxo obtido é válido.

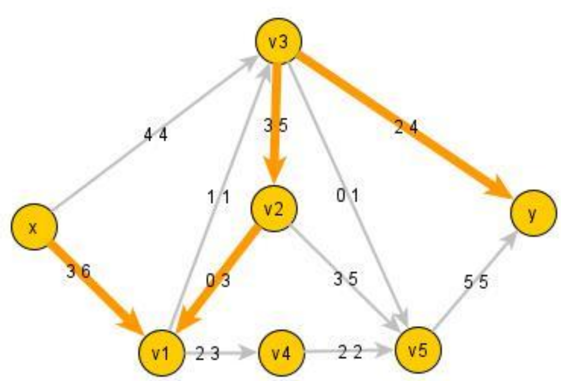


Figura 3

Algoritmo

O **algoritmo de Ford-Fulkerson** recebe como entrada uma rede de fluxo $N = N(x, y)$ e um fluxo viável f em N. Como saída, retorna um fluxo máximo f e um corte mínimo $\partial^+(X)$ em N, X é um conjunto de vértices.

O algoritmo é dividido em duas partes básicas. Na primeira (Linhas 1-5), encontramos uma árvore P no grafo base da rede onde o vértice raiz é o vértice origem x. Para compor esta árvore, escolhemos apenas arcos que ainda podem ter seu valor de fluxo incrementado ou decrementado. Na segunda parte (Linhas 6-11), se existir um caminho T de x para y na árvore P, calculamos o valor de incremento para os fluxos deste caminho e aplicamos conforme visto anteriormente. O fato deste caminho existir, significa que o fluxo atual não é máximo, visto que há um caminho de x para y onde arcos podem ainda ter seu valor ajustado a fim de obter um novo fluxo viável e válido e de maior valor. Depois retornamos ao passo 1 para encontrar novas possibilidades de incremento do fluxo. Por fim, o algoritmo retorna o valor atual do fluxo quando y não estiver incluído na árvore definida na primeira parte (ou seja, o algoritmo não executará a segunda parte).

1	$X := \{x\}; p(v) := \_, \forall v \in V$
2	**Enquanto existir*:*** um arco ***f-não-saturado*** $a := (u, v)$ ou um arco ***f-positivo*** $a := (v, u)$ onde $u \in X$ *e* $v \in V \setminus X$ **faça**
3	Substitua X por $X \cup \{v\}$
4	Substitua $p(v)$ por u
5	**fim enquanto**
6	**Se** $y \in X$ **então**
7	Compute $\varepsilon(T) := min\{\varepsilon(a) : a\ pertence\ A(T)\}$, onde T é o xy-caminho na árvore cuja função predecessor é p
8	Para cada arco avante de T, substitua $f(a)$ *por* $f(a) + \varepsilon(T)$
9	Para cada arco reverso de T, substitua $f(a)$ *por* $f(a) - \varepsilon(T)$
10	Retorne para o passo 1
11	**fim se**
12	retorne $(f, \partial^+(X))$

O algoritmo utiliza 2 variáveis:

- X, o conjunto de vértices adicionados à árvore que é inicializado com o vértice origem da rede que será a raiz da árvore (Linha 1).
- p, é a função predecessor que representa a árvore. Inicialmente o predecessor de todos os vértices da rede é indefinido (Linha 1).

Das Linhas 2 a 5, existe um ciclo repetitivo que será executado enquanto pudermos encontrar os seguintes tipos de arco (Linha 2). Seja $u \in X$ e $v \in V \setminus X$, isto é, u faz parte da árvore e v não faz parte da árvore:

- um arco (u, v) que não é *f-saturado*, ou seja, um arco avante não-saturado que ainda pode ter seu valor incrementado;
- um arco (v, u) *f-positivo*, ou seja, um arco reverso que ainda pode ter seu valor decrementado.

Escolhemos um destes arcos por vez, adicionamos o vértice v ao conjunto X e associamos o predecessor de v a u na árvore (Linhas 3 e 4).

Nas linhas 6-11, o algoritmo observa se existe um caminho T de x para y na árvore. Para tal é suficiente observar se o vértice y pertence ao conjunto de vértices da árvore (por definição toda árvore é um grafo conectado).

Em caso positivo, na Linha 7, o algoritmo calcula $\varepsilon(T)$, considerando os arcos que pertencem ao caminho T de x para y na árvore.

No Linha 8, para cada arco avante do caminho T, o valor do fluxo passa a ser o valor anterior somado a $\varepsilon(T)$. Na Linha 9, para cada arco reverso do caminho T, o valor do fluxo passa a ser o valor anterior subtraído de $\varepsilon(T)$. Na Linha 10, há um desvio para a Linha 1 a fim de encontrar novas possibilidades de incremento do fluxo.

Vejamos agora um exemplo de aplicação sobre a rede de fluxos apresentada na Figura 4 (a). Considere que, na primeira parte do algoritmo, escolhemos os seguintes arcos, nesta ordem: $(x, v1), (v2, v1), (v2, v5), (v3, v2), (v3, y), (v1, v4)$. Esta escolha pode ser visualizada na Figura 4(b). Como resultado desta escolha, as variáveis X e p do algoritmo terão os seguintes valores: $X = \{x, v1, v2, v5, v3, y, v4\}$ e $p = \{(x, v1), (v1, v2), (v2, v5), (v2, v3), (v3, y), (v1, v4)\}$. Note que a função predecessor p representa o arco $(v2, v1)$ com outra direção. Isto ocorre porque, na árvore, o predecessor de $v2$ é $v1$. O mesmo ocorre com o arco $(v3, v2)$. Ou seja, a árvore expressa uma orientação que parte de x em direção a y e é determinada de acordo com a ordem de escolha dos arcos e a inclusão dos vértices. Note que os arcos $(v2, v1)$ e $(v3, v2)$, *considerando esta árvore*, são reversos. Os demais são avante.

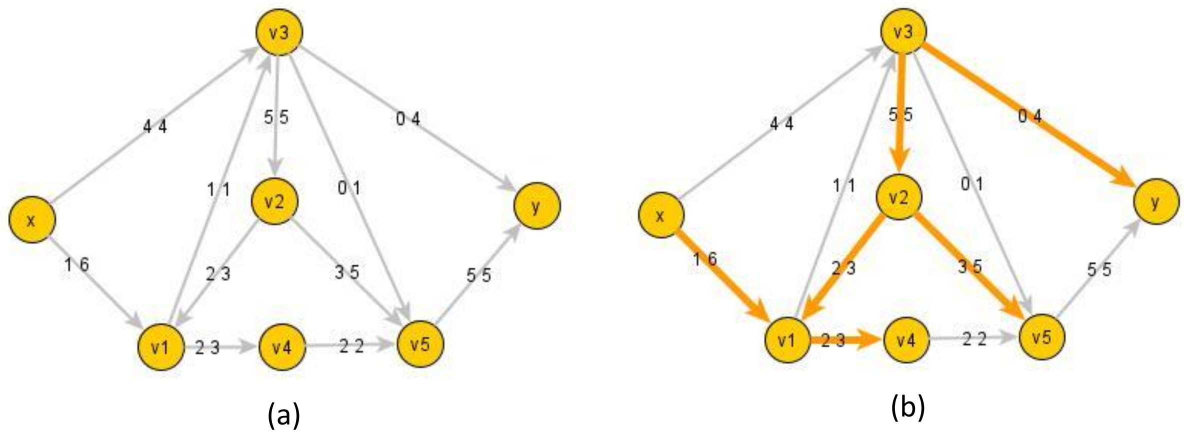
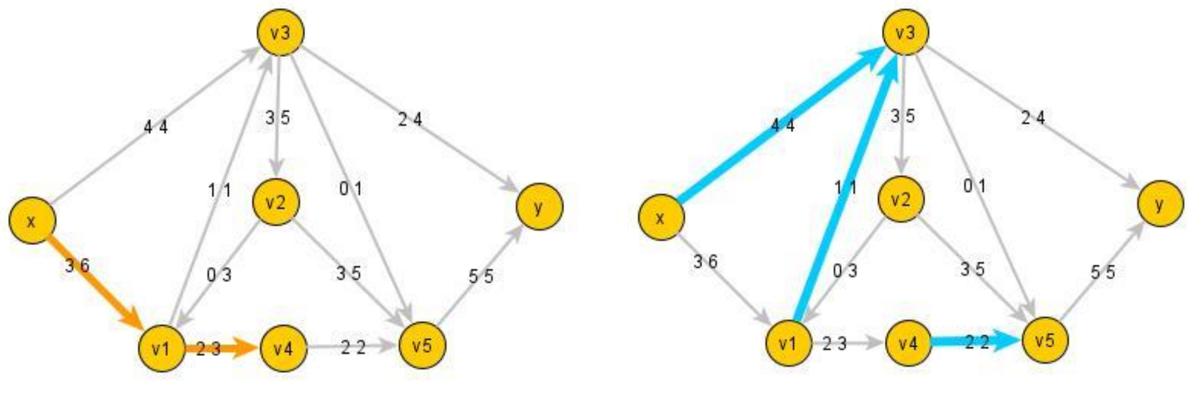


Figura 4

Como todos os vértices já foram incluídos na árvore, a execução do algoritmo procede para a segunda parte que será executada visto que $y \in X$. Para tal, o valor mínimo de incremento é calculado considerando o caminho T que vai de x para y na árvore. Este caminho está ilustrado na Figura 2 e, como já vimos anteriormente, $\varepsilon(T) = 2$. Assim, aplicando os incrementos aos arcos de T, obtemos os novos valores de fluxo ilustrados na Figura 3. Após os ajustes nos valores de fluxo, o algoritmo retorna a Linha 1 para identificar uma nova possibilidade de ajuste de valores de fluxo a fim de atingir o fluxo máximo.

Considere que nesta nova execução da primeira parte, escolhemos os seguintes arcos, nesta ordem: $(x, v1)$, $(v1, v4)$. Esta escolha está ilustrada na Figura 5 (a). Note que esta seria a única possibilidade de escolha, pois, o arco $(x, v3)$ é avante saturado, o arco $(v1, v3)$ é avante saturado e o arco $(v2, v1)$ é reverso zero. Observe também que não conseguimos escolher mais nenhum outro arco: $(v4, v5)$ é avante saturado. Não há arcos que atendam aos critérios da condição expressa na Linha 2 do algoritmo considerando u como um dos vértices $x, v1, v4$ que estão na árvore. Assim, a primeira parte encerra a sua execução com os seguintes valores para as variáveis X e p: $X = \{x, v1, v4\}$ e $p = \{(x, v1), (v1, v4)\}$.

Como y não pertence a X, a segunda parte do algoritmo não será executada e este encerra sua execução, retornando o fluxo atual como máximo e o corte mínimo calculado a partir de X. Este está representado pelos arcos em destaque na Figura 5 (b). O valor do fluxo desta rede é 7 que é exatamente a capacidade dos arcos deste corte mínimo.

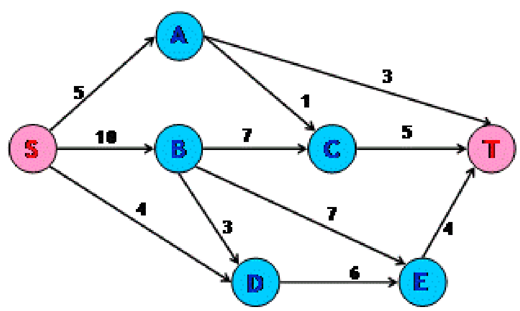


(a) (b)

Figura 5

Exercícios Propostos

1. Usando o algoritmo de Ford-Fulkerson, encontre um fluxo máximo e um corte mínimo na rede de fluxos N(S,T) abaixo. Qual o valor do fluxo máximo?

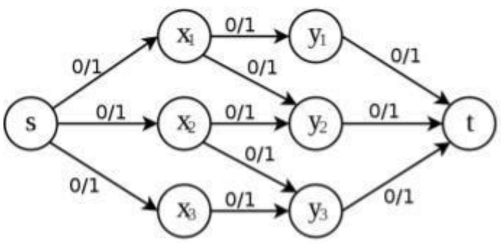


2. Para a rede abaixo, onde as capacidades dos arcos estão definidas pela função c, e a função fluxo f, responda. N(a,d) = (V(G),A(G)), V(G) = {a,b,c,d}, A(G)={(a,b),(a,c),(b,c),(c,d),(b,d)}c = {(a,b) -> 10, (a,c)->10, (b,c)->10, (c,d)->10, (b,d)->10}f = {(a,b) -> 10, (a,c)->0, (b,c)->10, (c,d)->10, (b,d)->0}: a) Aplique um passo completo do algoritmo de Ford-Fulkerson para determinar um novo valor de fluxo para a rede abaixo. Como resposta, apresente a árvore, o valor mínimo de incremento e a nova função fluxo; b) O fluxo determinado na letra a) é máximo? Justifique. c) O conjunto de vértices X determina o corte mínimo? Justifique.

3. Um agente de viagens deseja alocar congressistas em vôos de São Paulo a Porto Alegre, num mesmo dia. A disponibilidade de lugares nos vôos é dada pela tabela abaixo. Qual é o número máximo de lugares em vôos São Paulo → Porto Alegre neste dia?

S.P. - Porto Alegre	7
S.P. - Curitiba	9
Curitiba - Porto Alegre	4
S.P. - Florianópolis	8
Florianópolis - Porto Alegre	10
Curitiba - Florianópolis	3

4. Encontre um fluxo máximo e um corte mínimo para a rede abaixo usando o algortimo de Ford-Fulkerson. Qual o valor do fluxo máximo?



Referências

J. A. Bondy and U. S. R. Murty.Graph Theory. Springer, 2008,2010. (7.1 e 7.2)