

## ALGORITMO PARA O PROBLEMA DE FLUXO MÁXIMO MULTITERMINAL

**João Paulo de Freitas Araujo**

Departamento de Engenharia Industrial / PUC-Rio  
Rua Marquês de São Vicente, 225, Gávea - Rio de Janeiro, RJ - Brasil - 22451-900  
[johnpa@terra.com.br](mailto:johnpa@terra.com.br)

**Fernanda Maria Pereira Raupp**

Departamento de Engenharia Industrial / PUC-Rio  
Rua Marquês de São Vicente, 225, Gávea - Rio de Janeiro, RJ - Brasil - 22451-900  
[fraupp@puc-rio.br](mailto:fraupp@puc-rio.br)

**Madiagne Diallo**

Departamento de Engenharia Industrial / PUC-Rio  
Rua Marquês de São Vicente, 225, Gávea - Rio de Janeiro, RJ - Brasil - 22451-900  
[diallo@puc-rio.br](mailto:diallo@puc-rio.br)

### RESUMO

Neste trabalho é apresentado um algoritmo que resolve o problema de fluxo máximo multiterminal. O método proposto se baseia na teoria recente sobre análise de sensibilidade, que estuda a influência da variação da capacidade de uma aresta sobre os fluxos máximos multiterminais. Técnicas dos métodos tradicionais que abordam o problema original, tais como contração de nós e construção de árvores de corte, também fazem parte do método proposto. Por fim, é apresentada passo a passo a aplicação do algoritmo proposto em uma instância do problema original, assim como uma variante do algoritmo proposto.

**PALAVRAS-CHAVE.** Fluxos em redes multiterminais, Análise de sensibilidade, Árvore de cortes.

**TAG, TEL&SI.**

### ABSTRACT

In this work, we present an algorithm that solves the problem of multiterminal maximum flow. The proposed method is based on the recent theory on sensitivity analysis, which examines the influence of the capacity variation of an edge on the maxima multiterminal flows. Techniques of traditional methods that address the original problem, such as contraction of nodes and construction of cut trees, are also part of the proposed method. Finally, it is presented step by step the application of the proposed algorithm on an instance of the original problem, as well as a variant of the proposed algorithm.

**KEYWORDS.** Multi-terminal network flows, Sensitivity analysis, Cut trees.

**TAG, TEL&SI.**

## 1. Introdução

Derivado do famoso problema de encontrar o fluxo máximo entre um nó origem e um nó destino de uma rede, o *problema do fluxo máximo multiterminal* consiste em encontrar o fluxo máximo entre todos os pares de nós da rede. Este problema surge no contexto de fluxos em redes, tema que possui diversas aplicações, especialmente nos campos de transporte, telecomunicações e energia. Mais especificamente, problemas de fluxos em redes abordam, por exemplo, o problema de alocação, com a produção de fornecedores suprindo demandas de clientes, o problema de custo mínimo, que objetiva reduzir custos de operações, e o problema de fluxo máximo, que busca obter o maior valor do fluxo de operações. Mais exemplos podem ser encontrados em Agarwal e Arora (1976), Bako (1974) e Diallo (2011).

É importante ressaltar que o problema de fluxo multiterminal é diferente do problema de multifixos, ou multiproduto. Nestes, os fluxos entre várias origens e vários destinos são calculados simultaneamente na rede, enquanto naquele, apesar de o fluxo máximo entre todos os pares de nós ser determinado, são considerados, a cada momento, uma única origem e um único destino na rede.

Na década de 50, Ford e Fulkerson (1973) popularizaram o problema do fluxo máximo com seu método de resolução. Eles, especialmente, demonstraram a relação entre o fluxo máximo e o corte mínimo, uma extensão do Teorema de Menger. Assim, o problema multiterminal, por sua vez, estava resolvido, bastando aplicar o algoritmo  $n(n-1)/2$  vezes para determinar o fluxo máximo entre todos os pares de nós de uma rede simétrica com  $n$  nós.

Alguns anos mais tarde, Gomory e Hu (1961) desenvolveram um método capaz de resolver o problema de fluxo máximo multiterminal executando apenas  $n-1$  vezes o algoritmo de fluxo máximo. O resultado do algoritmo é exposto através de uma árvore de cortes, que reflete todos os fluxos máximos. Em 1990, Gusfield (1990) apresenta um procedimento mais simples de obter a mesma árvore de cortes, mas que também utiliza  $n-1$  algoritmos de fluxo máximo.

A teoria de análise de sensibilidade em fluxos multiterminais teve seu início na década de 60 com Elmaghraby (1964). Ele estudou os efeitos nos fluxos máximos de uma rede sob a variação da capacidade de uma única aresta (paramétrica) da rede. Ele observou que, para determinados valores de capacidade da aresta paramétrica, alguns fluxos máximos se alteravam. Ele nomeou estes valores de *capacidades críticas*, e transferiu a análise de sensibilidade de fluxos multiterminais para o problema de determinar todas as capacidades críticas. Por fim, concluiu que era necessário calcular uma árvore de cortes para obter cada capacidade crítica.

Recentemente, Berthomé *et al.* (2003) e Diallo (2003) demonstraram que, com apenas duas árvores de cortes, é possível determinar todas as capacidades críticas. Eles expandiram este resultado para o caso de uma rede com  $k$  arestas paramétricas, observando que  $2^k$  árvores de cortes são suficientes para calcular todos os fluxos máximos para quaisquer valores dos parâmetros.

Com o objetivo de reduzir a complexidade dos algoritmos que necessitam calcular duas ou mais árvores de cortes em seqüência, para o caso de uma única aresta paramétrica na rede, Barth *et al.* (2006) mostraram como utilizar as informações de uma árvore de cortes já calculada para construir a seguinte.

Neste trabalho, uma extensão do resultado de Barth *et al.* (2006) é apresentada, abordando não somente o caso de uma única aresta paramétrica na rede, mas, também, o caso de múltiplas arestas paramétricas.

Utilizar as informações de uma árvore de cortes já calculada para computar a próxima árvore é a idéia central do algoritmo proposto neste trabalho. Com o auxílio das técnicas de contração de Gomory e Hu e do procedimento algorítmico de Gusfield de construção de árvores de cortes, o método proposto busca construir uma árvore de cortes de uma rede através da construção de uma seqüência de árvores intermediárias, procurando sempre aplicar o algoritmo de fluxo máximo em redes consideravelmente menores do que a original.

Este trabalho está dividido em 5 seções, sendo esta introdução a primeira delas. A seção 2 apresenta a notação aqui usada, assim como conceitos básicos que se referem ao problema aqui

abordado. Na seção 3, são apresentados os métodos de Gomory e Hu (1961) e de Gusfield (1990), que serão usados pelo algoritmo proposto. Na Seção 4 é apresentada a teoria da análise de sensibilidade em fluxos multiterminal, a qual servirá de base para a construção do algoritmo proposto neste trabalho. A extensão para o caso paramétrico é brevemente abordada. A seção 5 apresenta, em detalhes, o algoritmo tema deste trabalho. Um exemplo é utilizado para ilustrar com maior clareza seu funcionamento. Na Seção 6 encontram-se os comentários finais seguidos por sugestões de trabalhos futuros.

## 2. Notação e Conceitos básicos

Assumiremos aqui que o leitor tenha conhecimentos básicos sobre a teoria dos grafos e de fluxos, assim como sobre a formulação do problema de fluxo máximo. Para mais informações sobre grafos e fluxos, ver as referências Ford e Fulkerson (1973), Hu e Shing (2002) e Araujo (2011).

Seja  $G = (V, E)$  um grafo conexo não direcionado, consistindo de um conjunto  $V$  de vértices  $v$ , também chamados de nós, e um conjunto  $E$  de arestas  $e$ , onde cada aresta é um par não ordenado  $(i, j)$  de vértices,  $i, j \in V$ . Uma *rede* é um grafo  $G$  associado a uma função *capacidade* nas arestas  $c: E \rightarrow R^+$ . Um fluxo de um vértice origem (*source*)  $s$  a um vértice destino (*terminal*)  $t$  em  $G$  é uma função  $f: E \rightarrow R^+$  com a propriedade de conservação do fluxo em cada vértice, exceto para  $s$  e  $t$ , ou seja,  $\forall v \in V \setminus \{s, t\}, \sum_{i \in V} f(i, v) = \sum_{j \in V} f(v, j)$ , e a propriedade de restrição de capacidade, ou seja,  $\forall i, j \in V, f(i, j) \leq c(i, j)$ .

Cabe notar que arestas podem ser representadas por dois arcos de sentidos opostos. Portanto, ao contrário dos arcos, as arestas não possuem sentido, ou igualmente falando, possuem duplo sentido. Dessa forma, uma rede não direcionada possui a mesma estrutura de uma rede direcionada simétrica, onde cada arco possui a mesma capacidade da aresta original, possibilitando assim capacidade igual para qualquer sentido.

Como nomenclatura, neste trabalho, a letra  $n$  representará o número de nós em um grafo, e a letra  $m$  o número de arestas de um grafo. Denotaremos por  $(s - t)$  o corte que separa os nós  $s$  e  $t$ , por  $c(s - t)$  a capacidade do corte, e por  $(X, \bar{X})$  um corte que separa os nós de um grafo em dois subconjuntos complementares, sejam  $X$  e  $\bar{X}$ . Dentre todos os possíveis cortes que separam  $s$  e  $t$ , aquele com a menor capacidade é chamado de corte mínimo. O fluxo máximo entre um par de nós  $s$  e  $t$  será representado por  $f_{s,t}$ .

## 3. Procedimentos básicos

Nesta seção serão detalhados os procedimentos de Gomory e Hu (1961) e de Gusfield (1990) que serão utilizados no algoritmo aqui proposto. É importante notar que esses algoritmos resolvem o problema de fluxo máximo multiterminal em redes não direcionadas. Para o leitor interessado no estudo do caso de redes direcionadas, ver Scutellà (2003).

### 3.1. Método de Gomory e Hu

Após a observação da existência de no máximo  $n - 1$  valores distintos de fluxo máximo em uma rede, Gomory e Hu desenvolveram um método capaz de obter os  $n(n - 1)/2$  valores de fluxo máximo por meio de contrações e de  $n - 1$  execuções do algoritmo de fluxo máximo, cujo resultado final é expresso pela árvore de cortes (definição 1).

**Definição 1.** Uma *árvore de cortes* (*cut tree*) de uma rede  $G = (V, E)$  é uma árvore  $CT = (V, E')$  obtida de  $G$ , com capacidades nas arestas e com os mesmos vértices de  $G$ .  $CT$  tem as seguintes propriedades:

1. Árvore de fluxo equivalente: o valor do fluxo máximo entre qualquer  $s$  e  $t$  de  $G$  é igual ao valor do fluxo máximo em  $CT$  entre  $s$  e  $t$ , isto é, a menor das capacidades das arestas do

único caminho de  $s$  para  $t$  em  $CT$ ; deste modo, os fluxos máximos entre todos os pares de vértices de  $G$  estão representados em  $CT$ ;

2. **Propriedade de corte:** a remoção de qualquer aresta  $e$  de capacidade  $c(e)$  em  $CT$  separa seus vértices em dois grupos. Esta partição corresponde a um corte em  $G$  com capacidade  $c(e)$ .

A Figura 1 ilustra um exemplo de árvore de cortes  $CT$  construída a partir de uma rede  $G$ . Pode-se observar, pelas propriedades mencionadas, que o corte mínimo entre os nós 2 e 3 e os nós 1 e 2 são respectivamente as arestas (2, 3) e (1, 2) da árvore de cortes  $CT$ . Seus fluxos máximos são as capacidades destas arestas, no caso, 3 e 4. O corte mínimo entre os nós 1 e 3 é por sua vez a aresta (2, 3), com fluxo máximo igual a 3.

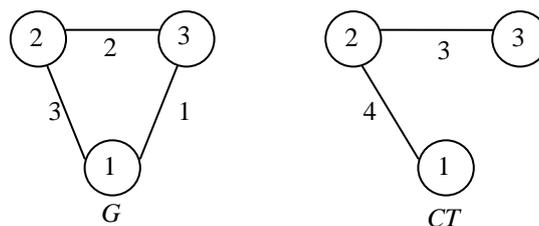


Figura 1: Exemplo de uma árvore de cortes  $CT$  de uma rede  $G$ .

O algoritmo criado por Gomory e Hu induz, por meio de processos de contrações de vértices, a formação de cortes mínimos que não se intersectam. Desta forma, eles mostraram que o fluxo máximo calculado entre dois nós não contraídos em uma rede contraída tem igual valor ao calculado na rede original. O corte mínimo encontrado na rede contraída também é corte mínimo na rede original, bastando substituir os super nós pelos nós que os compõem. Nesse algoritmo, a árvore de cortes final é alcançada quando todos os super nós possuem apenas um nó cada, e isto acontece após  $n - 1$  resoluções do problema de fluxo máximo. A Figura 2 apresenta o pseudocódigo do algoritmo de Gomory e Hu (GH).

Em geral, é possível encontrar várias árvores de cortes para uma mesma rede. A árvore de cortes apenas será única se todos os cortes ( $s$ - $t$ ) da rede forem únicos.

**procedimento GH ( $G$ );**

- 1 Calcule corte mínimo  $(X, \bar{X})$  entre dois nós arbitrários da rede  $G$ ;
  - 2 Construa árvore  $CT$  contraindo  $X$  e  $\bar{X}$  em dois super nós e os ligando através de uma aresta de capacidade  $c(X, \bar{X})$ ;
  - 3 **enquanto** existir super nó com mais de um nó **faça**
  - 4     Selecione arbitrariamente dois nós  $s$  e  $t$  de um super nó  $\bar{v}$ ;
  - 5     Contraia, em  $G$ , cada componente conexo de  $CT \setminus \bar{v}$  em um super nó, gerando uma rede contraída;
  - 6     Calcule corte mínimo  $(X_1, \bar{X}_1)$  entre  $s$  e  $t$  na rede contraída onde  $s \in X_1$  e  $t \in \bar{X}_1$ ;
  - 7     Conecte os super nós  $X_1$  e  $\bar{X}_1$  com uma aresta de capacidade  $c(X_1, \bar{X}_1)$ ;
  - 8     **se**  $\bar{X}$  estiver do mesmo lado do corte que  $s$
  - 9         Conecte  $\bar{X}$  à  $X_1$ ;
  - 10    **se não**
  - 11         Conecte  $\bar{X}$  à  $\bar{X}_1$ ;
  - 12    **finalize se**
  - 13 **finalize enquanto**
  - 14 **retorne**  $CT$ ;
- finalize** GH;

Figura 2: Pseudocódigo do algoritmo de Gomory e Hu.

### 3.2. Método de Gusfield

Gusfield desenvolveu um método muito simples para construir uma árvore de cortes. Assim como o método de Gomory e Hu, o método de Gusfield resolve o problema de fluxo máximo multiterminal com  $n - 1$  execuções do algoritmo de fluxo máximo, porém sua implementação é muito fácil: bastam apenas cinco linhas de código adicionadas a qualquer algoritmo que calcule corte mínimo. A Figura 3 apresenta o pseudocódigo do algoritmo de Gusfield (Gus).

Gusfield demonstrou que os procedimentos básicos do algoritmo de Gomory e Hu, isto é, a formação de cortes mínimos que não se intersectam, e a contração de vértices, não são necessários para a construção de uma árvore de cortes.

**procedimento** Gus ( $G$ );

```

1  Gere a árvore estrela  $T$  onde o vértice 1 é a raiz e os outros
   vértices de 2 a  $n$  são as folhas ligadas a 1;
2  para  $s = 2$  a  $n$  faça
3     Selecione  $t$  como sendo o (único) vizinho de  $s$  em  $T$ ;
4     Calcule o corte mínimo  $(X, \bar{X})$  entre  $s$  e  $t$  na rede original  $G$ ;
5     Marque a aresta  $(s, t) \in T$  com a capacidade do corte mínimo  $(X, \bar{X})$ ;
6     para todo vértice  $i \neq s$  faça
7         se  $i$  é vizinho de  $t$  e está do mesmo lado do corte  $(X, \bar{X})$  que  $s$  faça
8             Desconecte  $i$  de  $t$ ;
9             Conecte  $i$  a  $s$ ;
10            Marque a nova aresta  $(i, s)$  com a capacidade da antiga aresta  $(i, t)$ ;
11        finalize se
12    finalize para
13 finalize para
14 retorne  $T$ ;
finalize Gus;
```

Figura 3: Pseudocódigo do algoritmo de Gusfield.

### 4. Análise de Sensibilidade de fluxos máximos

Nesta seção será apresentada a versão paramétrica do *problema de fluxo máximo multiterminal* aqui chamada simplesmente de problema paramétrico, a qual é fundamentada na teoria de *análise de sensibilidade de fluxos máximos multiterminal*. Em termos gerais, esta teoria estuda o comportamento do fluxo máximo entre todos os pares de nós de uma rede frente à variação de capacidade de uma aresta da rede.

Pioneiro no desenvolvimento de um método de resolução do problema paramétrico, Elmaghraby (1964) analisou a influência nos fluxos máximos entre todos os pares de nós sob a condição de decréscimo linear da capacidade de uma aresta da rede. Ele lançou o conceito de *capacidade crítica* ao observar que, o valor do fluxo máximo entre um par de vértices, na maioria dos casos, frente a este decréscimo, se comporta da seguinte maneira:

- temporariamente sensível à variação de  $c(e)$ , ou seja, o valor do fluxo máximo permanece insensível até um valor crítico  $c^*$  de  $c(e)$ , quando então começa a decrescer proporcionalmente com o valor  $c(e)$ , indicando que, a partir de  $c^*$ , a aresta  $e$  está contida nos cortes mínimos que separam o par de vértices.

Portanto, para uma variação decrescente, a capacidade crítica  $c^*$  indica o valor da capacidade da aresta paramétrica a partir do qual a aresta passa a pertencer a um corte mínimo que separa os vértices.

#### 4.1. Variação crescente

Berthomé *et al.* (2003) e Diallo (2003) apresentaram um método de resolução mais eficiente para o problema paramétrico. Ao contrário de Elmaghraby, eles analisaram o problema paramétrico considerando a variação crescente da capacidade de uma aresta. O problema pode ser formulado assim:

Seja  $G = (V, E)$  uma rede com origem no vértice  $s$  e destino no vértice  $t$ . Seja uma aresta única  $e = (i, j) \in E$  com capacidade  $c(e) = \lambda$ ,  $\lambda \geq 0$ . Deseja-se determinar o fluxo máximo entre  $s$  e  $t$  para todos os valores crescentes do parâmetro  $\lambda$ .

Denote por  $f_{s,t}(\lambda)$  o valor do fluxo máximo entre  $s$  e  $t$  quando a capacidade da aresta  $e$  é  $\lambda$ . Ainda, denote por  $f_{s,t}^0$  (ou simplesmente  $f^0$ ) o valor do fluxo máximo  $f_{s,t}(0)$ , ou seja, quando a aresta  $e$  é removida da rede, e, por  $f_{s,t}^\infty$  (ou simplesmente  $f^\infty$ ) o  $\lim_{\lambda \rightarrow \infty} f_{s,t}(\lambda)$ , ou seja, o valor do fluxo máximo quando não há restrição de capacidade em  $e$ . Este valor é finito para todos os pares  $\{s, t\}$ , exceto quando  $s = i$  e  $t = j$ . Observe que calcular  $f_{s,t}^\infty$  pode ser facilmente executado marcando a aresta  $e$  com a soma das capacidades de suas arestas adjacentes.

O valor do fluxo máximo  $f_{s,t}(\lambda)$ , para qualquer par  $s, t$ , frente à variação crescente de  $\lambda$ , pode ser:

- Sensível à variação de  $c(e)$ . Este caso só ocorre quando  $s = i$  e  $t = j$ , pois, por definição, a aresta  $e$  sempre pertencerá a um corte mínimo que separa o par de vértices, e, desta forma, o valor do fluxo máximo crescerá proporcionalmente com o valor de  $\lambda$ . Logo  $f_{i,j} \rightarrow \infty$  quando  $\lambda \rightarrow \infty$ ;
- Temporariamente sensível à variação de  $c(e)$ , ou seja, o valor do fluxo máximo cresce proporcionalmente com o valor de  $\lambda$  até o valor crítico  $\lambda^*$  de  $c(e)$ . Neste estágio, a aresta  $e$  está contida em um corte mínimo que separa  $s$  e  $t$ . A partir de  $\lambda^*$ , o fluxo máximo permanece insensível ao aumento de  $\lambda$ , indicando que a aresta  $e$  não pertence a qualquer corte mínimo que separa  $s$  e  $t$ . Logo,  $\lambda^*$  é a capacidade crítica deste fluxo máximo. A Figura 4 ilustra este comportamento.

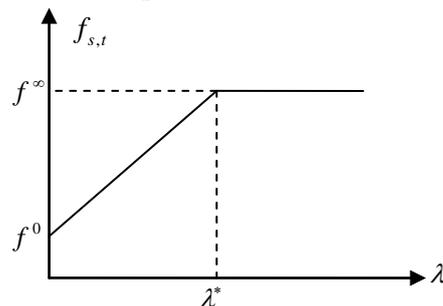


Figura 4: Fluxo máximo temporariamente sensível à variação de  $\lambda$ .

- Insensível à variação de  $c(e)$ . A variação de  $\lambda$ , para qualquer valor, não influencia o fluxo máximo entre  $s$  e  $t$ . A aresta  $e$ , neste caso, não pertence, em instante algum, a um corte mínimo ( $s$ - $t$ ). Observa-se que  $f^0 = f^\infty$ .

Portanto, para uma variação crescente, a capacidade crítica  $\lambda^*$  indica o valor da capacidade da aresta paramétrica a partir do qual a aresta deixa de pertencer a um corte mínimo que separa os vértices. É possível também afirmar que, se com capacidade nula a aresta não faz

parte de um corte mínimo, então ela continua não fazendo parte desse corte mínimo quando sua capacidade aumenta.

#### 4.2. Construção de Árvores de Cortes

Com o objetivo de diminuir a complexidade dos algoritmos que necessitam calcular duas ou mais árvores de cortes em sequência, Barth *et al.* (2006) mostraram como utilizar as informações de uma árvore de cortes já construída para construir eficientemente a seguinte.

Para uma rede que possui apenas uma aresta com capacidade paramétrica, a variação desta capacidade pode não influenciar o valor dos fluxos máximos (e dos cortes mínimos) para vários pares de vértices. Este resultado de Barth *et al.* é formalizado a seguir.

**Lema 1.** *Seja  $G = (V, E)$  uma rede com  $n$  vértices e  $e = (i, j) \in E$  tal que  $c(e) = \lambda$ . Seja  $s$  e  $t$  um par de vértices de  $G$ . Seja  $CT^\alpha$  uma árvore de cortes quando  $c(e) = \alpha$ . Se o caminho  $P_{s,t}$  em  $CT^\alpha$  não possui aresta em comum com  $P_{i,j}$ , então  $f_{s,t}^\lambda = f_{s,t}^\alpha, \forall \lambda > \alpha \geq 0$ .*

**Demonstração:** Utilizando a propriedade de corte da árvore de cortes (Definição 1, item 2), existe um corte mínimo  $C_{s,t}^\alpha$  separando  $s$  e  $t$  em que ambos os vértices  $i$  e  $j$  ( $e = (i, j)$ ) se encontram no mesmo lado do corte mínimo. Conseqüentemente, o corte não contém  $e$  para  $\lambda > \alpha$ , e é insensível à variação de  $\lambda$ .  $\square$

Para os próximos desenvolvimentos, a seguinte definição se faz necessária:

**Definição 2.** *Seja  $G = (V, E)$  uma rede conexa e acíclica, ou seja, uma árvore. Seja  $i$  e  $j$  um par de vértices de  $G$  e  $P_{i,j}$  o (único) caminho entre eles. Seja  $x_a$  um vértice de  $P_{i,j}$  e  $y_a^b$ ,  $0 \leq b < w_a$ , os vizinhos de  $x_a$  não contidos em  $P_{i,j}$ , onde  $w_a$  é o número total de vizinhos de  $x_a$ . Para cada  $y_a^b$ , seja  $T_a^b$  a maior subárvore de  $G$  enraizada em  $y_a^b$  não contendo  $x_a$ . O conjunto das árvores  $T_a^b$  é chamado de **floresta decomposta - (i, j)** de  $G$ , denotado por  $F_{i,j}$ .*

Considere uma árvore  $G$ , como na Figura 5, e o caminho  $i$ -3- $j$  em  $G$ . Então, a floresta decomposta- $(i,j)$  de  $G$  é formada pelas subárvores enraizadas nos nós  $y_i^0 = 4$ ,  $y_j^0 = 7$ ,  $y_3^0 = 1$  e  $y_3^1 = 2$  de  $G$ .

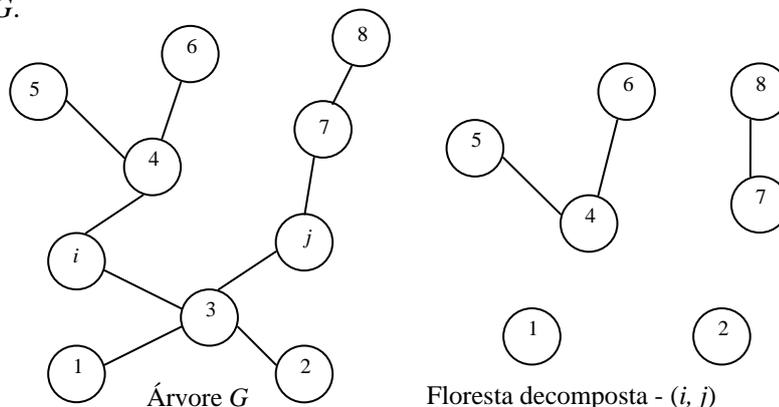


Figura 5: Exemplo de floresta decomposta- $(i, j)$  da árvore  $G$ .

**Lema 2.** *Seja  $G$  uma rede com uma aresta de capacidade paramétrica  $e = (i, j)$ . Seja  $CT^\alpha$  a árvore de cortes quando  $c(e) = \alpha$ . Seja  $F_{i,j}$  a floresta decomposta -  $(i, j)$  de  $CT^\alpha$ . Para cada*

árvore  $T_a^b \in F_{i,j}$ , existe uma árvore de cortes de  $G$  com  $c(e) = \lambda > \alpha$  que contém  $T_a^b$  como subárvore.

O Lema 2 pode ser demonstrado via aplicação do algoritmo de Gomory e Hu. Primeiramente, seja  $s_1 = x_a$  e  $t_1 = y_a^b$  pertencendo ao mesmo super nó. O caminho  $P_{s_1, t_1}$  em  $CT^\alpha$ , reduzido a aresta  $(x_a, y_a^b)$ , não possui aresta em comum com  $P_{i,j}$ . Assim, com base no Lema 1, o corte mínimo entre  $s_1$  e  $t_1$  pode ser obtido de  $CT^\alpha$ . A partir desta escolha, os super nós resultantes são  $SV_1$  com todos os elementos de  $T_a^b$  e  $SV_2$  com os outros elementos da rede. Continuando o algoritmo de Gomory e Hu para o super nó  $SV_1$  e sempre obtendo os cortes mínimos de  $CT^\alpha$ , ao final, teremos reconstruído toda a estrutura de  $T_a^b$ .

**Teorema 1.** *Seja  $G^\lambda$  uma rede com uma aresta  $e = (i, j)$  de capacidade paramétrica  $c(e) = \lambda$ . Seja  $CT^\alpha$  uma árvore de cortes obtida quando  $c(e) = \alpha$ . Seja  $P_{i,j}$  o caminho entre  $i$  e  $j$  em  $CT^\alpha$ . Para  $\lambda > \alpha$  é suficiente calcular  $|P_{i,j}| - 1$  cortes mínimos em  $G^\lambda$  a fim de obter a árvore de cortes  $CT^\lambda$ .*

A demonstração do Teorema 1 está baseada no algoritmo de Gomory e Hu. O ponto principal é escolher os nós  $s$  e  $t$  no algoritmo de forma a reconstruir, em primeiro lugar, de acordo com o Lema 2, todas as  $T_a^b$ , sem executar, sequer, um algoritmo de corte mínimo. Neste momento, a árvore intermediária encontra-se estruturada com um super nó composto por todos os elementos do caminho  $P_{i,j}$  ao qual todas as subárvores  $T_a^b$  estão conectadas. A partir deste ponto, o algoritmo de Gomory e Hu processará normalmente, executando, então,  $|P_{i,j}| - 1$  cortes mínimos.

Uma extensão ao Teorema 1 de Barth *et al.* (2006), para o caso de várias arestas com capacidades paramétricas, é apresentada neste trabalho a seguir.

**Teorema 2.** *Sejam  $CT$  uma árvore de cortes de uma rede  $G = (V, E)$  e  $P$  uma subárvore de  $CT$ . Considere  $E'$  um conjunto de arestas não contidas em  $E$ , mas com extremidades em  $P$ . Para se obter uma árvore de cortes  $CT'$  de  $G' = (V, E \cup E')$ , é suficiente calcular  $|P| - 1$  cortes mínimos em  $G'$ , sendo  $|P|$  é o número de vértices da subárvore.*

**Demonstração:** Duas observações importantes compõem esta demonstração.

1. Afirmar que uma dada aresta não está contida em uma rede é equivalente a afirmar que esta aresta está contida na rede e possui capacidade nula. Portanto, adicionar uma aresta a uma rede pode ser entendido como variar positivamente sua capacidade;
2. De acordo com o Lema 1, a variação crescente das capacidades das arestas  $E'$  não terá influência na árvore de cortes  $CT$ , exceto para a subárvore  $P$ .

Seguindo a idéia da demonstração do Teorema 1, são suficientes  $|P| - 1$  cálculos de corte mínimo em  $G'$  para se obter  $CT'$ .  $\square$

Note que os resultados nesta subseção, sobre construção de árvores de cortes, valem para o caso da variação crescente das capacidades de arestas paramétricas. Para o caso de variação decrescente, o problema se torna mais complicado, exigindo um estudo *à la* Elmaghraby.

Os Teoremas 1 e 2 formam a idéia central do algoritmo proposto neste trabalho.

## 5. Algoritmo proposto

O algoritmo proposto neste trabalho para resolver o problema de fluxo máximo multiterminal está baseado na teoria da análise de sensibilidade, e nas respectivas técnicas contração e construção de árvores de corte de Gomory e Hu e de Gusfield. Utilizar as informações de uma árvore de cortes existente para construir a próxima é a idéia central do algoritmo, permitindo, assim, que o procedimento para encontrar fluxos máximos/cortes mínimos seja aplicado sempre a redes menores que a original.

Para calcular o fluxo máximo entre todos os pares de nós de uma rede  $G = (V, E)$  não direcionada e com capacidades nas arestas, o algoritmo parte inicialmente de uma árvore geradora  $G_0$  (*spanning tree*) de  $G$ , ou seja, de um subgrafo de  $G$  contendo todos os seus vértices e que, também, é uma árvore. Esta poderá ser qualquer árvore geradora de  $G$ , não necessitando ser mínima ou máxima.

Sobre sua árvore de cortes (cut tree)  $CT_0$ , como  $G_0$  é uma árvore, ou seja, entre qualquer par de nós há somente um único caminho,  $CT_0$  é o próprio grafo  $G_0$ .

Considere  $E'$  o conjunto das arestas que estão em  $G$ , mas não em  $G_0$ , e escolha  $e = (i, j)$  pertencente a  $E'$  para ser adicionada ao grafo  $G_0$ , transformando-o em  $G_1$ .

Para o cálculo de  $CT_1$  lançamos mão dos seguintes resultados:

- Lema 2: existe uma  $CT_1$  idêntica a  $CT_0$  exceto pelo caminho  $i-j$ ;
- Teorema 1: são necessárias apenas  $P - 1$  aplicações do algoritmo de fluxo máximo/corte mínimo, sendo  $P$  o número de vértices no caminho  $i-j$ .

De acordo com a demonstração do Teorema 1, que se utiliza do método de Gomory e Hu, as  $P - 1$  execuções do algoritmo de fluxo máximo/corte mínimo poderão ser aplicadas a um grafo contraído de  $G_1$ .

De posse das extremidades  $i$  e  $j$  da aresta adicionada, o algoritmo identifica o caminho  $i-j$  em  $CT_0$ . A seguir, mapeia todas as subárvores  $T_a^b$  de  $CT_0$ . Cada subárvore será contraída a somente um vértice no grafo em que será aplicado o algoritmo de fluxo máximo. Identificado o caminho e mapeadas as subárvores, gera-se, então, utilizando as técnicas de contração do algoritmo de Gomory e Hu, o grafo contraído de  $G_1$ .

Por fim, o algoritmo, fazendo uso do algoritmo de Gusfield, calcula o fluxo máximo entre todos os pares de nós pertencentes ao caminho  $i-j$  de  $CT_0$ . Vale ressaltar que o algoritmo de fluxo máximo é sempre aplicado no grafo corrente contraído, do qual se obtém as duas informações seguintes: o valor do fluxo máximo entre o par de nós e o lado em que se encontra cada um dos nós do caminho  $i-j$  em relação ao corte mínimo.

Portanto, a árvore de cortes, resultante do algoritmo de Gusfield, substituirá o caminho  $i-j$  em  $CT_0$ . Desta forma, obtém-se  $CT_1$ .

Os procedimentos acima serão executados em um laço **para** até que todas as arestas de  $E'$  tenham sido adicionadas. Com essas adições, serão gerados  $G_2, G_3, \dots, G_y$  e suas respectivas árvores de corte  $CT_2, CT_3, \dots, CT_y$ , onde  $y = (m - n + 1)$  sendo  $m$  o número de arestas e  $n$  o número de nós do grafo  $G$ . O valor de  $y$  é simples de ser encontrado visto que o número de arestas de  $G_0$  é sempre  $n - 1$ . Note que apenas uma aresta é adicionada a cada iteração.

Ao término do laço, a árvore de cortes remanescente é  $CT_y$ , ou, apenas,  $CT$ , que é, portanto, a resposta ou a solução final do problema. A Figura 6 exibe o pseudocódigo do algoritmo denotado por A1.

**procedimento A1 ( $G$ );**

- 1 Construa uma árvore geradora  $G_0$  de  $G$ ;
  - 2 Faça  $CT_0 = G_0$ ;
  - 3  $E' \leftarrow$  conjunto de arestas que estão em  $G$  mas não estão em  $G_0$ ;
  - 4 **para**  $e_k \in E'$  onde  $k = 1, \dots, m - n + 1$  **faça**
  - 5     Adicione  $e_k$  ao grafo  $G_{k-1}$  criando  $G_k$ ;
  - 6     Identifique caminho  $i-j$  em  $CT_{k-1}$ ;
  - 7     Mapeie subárvores  $T_a^b$  em  $CT_{k-1}$ ;
  - 8     Gere o grafo contraído de  $G_k$ ;
  - 9     Calcule, com algoritmo de Gusfield, a árvore de cortes dos nós do caminho  $i-j$  no grafo contraído;
  - 10    Substitua o caminho  $i-j$  pela árvore de cortes em  $CT_{k-1}$ , gerando  $CT_k$ ;
  - 11 **finalize para**
  - 12 **retorne**  $CT_k$ ;
- finalize A1;**

Figura 6: Pseudocódigo do algoritmo A1.

A seguir, o algoritmo A1 é aplicado passo a passo em uma instância  $G$  da Figura 7. No primeiro passo do algoritmo constrói-se uma árvore geradora  $G_0$ , idêntica a  $CT_0$  (linhas 1 e 2).

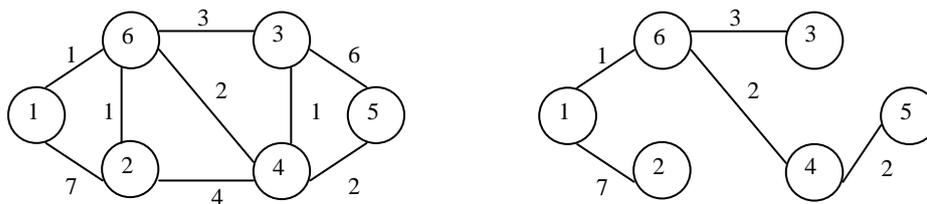


Figura 7: Redes  $G$  (esquerda) e  $G_0$  (direita).

Defini-se então o conjunto de arestas  $E' = \{(2,6), (2,4), (3,4), (3,5)\}$  que serão adicionadas a cada iteração ao grafo corrente na ordem em que se encontram no conjunto (linha 3).

Em seguida, a aresta  $(2,6)$  é adicionada, obtendo-se  $G_1$  (linha 5), e o caminho  $i-j$  é identificado em  $CT_0$  como sendo formado pelo nós 2, 1 e 6 (linha 6). Duas subárvores são então mapeadas: a composta pelo nó 3 e a composta pelos nós 4 e 5 (linha 7). Esta última, por ter mais de um nó, será contraída para o cálculo do algoritmo de fluxo máximo (linhas 8 e 9).  $G_1$  e  $G_1$  contraído estão ilustrados nas Figuras 8.

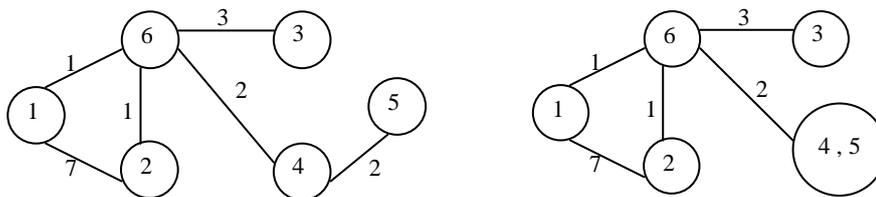


Figura 8: Grafo  $G_1$  (esquerda) e  $G_1$  contraído (direita).

Em  $G_1$  contraído será aplicado duas vezes o algoritmo de fluxo máximo, para construir, por meio da rotina de Gusfield, a árvore de cortes que substituirá o caminho 2-1-6 em  $CT_0$  (linha 10). Os nós origem e destino do algoritmo de fluxo máximo sairão do conjunto  $V = \{1, 2, 6\}$ . A Figura 9 nos mostra o resultado,  $CT_1$ .

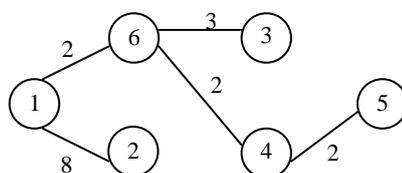


Figura 9:  $CT_1$ .

Observe que  $CT_1$  é a árvore de cortes de  $G_1$ . Termina aqui a execução da primeira iteração do laço **para** (linhas 4 - 11).

Pela ordem, a próxima aresta a ser adicionada é (2,4). Com suas extremidades, o caminho  $i-j$  em  $CT_1$  será 2-1-6-4. Neste caso não haverá contrações possíveis, pois as duas subárvores existentes possuem apenas um nó cada: subárvore nó 3 e subárvore nó 5. Portanto, os fluxos máximos serão computados no próprio  $G_2$ . A Figura 10 apresenta os grafos desta segunda iteração de A1.

A terceira aresta a ser incluída é (3,4), transformando  $G_2$  em  $G_3$ , que tem uma aresta a menos que  $G$  (Figura 7). Nesta iteração, o caminho  $i-j$  em  $CT_2$  é 3-6-2-4; nenhuma subárvore será contraída, portanto, não há  $G_3$  contraído; o algoritmo de fluxo máximo será aplicado três vezes em  $G_3$  para construir a árvore de cortes entre os nós do caminho  $i-j$ ; por fim, o caminho  $i-j$  será substituído pela árvore de cortes calculada. Figura 11 apresenta os grafos  $G_3$  e  $CT_3$ .

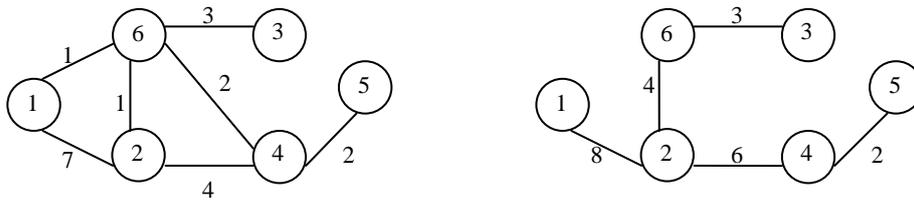


Figura 10:  $G_2$  (esquerda) e  $CT_2$  (direita).

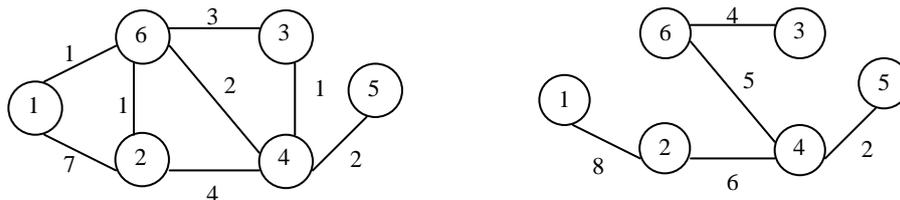


Figura 11:  $G_3$  (esquerda) e  $CT_3$  (direita).

A aresta (3,5) é a quarta e última a ser adicionada. Com ela,  $G_3$  se torna  $G_4$ , ou simplesmente  $G$ , o grafo original. As Figuras 12 e 13 ilustram os procedimentos executados nesta iteração. Observa-se que uma contração é possível e, então, o algoritmo de fluxo máximo é aplicado em  $G_4$  contraído.  $CT_4$ , ou  $CT$ , é a solução final do problema do fluxo máximo multiterminal.

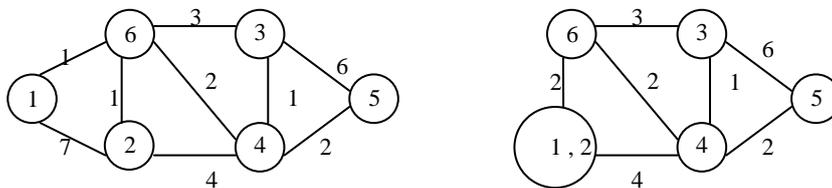


Figura 12:  $G_4$  ou  $G$  (esquerda) e  $G_4$  contraído (direita).

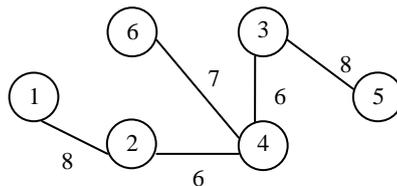


Figura 13:  $CT_4$  ou  $CT$ .

### 5.1. Algoritmo A2

Uma variante do algoritmo A1 é aqui apresentada. No algoritmo A2 é possível adicionar várias arestas ao grafo corrente na mesma iteração, diferentemente de A1 que tem somente uma única aresta adicionada ao grafo corrente por iteração.

Do Teorema 2 verifica-se que esta alteração no Algoritmo 1 é permitida, desde que todas as extremidades das arestas adicionadas em uma mesma iteração estejam no caminho  $i-j$ . Desta forma, o algoritmo procederá primeiramente, adicionando uma aresta ao grafo, em seguida, identificando o caminho  $i-j$ , e por último, procurando novas arestas a adicionar que satisfaçam o requisito acima.

### 6. Conclusão

Neste trabalho foi proposto um novo algoritmo para a resolução do *problema de fluxo máximo multiterminal* com o objetivo de reduzir o tempo de execução, ou seja, a complexidade em relação aos métodos tradicionais de Gomory e Hu e de Gusfield. O algoritmo foi elaborado utilizando funções de árvores geradoras, técnicas de contração de Gomory e Hu e de construção de árvore de cortes de Gusfield, e a teoria da análise de sensibilidade em redes. O passo a seguir é realizar testes computacionais com o algoritmo proposto comparando-o com a literatura, e verificar se o mesmo possui bom desempenho, e para quais famílias ou classes de grafos. Adicionar heurísticas como, não adicionar, em um primeiro momento, arestas que gerem um caminho  $i-j$  longo, e implementar diferentes funções de árvore geradora (mínima, máxima, ...), são opções para novas variâncias do algoritmo.

### Referências

- Agarwal, K. e Arora, S. R.**, Synthesis of multi-terminal communication nets: Finding one or all solutions. *IEEE Transactions on Circuits and Systems*, 23(3), 141-146, Março de 1976.
- Araujo, J. P. F.**, Algoritmos para acelerar a computação de Árvores de corte de Gomory e Hu. Tese de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro. Disponível em [http://www.maxwell.lambda.ele.puc-rio.br/18109/18109\\_1.PDF](http://www.maxwell.lambda.ele.puc-rio.br/18109/18109_1.PDF), Abril de 2011.
- Bako**, Solution of the multi-terminal minimal path problem in a network with gains. *Progress in Operations Research*, 12, 1974.
- Barth, D., Berthomé, P., Diallo, M. e Ferreira, A.**, Revisiting parametric multi-terminal problems: Maximum flows, minimum cuts and cut-tree computations. Em *Discrete Optimization*, 3 :195-205, 2006.
- Berthomé, P., Diallo, M. e Ferreira, A.**, Generalized parametric multi-terminal flows problem. Em *Graph-theoretic concepts in computer science*, Lecture Notes in Computer Science 2880, 71-80, 2003.
- Diallo, M.**, Réseaux de flots: Flots paramétrés et tarification. Ph.D. Thesis, Université de Versailles, França, (em francês). Dezembro de 2003.
- Diallo, M.**, *Méthodes d'Optimisation Appliquées aux Réseaux de Flots et Télécoms*. Editions universitaires européennes, 2011.
- Elmaghraby, S. E.**, Sensitivity analysis of multi-terminal network flows. *Journal of ORSA*, 12(5) :680-688, Janeiro de 1964.
- Ford, L. R., e Fulkerson, D. R.**, *Flows in Networks*. Princeton University Press, Princeton, NJ, 1973.
- Gomory, R. E. e Hu, T. C.**, Multi-terminal network flows. *SIAM Journal of Computing*, 9(4) :551-570, Dezembro de 1961.
- Gusfield, D.**, Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing*, 19 :143-155, 1990.
- Hu, T. C. e Shing, M. T.**, *Combinatorial Algorithms, Enlarged 2nd Ed.* Dover Publications, INC, Mineola, New York, 2002.
- Scutellà, M.**, A note on the parametric maximum flow problem and some related reoptimization issues. *Annals of Operations Research*(in press). Versão da conferência apareceu em INOC, 516-520, 2003.