CAPÍTULO V

FLUXO MÁXIMO

Um dos mais importantes problemas de otimização em redes juntamente com caminho mínimo. Seja um grafo G = (N, A), $|N|$ = n e $|A|$ = m, onde para cada arco (i, j) $\in$ A existe um limite inferior, l_{ij}, e um limite superior w_{ij} para o fluxo x_{ij}. Determinar o fluxo máximo de um nó fonte inicial f a um nó sumidouro s, final.

Esse problema é um caso particular do modelo de fluxo de custo mínimo, com um único centro de oferta, O = {f} e um único centro de demanda, D = {s}, os demais nós são de transbordo. Denominando esse fluxo de *f* a *s* por *t*, resulta:

$$\text{maximizar } t \tag{1}$$

(M1)

$$\sum_{(i,j)\in S(i)} x_{ij} - \sum_{(i,j)\in E(i)} x_{ij} = t, \quad i = f \tag{2}$$

$$\sum_{(i,j)\in S(i)} x_{ij} - \sum_{(i,j)\in E(i)} x_{ij} = 0, \quad i \in (N - \{f\} - \{s\}) \tag{3}$$

$$\sum_{(i,j)\in S(i)} x_{ij} - \sum_{(i,j)\in E(i)} x_{ij} = -t, \quad i = s \tag{4}$$

$$l_{ij} \leq x_{ij} \leq w_{ij}, \quad \forall (i,j) \in A \tag{5}$$

O modelo (M1) pode ser transformado em um modelo de circulação de fluxo, conforme mostrado anteriormente. E, uma vez que *t* é uma variável então (M1) pode também ser escrito de forma matricial por:

$$\text{maximizar } t \tag{6}$$

(M2)

$$(e_s - e_f)t + Ax = 0 \tag{7}$$

$$x \leq w \tag{8}$$

$$x \geq l \tag{9}$$

onde e_i é o vetor canônico, com a $i^{\underline{a}}$ coordenada unitária. Este é um problema linear que pode ser resolvido pelo Simplex padrão, ou adaptado para o problema de fluxos de custo de mínimo ou ainda, por um algoritmo ainda mais simplificado.

Associando variáveis duais u_i, $\forall i \in N$, as restrições (7), v_{ij}, $\forall (i,j) \in A$, as restrições (8) e h_{ij}, $\forall (i,j) \in A$, as restrições (9), resulta o problema dual:

$$\text{minimizar} \sum_{(i,j)\in A} w_{ij} v_{ij} - \sum_{(i,j)\in A} l_{ij} h_{ij} \tag{10}$$

$$u_f - u_s = 1 \tag{11}$$

$$u_i - u_j + v_{ij} - h_{ij} = 0, \quad \forall (i, j) \in A \tag{12}$$

$$v_{ij}, h_{ij} \geq 0, \quad \forall (i, j) \in A \tag{13}$$

Uma solução ótima para (M2) deverá atender a viabilidade primal e dual além da complementaridade de folga:

$$(u_i - u_j + v_{ij} - h_{ij}) x_{ij} = 0, \quad \forall (i, j) \in A \tag{14}$$

O algoritmo Simples parte de uma solução inicial e procura, a cada iteração, uma solução melhor que a atual. Para o problema em estudo, a partir de fluxos viáveis, um fluxo de f a s, superior ao dual, será obtido se for encontrado um caminho de f a s que permite o acréscimo do fluxo. À medida que esse acréscimo é efetuado, alguns arcos atingem os seus limites de capacidade, tornando-se saturados, e um novo caminho de acréscimo é procurado. Assim, quando todos os caminhos de acréscimo de f a s tiverem sido explorados o fluxo é ótimo. Um conjunto de arcos saturados separam o conjunto de nós em dois subconjuntos P e $\overline{P}$, $f \in P$ e $s \in \overline{P}$. Os arcos ligando nós de P e $\overline{P}$ compõem um corte:

CONJUNTO CORTE: Seja $P \subset N$ um conjunto de nós tal que $f \in P$ e $\overline{P} = N - P$, $s \in \overline{P}$. Então $(P, \overline{P}) \equiv \{ (i, j) \in A \mid i \in P, j \in \overline{P}$ ou $i \in \overline{P}, j \in P\}$ é um conjunto corte separando o nó f de s.

Para cada corte $(P, \overline{P})$ pode ser definida a CAPACIDADE DO CORTE: $\text{CAP}(P, \overline{P}) = \sum_{(i,j) | i\in P, j\in \overline{P}} w_{ij} - \sum_{(i,j) | i\in \overline{P}, j\in P} l_{ij}$.

EXEMPLO 1. Seja o grafo da figura (1). Entre os cortes existentes esboçar pelo menos três.

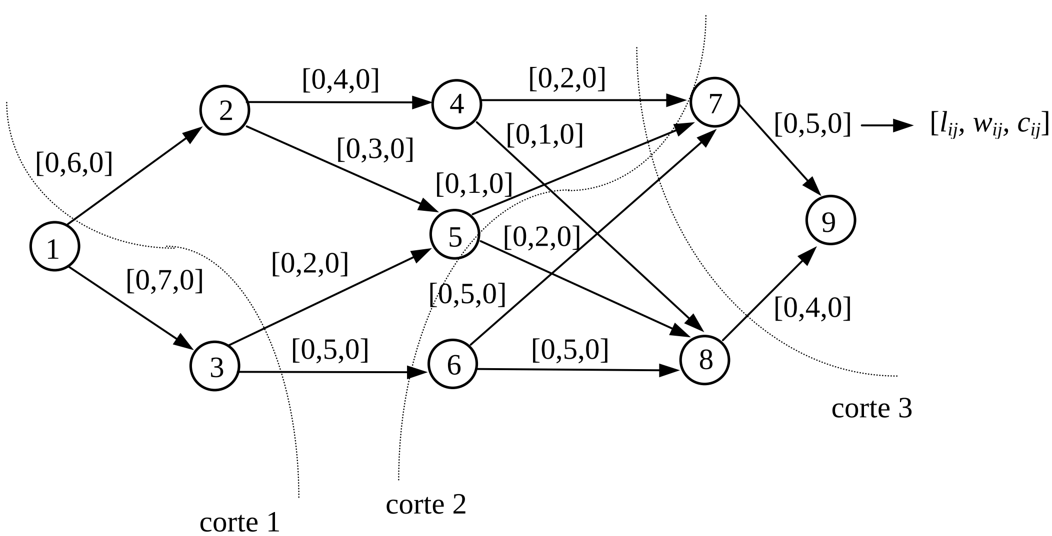


Figura 1. Grafo do exemplo (1).

CORTE 1 - $P = \{1, 3\}$, $\overline{P} = \{2, 4, 5, 6, 7, 8, 9\}$; $(P, \overline{P}) = \{(1, 2), (3, 5), (3, 6)\}$; CAP $(P, \overline{P}) = 13$

CORTE 2 - $P = \{1, 2, 3, 4, 5\}$, $\overline{P} = \{6, 7, 8, 9\}$; $(P, \overline{P}) = \{(4, 7), (4, 8), (5, 7), (5, 8), (3, 6)\}$; CAP$(P, \overline{P}) = 13$

CORTE 3 - $P = \{1, 2, 3, 4, 5, 6, 8\}$, $\overline{P} = \{7, 9\}$; $(P, \overline{P}) = \{(4, 7), (5, 7), (6, 7), (8, 9)\}$; CAP$(P, \overline{P}) = 12$

Pela teoria da dualidade pode ser escrito:

$$\text{máximo } t \leq \text{mínimo } \sum_{(i,j)\in A} w_{ij} v_{ij} - \sum_{(i,j)\in A} l_{ij} h_{ij} \qquad (15)$$

Para um certo corte $(P, \overline{P})$, fazendo $v_{ij} = 1$, $\forall(i, j) \in A | i \in P$ e $j \in \overline{P}$; e $h_{ij} = 1$, $\forall(i, j) \in A | i \in \overline{P}, i \in P$ e $v_{ij} = h_{ij} = 0$ para os demais arcos, então, de (15), o fluxo máximo t é menor ou igual à capacidade do corte $(P, \overline{P})$. Associando cada corte a uma solução dual viável, de (15) pode-se concluir que o máximo fluxo é menor ou igual à capacidade mínima entre todos os cortes. Portanto, seja o corte $(P, \overline{P})$ e a respectiva solução dual viável:

$$u_i = \begin{cases} 0, \text{ se } i \in P \\ 1, \text{ se } i \in \overline{P} \end{cases} \qquad v_{ij} = \begin{cases} 1, \text{ se } (i, j) \in A | i \in P \text{ e } j \in \overline{P} \\ 0, \text{ caso contrário} \end{cases} \qquad h_{ij} = \begin{cases} 1, \text{ se } (i, j) \in A | i \in \overline{P} \text{ e } j \in P \\ 0, \text{ caso contrário} \end{cases}$$

Então: $$\text{máximo } t \leq \text{CAP}(P, \overline{P}) \leq \underset{(P, \overline{P}) \in A}{\text{mínimo}} \text{CAP}(P, \overline{P})$$

Uma solução básica primal viável correspondente é facilmente obtida pela complementaridade das folgas. Uma base é uma árvore geradora do grafo, tal como no problema de transporte. Os arcos da base são aqueles cujos fluxos não estão nos limites de capacidade ($l_{ij} < x_{ij} < w_{ij}$), mais um subconjunto de arcos nos limites, complementando a base, com um total de (n - 1) arcos.

TEOREMA. O valor do fluxo máximo em G é igual a capacidade do corte mínimo em G.

A demonstração é conseqüência direta da aplicação da teoria da dualidade.

Na maioria dos problemas práticos os limites inferiores são nulos, $l_{ij} = 0$. Os algoritmos apresentados abaixo, partem desse pressuposto. A estrutura é a mesma para os casos em que $l_{ij} \neq 0$, a menos da solução inicial que torna-se um pouco mais complexa para ser obtida.

Com os $l_{ij} = 0$, $\forall(i, j) \in A$, uma solução inicial imediata é dada por $x_{ij} = 0$, $\forall(i, j) \in A$. A partir daí pode-se construir, a partir de G, um grafo parcial G' com os possíveis caminhos de aumento de fluxo de f a s. Para isso basta seguir a seguinte orientação:

(i) Se $x_{ij} < w_{ij}$, então $(i, j) \in G'$ com capacidade de variação de fluxo dada por $\Delta_{ij} = w_{ij} - x_{ij}$;

(ii) Se $x_{ij} > l_{ij} = 0$, então $(j, i) \in G'$ com capacidade de variação de fluxo dada por $\Delta_{ji} = x_{ij} - l_{ij} = x_{ij}$.

Se for encontrado um caminho C de f a s, com capacidade de variação do fluxo dada por $\Delta = \min \Delta_{ij}, \forall(i, j) \in C$, então atualizar os fluxos no caminho, fazendo $x_{ij} = x_{ij} + \Delta$, para todos os arcos em G, pertencentes ao caminho C em G', e com a mesma direção do caminho. E, $x_{ij} = x_{ij} - \Delta$, para todos os arcos em G, pertencentes ao caminho C em G', e com a direção contrária à do caminho. Esse processo consiste de duas etapas básicas a cada iteração:

(a) montar o grafo G', orientando-se pelos itens (i) e (ii) acima

(b) encontrar um caminho de f a s em G'.

Uma das formas de implementar essas etapas é o algoritmo de rotulação proposto por Ford e Fulkerson [1956, 1957, 1962]. A cada nó j associa um rótulo composto por duas informações: o predecessor ou sucessor de j e a capacidade de variação de fluxo até j. Na etapa (a), inicia rotulando o nó f, (f, f, ∞), e seleciona um nó rotulado k e expande, ou seja, procura rotular todos os nós adjacentes a k ainda não rotulados. Um nó j adjacente a k não rotulado, terá o seguinte rótulo: $(\pm k, \Delta j)$, onde $(+k)$ implica que k é um predecessor de j, $(-k)$ então k é um sucessor de j e Δj é a capacidade de variação do fluxo até j, $\Delta j = \min \{\Delta_k, w_{ij} - x_{ij}\}$ se $(+k)$ ou $\Delta j = \min \{\Delta_k, x_{ij} - l_{ij}\}$ se $(-k)$. Prossegue-se até rotular o nó final s. Nesse caso, executar a etapa (b). Recuperar o caminho até f atualizando os fluxos nesse caminho de Δs. Apaga todos os rótulos e inicia nova iteração. O processo continua até que não encontre um caminho de aumento de fluxo de f a s. Nesse ponto os nós rotulados e não rotulados definem um corte de capacidade mínima.

O algoritmo de rotulação apresentado a seguir foi proposto por Ford e Fulkerson [1956, 1957, 1962]. Inicia rotulando o nó f. A cada iteração seleciona um nó rotulado k e expande, ou seja, procura rotular todos os nós adjacentes a k ainda não rotulados. O processo prossegue até rotular os nós ou até que não exista caminhos de aumento de fluxo de f a s. Rotulando o nó s, recuperar o caminho até f atualizando os fluxos nesse caminho.

<u>Algoritmo</u> <u>de</u> <u>Fluxo</u> <u>Máximo</u>

Seja L^1(j) - vetor contendo para cada nó $j \in N$ o seu predecessor (+i) ou o seu sucessor (-i).

L^2(j) - vetor contendo para cada nó $j \in N$ a capacidade de variação de fluxo até j (Δ).

P - conjunto de nós, onde para cada nó foi encontrado um caminho de aumento de fluxo de f até ele e já expandido, ou seja, todos seus nós adjacentes já foram rotulados. Ao início de cada iteração $P = \emptyset$.

$\overline{P}$ - conjunto de nós, onde para cada nó foi encontrado um caminho de aumento de fluxo de f até ele.

{Solução inicial}

$x_{ij} \leftarrow 0, \forall (i, j) \in A$; $L^1(f) \leftarrow +f$; $L^2(f) \leftarrow \infty$;

{Processo iterativo}

<u>repita</u>

<u>para</u> todo $j \in (N - \{f\})$ <u>faça</u>

$L^1(j) \leftarrow +0$; $L^2(j) \leftarrow 0$;

<u>fimpara</u>;

$P \leftarrow \emptyset$; $\overline{P} \leftarrow \{f\}$;

{Pesquisar um caminho de f a s}

<u>enquanto</u> $s \notin \overline{P}$ e $\overline{P} \neq \emptyset$ <u>faça</u>

<u>escolha</u> $k \in \overline{P}$;

<u>para</u> todo $j \in N \mid \exists (k, j) \in A$ e $L^1(j) = 0$ <u>faça</u>

<u>se</u> $x_{kj} < w_{kj}$ <u>então</u> $\Delta_{kj} \leftarrow w_{kj} - x_{kj}$; $\Delta \leftarrow \min \{\Delta_{kj}, L^2(k)\}$

$L^1(j) \leftarrow +k$; $L^2(j) \leftarrow \Delta$; $\overline{P} \leftarrow \overline{P} \cup \{j\}$;

<u>fimse</u>;

<u>fimpara</u>;

para todo $j \in N \mid \exists (j, kj) \in A$ e $L^1(j) = 0$ faça

se $x_{jk} > l_{jk}$ então $\Delta_{jk} \leftarrow x_{jk} - l_{jk}$; $\Delta \leftarrow \min\{\Delta_{jk}, L^2(k)\}$

$L^1(j) \leftarrow -k$; $L^2(j) \leftarrow \Delta$; $\overline{P} \leftarrow \overline{P} \cup \{j\}$;

fimse;

fimpara;

$P \leftarrow P \cup \{k\}$; $\overline{P} \leftarrow \overline{P} - \{k\}$;

fimenquanto;

se $\overline{P} \neq \varnothing$ então

{Recuperar o caminho de aumento de f a s}

$j \leftarrow s$; $\Delta = L^2(j)$;

enquanto $j \neq f$ faça

$i \leftarrow | L^1(j)|$;

se $L^1(j) < 0$ então $\quad x_{ij} \leftarrow x_{ij} - \Delta$;

senão $\quad x_{ij} \leftarrow x_{ij} + \Delta$;

fimse;

$j \leftarrow i$;

fimenquanto;

fimse;

até $\overline{P} = \varnothing$;

fimalgoritmo.

Ao término do algoritmo em P estará todos os nós tais que existe um caminho de aumento de f até eles. Os demais nós não foram rotulados e não existe um caminho de aumento de f a s. Fazendo $\overline{P} = N - P$, resulta em G em corte $(P, \overline{P})$ cuja capacidade é igual ao fluxo máximo obtido. Este é portanto o corte mínimo e todos os arcos do corte $(P, \overline{P})$ limitam o crescimento do fluxo. Os arcos (i, j) onde $i \in P$ e $j \in \overline{P}$, estão no seu limite superior $x_{ij} = w_{ij}$, e os arcos (i, j) onde $i \in \overline{P}$ e $j \in P$ estão no seu limite inferior $x_{ij} = l_{ij}$.

EXEMPLO 2. Seja o grafo da figura (2). Determinar o fluxo máximo do nó inicial 1 ao nó final 9.

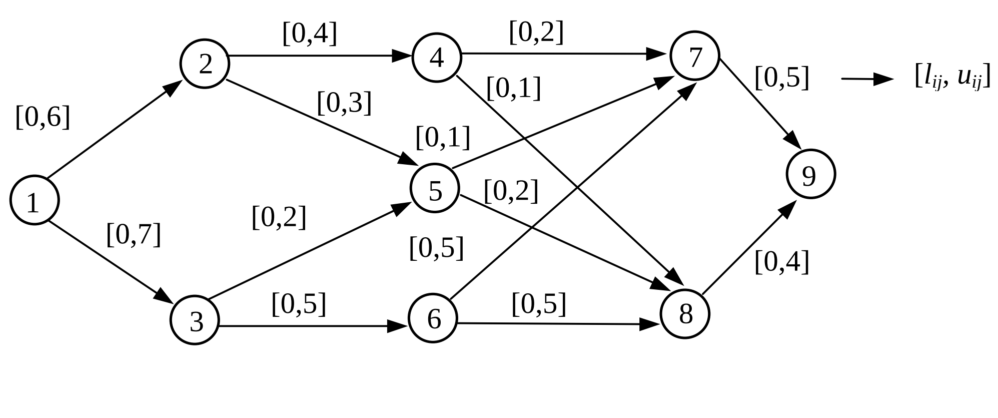


Figura 2. Grafo do exemplo (2).

Aplicando o algoritmo de fluxo máximo resulta:

$x_{ij} = 0, \ \forall\, (i, j) \in A$

$L^1(1) = +1, L^2(1) = \infty; L^1(2) = ... = L^1(9) = 0; L^2(2) = ... = L^2(9) = 0$

$P = \varnothing; \ \overline{P} = \{1\}$

$k = 1, j = 2 \Rightarrow L^1(2) = +1, L^2(2) = 6, \ \overline{P} = \{1,2\}$

$j = 3 \Rightarrow L^1(3) = +1, L^2(3) = 7, \ \overline{P} = \{1,2,3\}$

$P = \{1\}; \ \overline{P} = \{2,3\}$

$k = 2, j = 4 \Rightarrow L^1(4) = +2, L^2(4) = 4, \ \overline{P} = \{2,3,4\}$

$j = 5 \Rightarrow L^1(5) = +2, L^2(5) = 3, \ \overline{P} = \{2,3,4,5\}$

$P = \{1,2\}; \ \overline{P} = \{3,4,5\}$

$k = 3, j = 6 \Rightarrow L^1(6) = +3, L^2(6) = 5, \ \overline{P} = \{3,4,5,6\}$

$P = \{1,2,3\}; \ \overline{P} = \{4,5,6\}$

$k = 4, j = 7 \Rightarrow L^1(7) = +4, L^2(7) = 2, \ \overline{P} = \{4,5,6,7\}$

$j = 8 \Rightarrow L^1(8) = +4, L^2(8) = 1, \ \overline{P} = \{4,5,6,7,8\}$

$P = \{1,2,3,4\}; \ \overline{P} = \{5,6,7,8\}$

$k = 7, j = 9 \Rightarrow L^1(9) = +7, L^2(9) = 2, \ \overline{P} = \{5,6,7,8,9\}$

$P = \{1,2,3,4,7\}; \ \overline{P} = \{5,6,8\}$

$(s = 9) \in \overline{P}$ e $P = \varnothing$

$\Delta = 2, j = 9, i = 7, L^1(9) > 0 \Rightarrow x_{79} = 0 + 2 = 2$

$j = 7, i = 4, L^1(7) > 0 \Rightarrow x_{47} = 0 + 2 = 2$

$j = 4, i = 2, L^1(4) > 0 \Rightarrow x_{24} = 0 + 2 = 2$

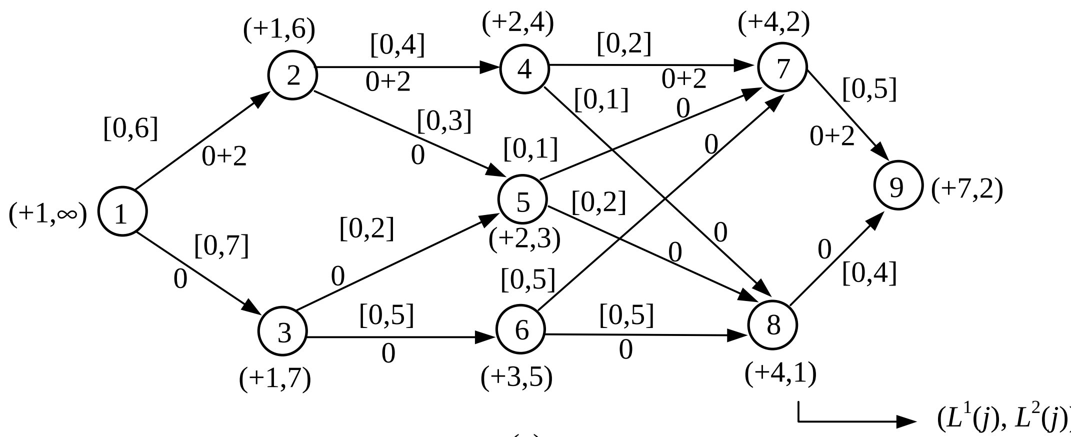
$j = 2, i = 1, L^1(2) > 0 \Rightarrow x_{12} = 0 + 2 = 2$

$L^1(2) = ... = L^1(9) = 0; L^2(2) = ... = L^2(9) = 0$

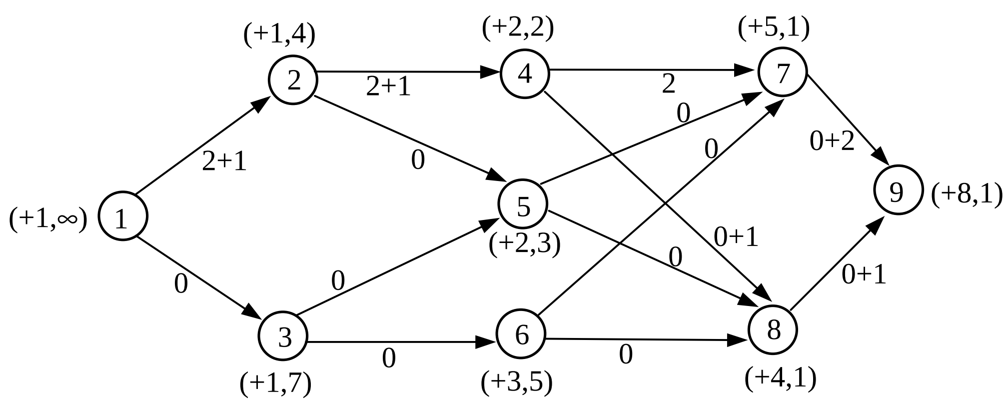
$P = \varnothing; \ \overline{P} = \{1\}$

$k = 1, j = 2 \Rightarrow L^1(2) = +1, L^2(2) = 4, \ \overline{P} = \{1,2\}$

$j = 3 \Rightarrow L^1(3) = +1, L^2(3) = 7, \ \overline{P} = \{1,2,3\}$

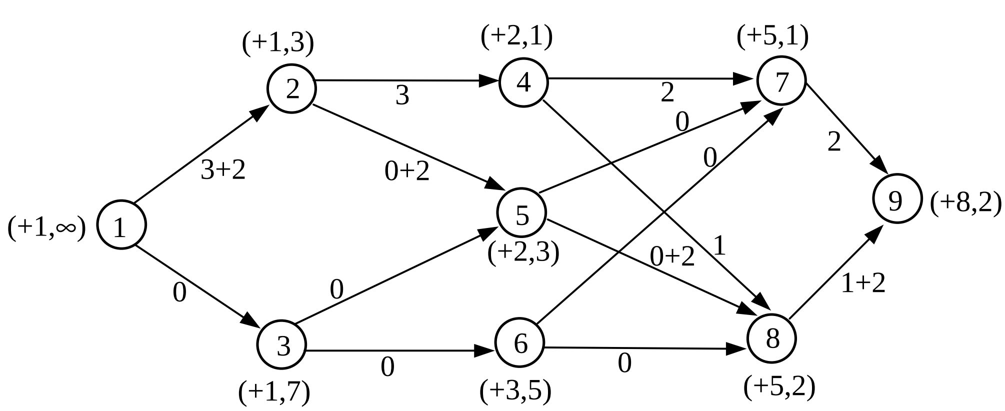


(a)



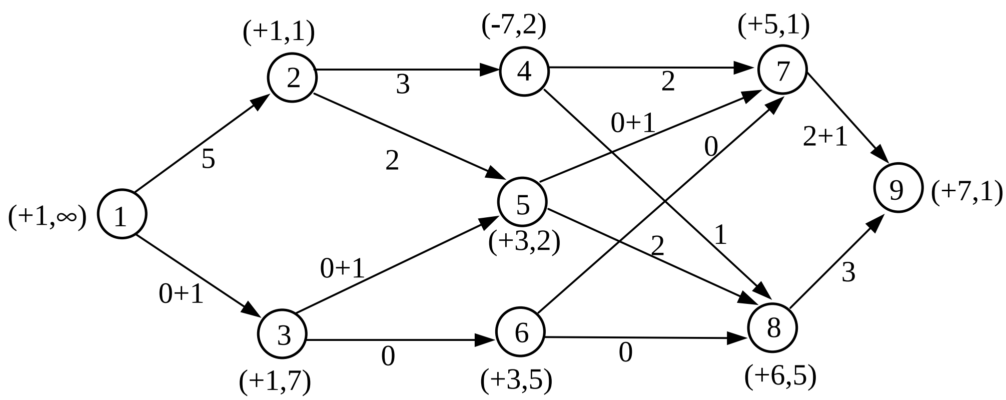
k = 2, k = 3, k = 4, k = 5, k = 8 $\Rightarrow \Delta = 1$

(b)



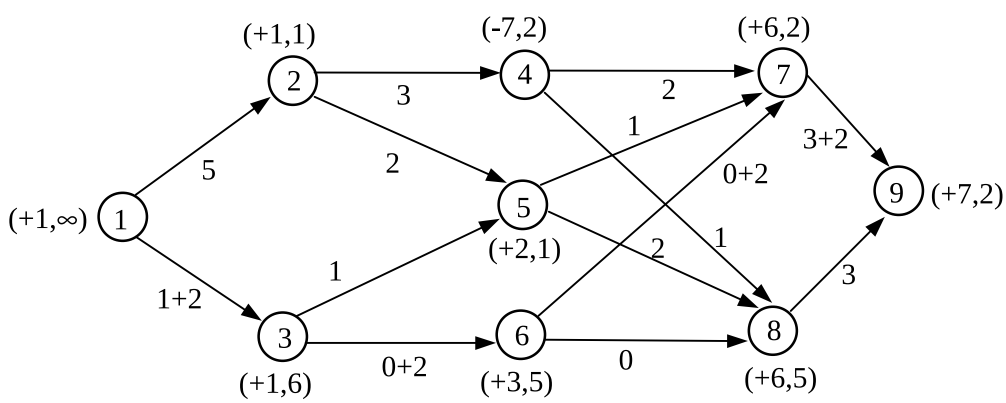
k = 1, k = 2, k = 3, k = 5, k = 8 $\Rightarrow \Delta = 2$

(c)



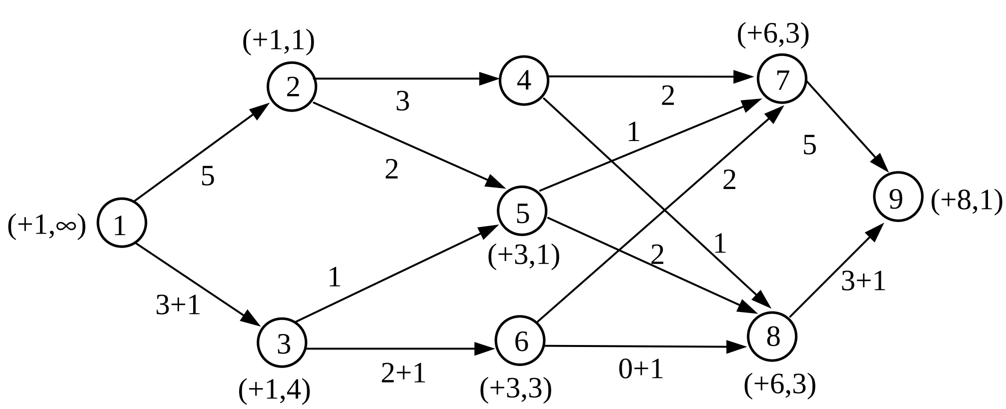
k = 1, k = 3, k = 5, k = 7 ⇒ Δ = 1

(d)

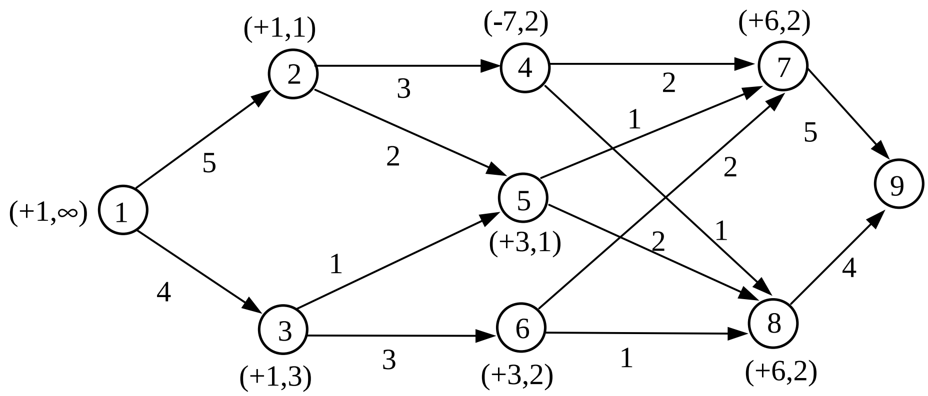


k = 1, k = 2, k = 3, k = 6, k = 7 ⇒ Δ = 2

(e)



k = 1, k = 3, k = 6, k = 8 ⇒ Δ = 1



k = 1, k = 3, k = 5, k = 6, k = 7, k = 2, k = 8, k = 4 $\Rightarrow$
P = {1,2,3,4,5,6,7,8}, $\overline{P} = \emptyset$

(g)

Após as iterações (a), (b), (c), (d), (e) e (f), a última iteração (g) pode ser assim detalhada:

$L^1(2) = L^1(3) = ... = L^1(9) = 0$; $L^2(2) = L^2(3) = ... = L^2(9) = 0$

$P = \emptyset$; $\overline{P}$ = {1}

k = 1, j = 2 $\Rightarrow$ $L^1(2) = +1$, $L^2(2) = 1$, $\overline{P}$ = {1,2}

j = 3 $\Rightarrow$ $L^1(3) = +1$, $L^2(3) = 3$, $\overline{P}$ = {1,2,3}

P = {1}; $\overline{P}$ = {2,3}

k = 3, j = 5 $\Rightarrow$ $L^1(5) = +3$, $L^2(5) = 1$, $\overline{P}$ = {2,3,5}

j = 6 $\Rightarrow$ $L^1(6) = +3$, $L^2(6) = 2$, $\overline{P}$ = {2,3,5,6}

P = {1,3}; $\overline{P}$ = {2,5,6}

k = 5

P = {1,3,5}; $\overline{P}$ = {2,6}

k = 6, j = 7 $\Rightarrow$ $L^1(7) = +6$, $L^2(7) = 2$, $\overline{P}$ = {2,6,7}

j = 8 $\Rightarrow$ $L^1(8) = +6$, $L^2(8) = 2$, $\overline{P}$ = {2,6,7,8}

P = {1,3,5,6}; $\overline{P}$ = {2,7,8}

k = 7, j = 4 $\Rightarrow$ $L^1(4) = -7$, $L^2(4) = 2$, $\overline{P}$ = {2,7,8,4}

P = {1,3,5,6,7}; $\overline{P}$ = {2,8,4}

k = 2

P = {1,3,5,6,7}; $\overline{P}$ = {8,4}

$k = 8$

$P = \{1,3,5,6,7,8\}$; $\overline{P} = \{4\}$

$k = 4$

$P = \{1,3,5,6,7,8,4\}$; $\overline{P} = \emptyset$

$P = \emptyset \Rightarrow$ Fim

Ao aplicar o algoritmo de fluxo máximo, a escolha do nó, $k \in \overline{P}$, rotulado, a ser expandido, a escolha é aleatória. Pode-se, porém, estabelecer algum critério. Se o nó escolhido for aquele rotulado mais antigo tem-se uma busca horizontal ou em largura. Se for escolhido aquele rotulado mais recente tem-se uma busca em profundidade. No primeiro caso a tendência é pesquisar todos os nós do grafo por camadas (iteração (a) do exemplo (2)). No segundo, a rotulação se aprofunda em direção ao nó final (iteração (f) do exemplo (2)). Dependendo das circunstâncias uma opção é melhor que outra.

Outros critérios podem ser definidos para escolha do nó a ser expandido. Um caso específico é o problema de mínimo custo - máximo fluxo. Determinar o fluxo máximo de mínimo custo. O nó escolhido recai sobre aquele cujo caminho de *f* a ele é mínimo entre todos os nós passíveis de serem selecionados.

A escolha pode alterar significativamente o desempenho do algoritmo. Por esse aspecto o algoritmo apresentado é de complexidade exponencial. Trabalhando nas estruturas de dados a complexidade pode ser sensivelmente reduzida. Assim, se a escolha dos nós implica em um caminho de aumento de fluxo com o menor número de arcos entre todos os caminhos, então o algoritmo torna-se polinomial e complexidade $O(nm^2)$. É o caso do algoritmo proposto por Edmonds e Karp [1972]. Em ordem decrescente de complexidade surgem os algoritmos de Dinic [1970], $O(n^2m)$, Malhotra, Pramodh Kumar e Maheshwari [1978], também conhecido por método dos três indianos e Karzanov [1974], ambos $O(n^3)$. Mais recentemente surgiu o algoritmo de Sleator [1980], $O(nm \log n)$.