Problema do Fluxo Máximo e Algoritmo de Ford-Fulkerson

João Victor Canavarro[1], **Renan Fonseca Cunha**[1]

[1] Faculdade de Computação - FACOMP
Instituto de Ciências Exatas e Naturais - ICEN
Universidade Federal do Pará (UFPA) – Belém, PA – Brasil

{canavarro,renancunhafonseca}@gmail.com

Abstract. *This paper presents a study on graph flow networks, the problem of maximum flow and its applications. The article also explains the Ford-Fulkerson algorithm, and presents an implementation of it performed by the authors of the article.*

Resumo. *Este artigo apresenta um estudo sobre redes de fluxo em grafos, o problema de fluxo máximo e suas aplicações. O artigo também explica o funcionamento do algoritmo de Ford-Fulkerson, e apresenta uma implementação do mesmo realizada pelos autores do artigo.*

1. Introdução

Este artigo está dividido da seguinte forma, a seção 2 apresenta a definição de rede de fluxo. A seção 3 apresenta o conceito de fluxo e fluxo máximo. A seção 4 mostra como grafos com arestas antiparalelas e com várias fontes ou sumidouros podem ser modelados em redes de fluxo. A seção 5 mostra exemplos de aplicações do problema de fluxo máximo e por último, a seção 6 apresenta o método de Ford-Fulkerson, seus conceitos e uma implementação do mesmo.

2. Redes de Fluxo

Uma rede de fluxo consiste em um grafo conexo, direcionado e anti-simétrico em que cada aresta possui uma capacidade (ou taxa) de transmissão positiva. Outra restrição de uma rede de fluxo é a proibição de laços [Cormen et al. 2009].

Nesse tipo de grafo, há dois vértices destacados, uma fonte e um sumidouro. Todo outro vértice do grafo se encontra no caminho entre a fonte e o sumidouro. Como todo vértice exceto a fonte possui no mínimo uma aresta de entrada, podemos afirmar que $E \geq V - 1$.

3. Fluxo

Fluxo é a quantidade (ou taxa) transmitida da fonte ao sumidouro em uma rede de fluxo. Algumas restrições precisam ser respeitadas:

- *Restrição de Capacidade:* Uma aresta deve ter fluxo de no mínimo 0 e de no máximo a sua capacidade.

$$0 \leq f(u,v) \leq c(u,v) \tag{1}$$

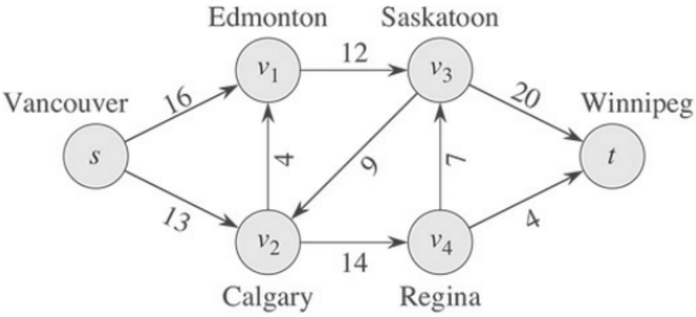

Figura 1. Exemplo de rede de fluxo

- *Conservação de Fluxo:* O fluxo de saída de um vértice deve ser igual ao fluxo de entrada deste mesmo vértice, com exceção da fonte e do sumidouro.

$$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v) \tag{2}$$

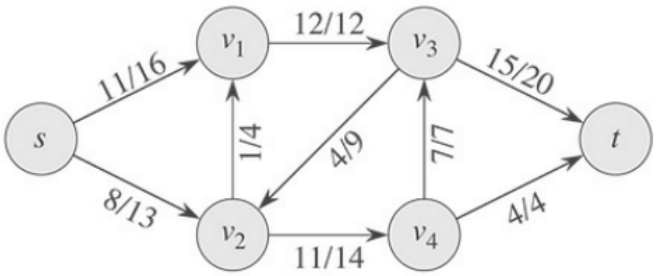

Figura 2. Uma rede de fluxo em que cada aresta possui os valores de fluxo e capacidade separados por uma barra e em ordem.

3.1. Fluxo Máximo

O fluxo máximo consiste na quantidade (ou taxa) máxima que uma rede pode transmitir de uma fonte para um sumidouro.

4. Modelagem do Problema

4.1. Arestas Antiparalelas

Como foi visto na seção 2, uma rede de fluxo deve ser anti-simétrica, ou seja, se houver uma aresta *(x, y)* não deve haver uma aresta *(y, x)*. Mas e se esse for o caso?

Devemos transformar este grafo em uma rede de fluxo equivalente que satisfaça a propriedade anti-simétrica através dos seguintes passos:

1. Identificar um par de arestas antiparalelas.
2. Escolher uma das arestas desse par, por exemplo (v_1, v_2).
3. Guardar a capacidade dessa aresta.
4. Remover essa aresta.
5. Criar um novo vértice *v'*.
6. Criar as arestas (v_1, v') e $v', v_2)$.

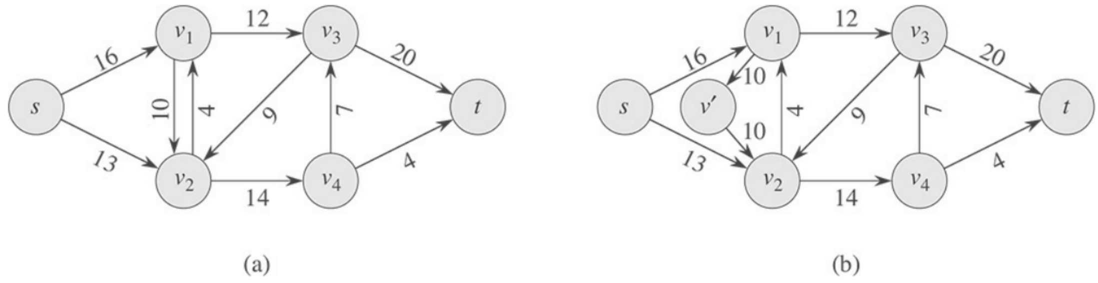

Figura 3. Ilustração da modelagem de um grafo que possui arestas antiparalelas para que possa ser aplicado no problema do fluxo máximo. A figura *a* possui o grafo com arestas antiparalelas, o grafo então é convertido na rede de fluxo da figura *b*.

4.2. Várias Fontes ou Vários Sumidouros

Para converter um grafo com várias fontes para uma rede de fluxo, devemos adicionar uma *superfonte* que possui aresta com todas as fontes com capacidade infinita.

Caso seja necessário aplicar o problema de fluxo máximo em um grafo com vários sumidouros, devemos fazer o mesmo, porém, criando um *supersumidouro* e adicionando aresta dos múltiplos sumidouros a este *supersumidouro* com capacidade infinita.

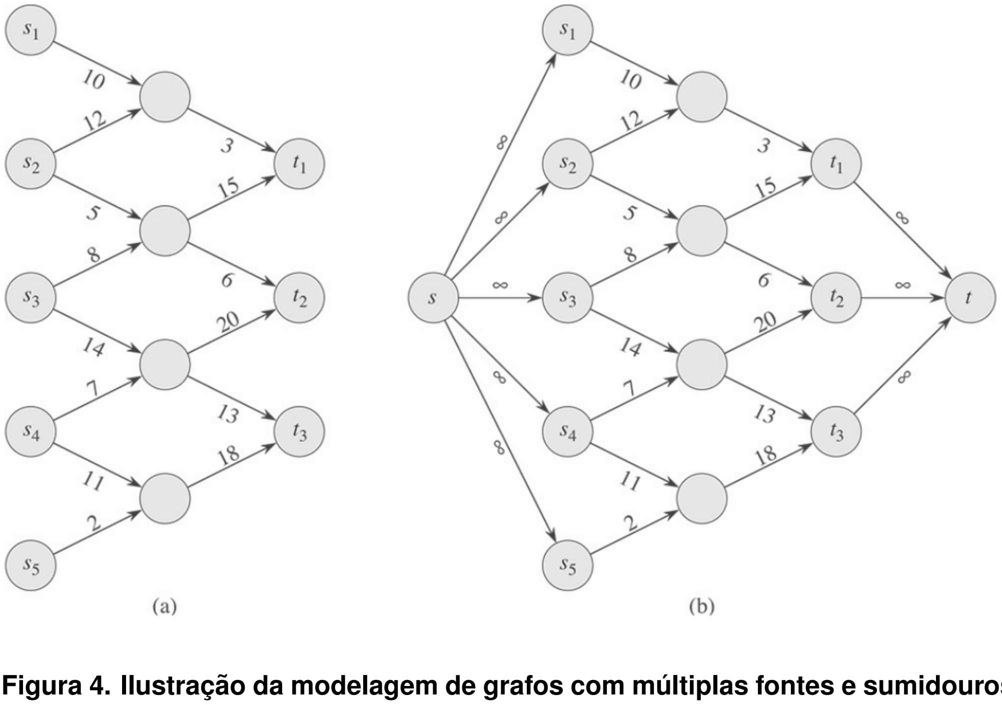

Figura 4. Ilustração da modelagem de grafos com múltiplas fontes e sumidouros para o problema de fluxo máximo. Na figura *a* há um grafo com 5 fontes e 3 sumidouros. Na figura *b* é criado uma rede de fluxo com uma *superfonte*, se conectando as 5 fontes, e um *supersumidouro*, se conectando aos 3 sumidouros

5. Aplicações

1. *Envio de Materiais:* Há um sistema com fontes, sumidouros e vértices interiores e o objetivo é saber qual a máxima taxa de envio de materiais das fontes para o

sumidouros.

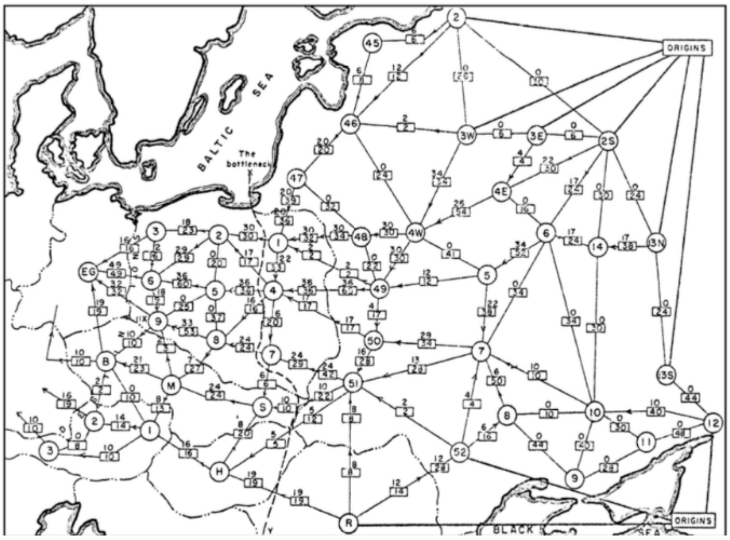

Figura 5. Linha de trem entre a Rússia e a Europa Oriental. Os pesos representam a taxa que um material poderia ser enviado de um lugar para o outro.

2. *Envio de Informações pela Internet:* Qual a máxima taxa de transmissão de informação que eu posso ter de um ponto a outro dado uma rede com vários nós e capacidades diferentes?

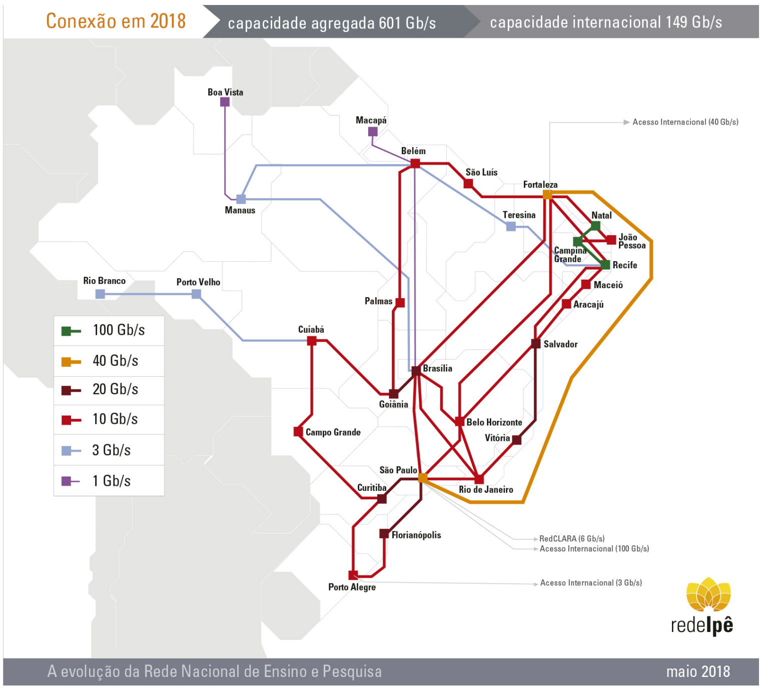

Figura 6. Ilustração da capacidade de transmissão da Rede Nacional de Ensino e Pesquisa - RNP

6. Método Ford-Fulkerson

O método Ford-Fulkerson é a abordagem mais conhecida para o problema do fluxo máximo um uma rede. Consideramos ele como um método ao invés de um algoritmo porque engloba várias implementações com diferentes tempos de execução.

Três ideias principais são utilizadas nesse método e em vários outros relacionados à análise de fluxo em uma rede: redes residuais, caminho de aumento ou caminho aumentador e cortes.

Dada uma rede de fluxo e um fluxo, uma rede residual consiste na representação das arestas que conseguem suportar mais fluxo que o atual. Uma aresta da rede de fluxo pode admitir uma quantidade de fluxo adicional igual à capacidade c da aresta menos o fluxo f nessa aresta. Se tal valor é positivo, colocamos essa aresta em uma rede residual Gf com uma capacidade residual de $cf(u, v) = c(u, v) - f(u, v)$.

Em uma rede residual Gf de um grafo G, a aresta (u, v) dirá quanto de capacidade residual a aresta do grafo G possui. Já a aresta (v, u) de Gf dirá quanto de fluxo está sendo transmitido pela aresta (u, v) do grafo G. A rede residual não aceita arestas com valor 0.

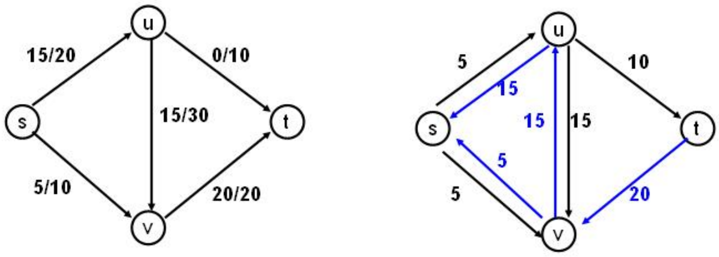

Figura 7. Exemplo de rede residual (a direita) de uma rede de fluxo (a esquerda).

Um caminho de aumento é um caminho de $S(fonte)$ para $T(sumidouro)$ por uma rede residual Gf de um grafo G. Cada aresta do caminho de aumento aceita mais um certo valor de fluxo sem violar a restrição de capacidade da aresta. O fluxo na rede pode ser incrementado igual a capacidade residual mínima do caminho aumentante.

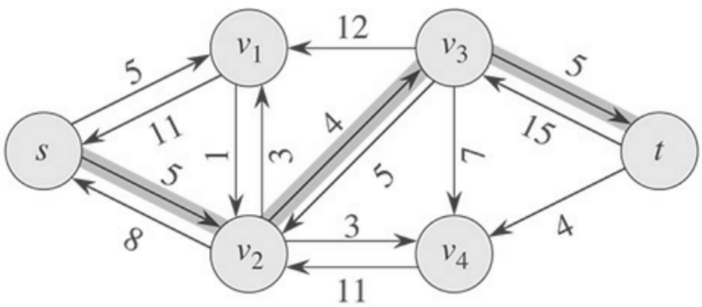

Figura 8. Rede Residual com Caminho de Aumento Sombreado

Define-se um corte (S, T) em uma rede de fluxo $G = (V, E)$ como sendo uma partição de V em S (fonte) e T (sumidouro), sendo $T = V - S$. Em outras palavras, um corte é qualquer conjunto de arestas que separa S de T. A capacidade de um corte é soma das capacidades das arestas que vão do conjunto de vértices que possui S para o conjunto de vértices que possui T.

A expressão corte mínimo é uma abreviatura de corte de capacidade mínima. Sendo assim, um corte mínimo tem a menor capacidade entre todos os cortes possíveis no grafo que representa a rede de fluxo. O teorema do Fluxo Máximo/Corte Mínimo diz que o fluxo máximo de uma rede é igual a capacidade do corte mínimo e que o fluxo em uma rede é máximo se e somente se sua rede residual não contém mais nenhum caminho de aumento.

6.1. Implementação

Em cada iteração do método Ford-Fulkerson é encontrado um caminho de aumento p que incrementa o fluxo f em cada aresta de p pela capacidade residual mínima $c_f(p)$ [Kumar 2019]. A implementação apresentada calcula o fluxo máximo em um grafo $G = (V, E)$ atualizando o fluxo $(u, v).f$ entre cada par u, v de vértices que são conectados por uma aresta.

O algoritmo que encontra um caminho aumentante pode ser implementado com o algoritmo de busca em largura - BFS. Nele, pode ser verificado se há um caminho se a fonte da rede de fluxo for antecessor de forma direta ou indireta do sumidouro.

```
FORD-FULKERSON(G, s, t)
for cada aresta (u, v) ∈ G.E
   (u, v).f = 0
while existir um caminho p de s a t na rede residual G_f
   c_f(p) = min{c_f(u, v) : (u, v) está em p}
   for cada aresta (u, v) em p
      if (u,v) ∈ E
         (u, v).f = (u, v).f + c_f(p)
      else (v, u).f = (v, u).f − c_f(p)
```

Se u e v não estão conectados por uma aresta o método assume que $(u, v).f = 0$. Por fim, a expressão $c_f(p)$ é uma variável temporária utilizada no código com o objetivo de armazenar a capacidade residual do caminho p.

Para cada aresta do caminho aumentante encontrado, o fluxo é aumentado na aresta caso ela exista na rede de fluxo. Caso contfrario, o fluxo da aresta na direção contrária é decrementado de acordo com a capacidade residual mínima.

O algoritmo finaliza quando não existir mais um caminho aumentante na rede residual do grafo. A partir da rede residual formada, podemos inferir o fluxo máximo e o fluxo que passa por cada aresta.

Para calcular o fluxo em cada aresta da rede de fluxo, podemos verificar o valor das arestas de direção contrária na rede residual. Para calcular o fluxo máximo, basta realizar o somatório dos valores de entrada na fonte da rede residual, lembrando que devem ser as arestas de entrada já que a rede residual representa o fluxo por uma aresta na direção contrária da aresta da rede de fluxo.

Uma implementação desse algoritmo foi realizada na linguagem de programação Python e está disponível de forma livre no seguinte endereço `https://github.com/renan-cunha/ford_fulkerson`.

Referências

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

Kumar, V. (2019). Maximum flow. `https://www.hackerearth.com/pt-br/practice/algorithms/graphs/maximum-flow/tutorial/`. Acessado em 20 de Junho de 2019.