

# MODELAGEM E OTIMIZAÇÃO DE FLUXO EM UMA REDE REAL CONECTADA

*Modeling and flow optimization in a real connected network*

*Modelado y optimización de flujo en una red real conectado*

Cézanne Alves Mendes Motta<sup>1</sup>, George Lauro Ribeiro de Brito<sup>\*1</sup>

<sup>1</sup>Curso de Ciências da Computação, Campus de Palmas, Universidade Federal do Tocantins, Palmas, Tocantins, Brasil.

\*Correspondência: Curso de Ciências da Computação – Universidade Federal do Tocantins, Av. NS 15, 109 Norte, S/N, Tocantins, Brasil. CEP:77.010-090. e-mail: gbrito@uft.edu.br



Artigo Original  
Original Article  
Artículo Original

Artigo recebido em 30/10/2015. Aprovado em 09/12/2015. Publicado em 24/02/2017.

## RESUMO

Este trabalho aborda o problema do fluxo máximo sob a perspectiva da teoria dos grafos, com o objetivo de implementar um algoritmo para a solução do problema, que poderá ser utilizado posteriormente para otimização de redes e de roteamento de pacotes *IP*. Inicialmente, são trazidas contextualização e definição do problema. Em seguida são apresentados os algoritmos estudados, metodologia e experimentos realizados. Foram implementadas duas técnicas da literatura para solução do problema: O algoritmo de Ford-Fulkerson e o de Edmonds-Karp. Um sistema no formato de uma rede de fibra ótica foi modelada em forma de grafo e os algoritmos foram aplicados ao modelo como prova de conceito. Para avaliar empiricamente a correta implementação dos algoritmos, bem como seu desempenho, uma base de grafos de variados tamanhos e densidades de arestas foi gerada aleatoriamente. Os algoritmos foram aplicados a estas instâncias e a comparação dos resultados mostram que o algoritmo de Edmonds-Karp possui desempenho superior.

**Palavras-chave:** Fluxo máximo; Otimização de Sistemas; Teoria dos grafos.

## ABSTRACT

*In this work, we approached the maximum flow problem from the graph theory perspective with the goal of providing a software implementation of an algorithmic solution to the problem for later use in optimization of network design and IP routing. Initially, we bring context and definition to the problem. Next, we present the studied algorithms, the methodology and the executed experiments. We coded two solutions to this problem from the literature: The Ford-Fulkerson algorithm and the Edmonds-Karp algorithm. A system in the form of a fiber optic network was modeled as a graph and the algorithms were applied to the model as proof of concept. Additionally, to empirically evaluate the algorithms' correctness and performance, we made a random graph base with varying sizes and edge densities and ran the algorithms on those. The comparison of the results show that the Edmonds-Karp algorithm has greater performance.*

**Keywords:** Maximum flow; Optimization of systems; Graph theory.

## RESUMEN

*Este documento aborda el problema de flujo máximo desde la perspectiva de la teoría de grafos, a fin de aplicar un algoritmo para resolver el problema, que luego se pueden utilizar para optimizar las redes y el enrutamiento de paquetes IP. Inicialmente, son llevados de contextualización y definición del problema. A continuación se presenta los algoritmos estudiados, la metodología y los experimentos. Ellos se llevaron a cabo dos técnicas literarias para resolver el problema: El algoritmo de Ford-Fulkerson y Edmonds-Karp. Un sistema en forma de red de fibra óptica se modeló en forma de gráfico y algoritmos se aplicaron al modelo como prueba de concepto. Para evaluar empíricamente la correcta aplicación de los algoritmos y su rendimiento, gráficos de base de diferentes tamaños y densidad de borde fue generado aleatoriamente. Los algoritmos se aplican a estos casos y la comparación de los resultados muestran que el algoritmo de Edmonds-Karp tiene un rendimiento superior.*

**Descriptor:** Flujo máximo; Optimización de los sistemas; La teoría de grafos.

## INTRODUÇÃO

O problema do fluxo máximo é um problema de otimização de sistemas que consiste em encontrar, dada uma rede, qual o maior fluxo possível de determinada grandeza partindo de um ponto a outro da rede, e por quais canais da rede e em que quantidade a grandeza deve escoar para que se alcance este fluxo máximo.

Existem diversas aplicações para esse problema, das quais podemos destacar: Projeto, manutenção e otimização de redes elétricas, de saneamento, de transporte (rodovias, vias aéreas e aeroporto) e de computadores; otimização de sistemas de distribuição de bens e serviços (uma rede de caminhões de uma empresa de entregas, por exemplo), e roteamento de pacotes IP. A otimização do fluxo é útil não só para quem projeta qualquer rede de recursos compartilhados, mas também para quem precisa utilizar uma rede disponibilizada por terceiros.

Diante da possibilidade de aplicação desse problema, foi proposta a implementação de um algoritmo para obter o fluxo máximo em uma rede de computadores, a fim de que pudesse ser utilizado posteriormente para otimização de roteamento, e auxílio a decisões na expansão das redes metropolitanas de ensino e pesquisa.

Ao longo desse trabalho foram realizados: Estudo do problema de fluxo máximo; pesquisa de métodos de solução de sistemas; modelagem matemática e computacional de uma rede de fibra ótica com foco na otimização; implementação de algoritmos computacionais para solução do problema; análise e avaliação dos resultados dos algoritmos no modelo dos sistemas; implementação de um gerador de redes aleatórias, execução dos algoritmos em uma base de instâncias geradas aleatoriamente para avaliação empírica do desempenho dos algoritmos.

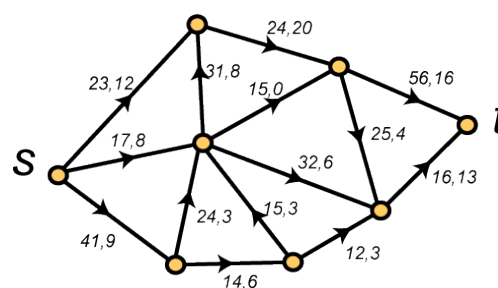
## Definição do problema

Em muitos problemas relacionados a sistemas que nos surgem, a forma mais simples de descrevê-los, é representá-los na forma de um grafo, uma vez que um grafo oferece uma representação geométrica que possibilita a obtenção de recursos algébricos na construção e desenvolvimento de um modelo matemático com vistas à resolução do problema (Goodaire & Parmenter, 1997)

Um grafo direcionado é uma dupla ordenada  $G = (V, E)$  de conjuntos, onde  $V$  é um conjunto de elementos quaisquer, e onde cada elemento “ $e$ ” de  $E$  é uma dupla ordenada  $(i, j)$  onde  $i$  e  $j$  são elementos de  $V$ .

Um sistema no formato de uma rede com fluxo pode ser entendido na forma de um grafo direcionado  $G$  onde cada aresta  $e = (i, j) \in E$  tem uma capacidade  $c_{ij}$  e um fluxo  $x_{ij}$  associados, e o fluxo da aresta não pode ultrapassar sua capacidade.

**Figura 1.** Exemplo de rede com fluxo no formato de um grafo.



Além disso, um fluxo deve satisfazer a restrição de conservação de fluxo, ou seja, a quantidade de fluxo que entra em um vértice é a mesma que sai, com exceção de um nó fonte (denotado por  $s$ ), de onde só sai fluxo e um nó sorvedouro (denotado por  $t$ ) onde só entra fluxo. Formalmente, a rede de fluxo é sujeita às seguintes restrições:

$$(1) \quad \forall v \in V - \{s, t\}: \sum_{(i,v) \in E} x_{iv} = \sum_{(v,j) \in E} x_{vj}$$

$$(2) \quad \forall (i,j) \in E: x_{ij} \leq c_{ij}$$

$$(3) \quad \forall (i,j) \in E: x_{ij} \geq 0$$

Assim, o problema de maximizar o fluxo da fonte para o sorvedouro pode ser definido da seguinte forma:

$$(4) \quad \max \sum_{(s,i) \in E} x_{s,i}$$

## MATERIAIS E MÉTODOS

### Algoritmo de Ford-Fulkerson

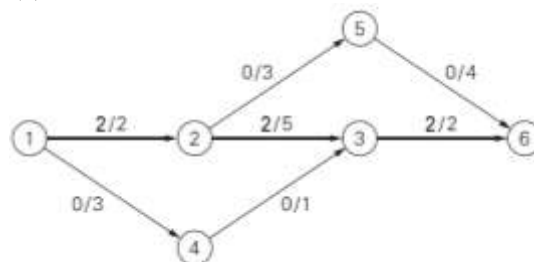
As restrições e a função objetivo do problema de fluxo máximo de um sistema no formato de uma rede são lineares, portanto este problema poderia ser resolvido por métodos de programação linear como o *Simplex*. Porém sua estrutura particular permite encontrar a solução através de algoritmos mais eficientes (Levitin, 2011).

Sendo assim, foi adotada a técnica de melhoria iterativa proposta por L. R. Ford, Jr., e D. R. Fulkerson (Ford & Fulkerson, 1956), que tem complexidade  $O(E|f|)$ , onde  $|f|$  é o valor do fluxo máximo (Cormen *et al.*, 2009) e se baseia em encontrar um caminho de aumento a cada iteração e aumentar o fluxo das arestas do caminho de acordo com o menor valor de capacidade restante de todas as arestas. Para isso, o algoritmo traça caminhos onde ele possa aumentar o fluxo de arestas com o mesmo sentido do caminho, ou diminuir o fluxo de arestas contrárias (Chen, 2003).

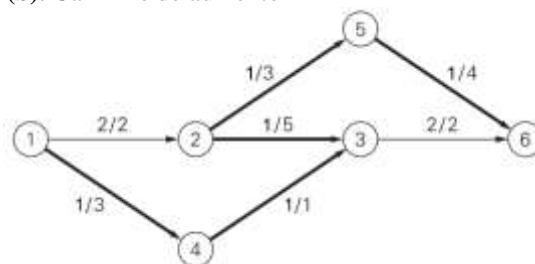
A **Figura 2 (b)** ilustra um caminho de aumento para o grafo da **Figura 2 (a)**, que não pode mais ter seu fluxo aumentado sem redução do fluxo de alguma aresta. Este procedimento garante que o fluxo obtido será máximo. A prova formal pode ser

encontrada no referido trabalho (Ford & Fulkerson, 1956).

**Figura 2(a).** Caminho de aumento



**Figura 2(b).** Caminho de aumento



### Algoritmo de Edmonds-Karp

A eficiência do método anterior pode ser aprimorada se determinarmos que a busca pelo caminho de aumento ocorra através de uma busca em largura, de maneira a tentar primeiro os caminhos com a menor quantidade de arestas. Este algoritmo foi proposto originalmente por J. Edmonds e R. M. Karp (Edmonds & Karp, 1972), e consiste em marcar os nós, na ordem em que forem encontrados pela busca, com o nó a partir do qual foi encontrado para que se possa fazer o caminho de volta a fonte a partir do nó destino. Este procedimento, definido no **Algoritmo 1**, resulta numa performance mais rápida, com complexidade de  $O(VE^2)$  (Cormen *et al.*, 2009).

### Sistema teste

Operada pela Rede Nacional de Ensino e Pesquisa (RNP)<sup>1</sup>, a rede Ipê é uma infraestrutura de um sistema no formato de uma rede de internet

<sup>1</sup> Maiores informações no site: [www.rnp.br](http://www.rnp.br).

dedicada à comunidade brasileira de ensino técnico e superior e de pesquisa, que interconecta universidades e seus hospitais, institutos federais, institutos de pesquisa e instituições culturais.

**Algoritmo 1.** Edmonds-Karp, pseudocódigo.

```

cria uma fila Q vazia
atribui  $x_{ij}=0$  para toda aresta  $(i,j)$  na rede
marca a origem com  $(\infty,-)$  e adiciona a origem à fila Q
enquanto (a fila Q não está vazia) faça
   $i = \text{RemoveFrente}(Q)$ ;
  para cada aresta de  $i$  para  $j$  faça
    se  $j$  está desmarcado
       $r_{ij} = U_{ij} - x_{ij}$ 
      se  $r_{ij} > 0$ 
         $l_j = \min\{l_i, r_{ij}\}$ ; marque  $j$  com  $(l_j, i-)$ ;
         $\text{Enfileira}(Q, j)$ ;
      para cada aresta de  $j$  para  $i$  faça
        se  $j$  está desmarcado
          se  $x_{ij} > 0$ 
             $l_j = \min\{l_i, x_{ij}\}$ ; marque  $j$  com  $(l_j, i+)$ ;
             $\text{Enfileira}(Q, j)$ ;
        se o destino foi marcado
           $j=n$ ; //começa no destino e segue as marcas de volta
          enquanto  $j \neq 1$  //a origem não foi alcançada
            se a segunda marca do vértice  $j$  é  $i+$ 
               $x_{ij} = x_{ij} + l_n$ ;
            senão
               $x_{ij} = x_{ij} - l_n$ ;
               $j=i$ ;  $i=0$  vértice indicado na segunda marca
              apague todas as marcas exceto as da fonte
              reinicia Q com a fonte apenas
          retorna  $\sum_j x_{ij}$  //a soma dos fluxos deixando a origem.

```

A infraestrutura da rede Ipê engloba 27 Pontos de Presença (PoPs), um em cada unidade da federação, além de ramificações para atender 1219 campi e unidades de instituições de ensino, pesquisa e saúde em todo o país, beneficiando mais de 3,5 milhões de usuários.

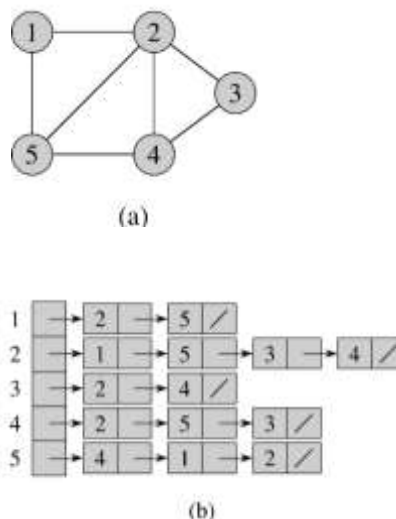
Para representar o grafo de fluxo da rede, a estrutura de dados escolhida foi a lista de adjacência, pois assim, a partir de um determinado nó  $v$ , temos acesso à uma lista encadeada dos nós que são adjacentes a  $v$ , não necessitando passar por todos os nós do grafo para checar se são adjacentes como no caso da matriz de adjacências. A **Figura 3(b)** ilustra uma representação no formato de lista de adjacências do grafo da **Figura 3 (a)**.

**Experimentos**

Os dois algoritmos foram implementados na linguagem de programação Java e foram testados no modelo de um grafo do sistema da Rede Ipê.

Para comparação e avaliação empírica dos algoritmos, criou-se uma base de 25 grafos conexos, ou seja, grafos nos quais se é possível chegar a qualquer nó a partir de qualquer outro nó do grafo.

**Figura 3.** Lista de adjacências.



Foram gerados grafos de 64, 128, 256, 512 e 1024 nós, com densidade de arestas de 1%, 10%, 25%, 50% e 75%, com a exceção dos grafos de 64 e 128 nós, onde 1% não seriam suficientes para manter o grafo conexo. Nestes casos o grafo foi construído com  $(n - 1) \times 2$  arestas. Além disso, para toda aresta  $(i, j)$ , os grafos possuem também a aresta  $(j, i)$ . Isso foi feito para melhor representar uma rede de computadores, onde, na maioria dos casos, a comunicação ocorre nos dois sentidos. Foram tomados aleatoriamente: os valores de capacidade das arestas (entre 1 e 10000); os nós ligados pelas arestas; e os nós fonte e o sorvedouro.

Cada grafo foi submetido aos dois algoritmos, utilizando-se os mesmos nós fonte e sorvedouro nas duas execuções.

Os algoritmos foram compilados na IDE NetBeans 8.0.2 com JDK 8, atualização 31. As execuções ocorreram na máquina virtual Java Versão 8, atualização 66, em um único processo no sistema

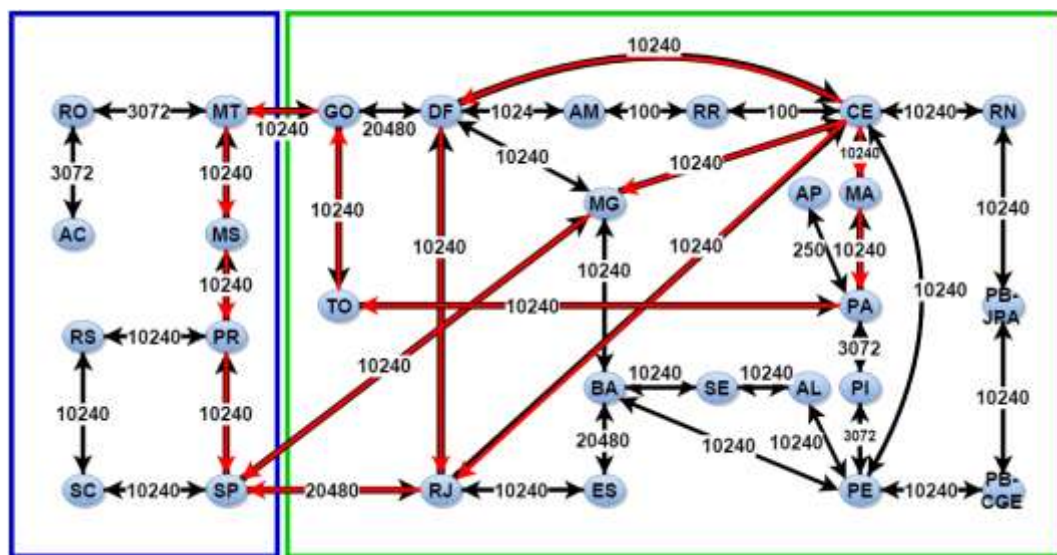
operacional Microsoft Windows 8.1 64-bit, em um processador Intel Core i7-4500U com *clock* em 2.78 GHz e 8 GB de memória RAM.

## RESULTADOS E DISCUSSÃO

No sistema da rede Ipê foram escolhidos como origem e destino para execução do algoritmo

dois nós com grande conectividade na rede para que o comportamento do algoritmo pudesse ser melhor visualizado. Como podemos ver na **Figura 4**, apesar dos estados do Ceará (CE) e de São Paulo (SP) terem respectivamente um total de 6244 Mb/s e 5120 Mb/s em links adjacentes, o fluxo total entre eles fica limitado por conexões intermediárias.

**Figura 4.** Fluxo na rede Ipê.



Ainda na **Figura 4**, pode-se confirmar que o fluxo obtido é máximo: Se particionarmos o grafo em dois, um lado contendo a fonte, e o outro contendo o destino do fluxo, podemos perceber que qualquer fluxo que saia do CE para SP precisa passar por uma das 3 arestas que cruzam do lado direito para o esquerdo, e todas elas estão saturadas, totalizando um fluxo de 40960 Mb/s.

### Teste de desempenho

Como esperado, os dois algoritmos obtiveram o mesmo resultado quando aplicados ao mesmo grafo, diferindo apenas no tempo gasto para obter o fluxo máximo. Tal resultado indica empiricamente que os

algoritmos funcionam corretamente, uma vez que mesmo expandindo os caminhos de aumento em ordens distintas, ambos obtiveram resultados idênticos em toda a base de grafos aleatória.

O resultado do teste de desempenho dos algoritmos pode ser visto nas **Tabelas 1 e 2**, e nos gráficos da **Figura 5**, que estão em escala logarítmica, mostrando que o algoritmo Edmonds-Karp (Figura 5(b)) possui desempenho superior. Os valores ausentes se devem a tempos de execução tão curtos que não puderam ser medidos.

Figura 5. Desempenho dos algoritmos.

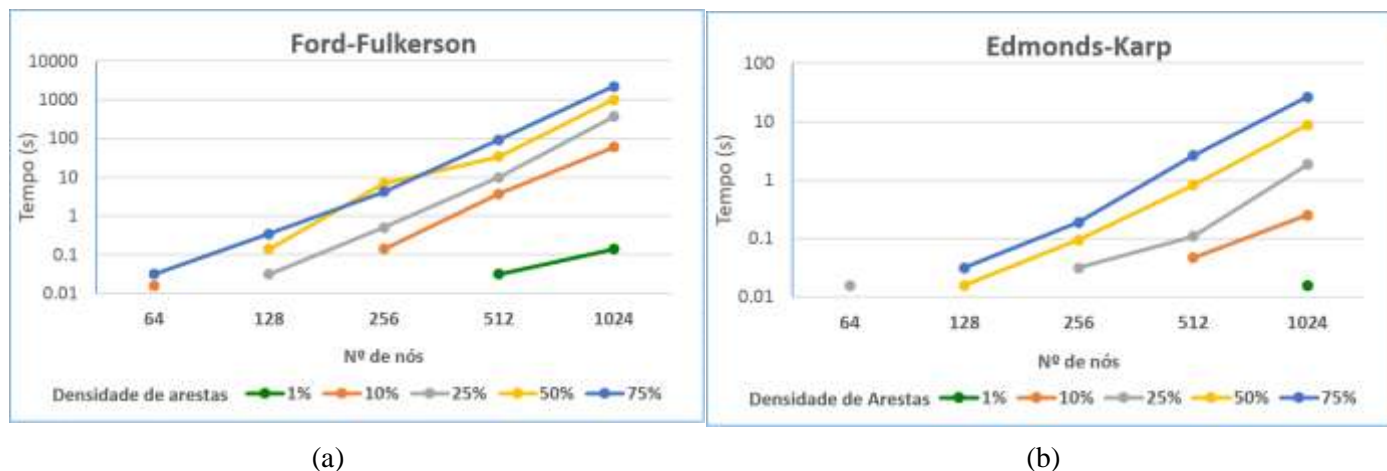


Tabela 1. Ford-Fulkerson.

Ford-Fulkerson – Tempo em segundos					
Densidade de arestas	64	128	256	512	1024
1%	-	-	-	0,03	0,14
10%	0,01	-	0,14	3,70	59,5
25%	-	0,03	0,5	9,82	364,25
50%	-	0,14	6,93	33,90	1007,09
75%	0,03	0,34	4,14	92,25	2231,42

Tabela 2. Edmonds-Karp.

Edmonds-Karp – Tempo em segundos					
Densidade de arestas	64	128	256	512	1024
1%	-	-	-	-	0,01
10%	-	-	-	0,04	0,25
25%	0,01	-	0,03	0,10	1,84
50%	-	0,01	0,09	0,81	8,76
75%	-	0,03	0,18	2,59	26,39

## CONCLUSÃO

O estudo desenvolvido ao longo deste trabalho resultou no desenvolvimento e implementação de dois algoritmos conhecidos da literatura, o algoritmo de Ford-Fulkerson e o de Edmonds-Karp, os quais foram testados num modelo de um sistema no formato rede conectada real, e em uma base de grafos gerados aleatoriamente.

A aplicação dos algoritmos no grafo modelado da rede Ipê mostra como os algoritmos podem ser utilizados para identificar gargalos na rede e fornecer métricas para um roteamento otimizado.

A aferição definitiva da corretude dos algoritmos só pode ser alcançada através de provas teóricas, as quais seus autores já proveram nos trabalhos citados. Entretanto os experimentos com a base de grafos aleatórios e os seus resultados consistentes corroboram empiricamente com os resultados dos autores. Além disso, os experimentos demonstram o desempenho de ambos os algoritmos à medida que a rede cresce tanto em nós, quanto em arestas, mostrando que o algoritmo de Edmonds-Karp, que propõe uma melhoria sobre o de Ford-Fulkerson, de fato possui um desempenho superior.

## AGRADECIMENTO

O presente trabalho foi realizado com o apoio da UFT e do CNPq.

---

Todos os autores declararam não haver qualquer potencial conflito de interesses referente a este artigo.

---

## REFERÊNCIAS

CHEN, W. K. **Net Theory and its Applications: Flows in Networks**, Series in Electrical and Computer Engineering, Vol. 1, Imperial College Press, 2003.

CORMEN, T.H.; LEISERSON, C.E.; RIVEST, R.L.; STEIN, C. **Introduction to Algorithms** (3 ed.), Capítulo 26. MIT Press, 2009.

EDMONDS, J.; KARP, R.M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, **Journal of the Association for Computing Machinery**, Vol. 19, No. 2, 1972.

FORD, L. R.; FULKERSON, D. R. "Maximal flow through a network". **Canadian Journal of Mathematics**, Vol 8, 1956.

GOODAIRE, E. G.; PARMENTER, M. M. **Discrete Mathematics with Graph Theory**. Prentice-Hall, 1997.

LEVITIN, A. V. **Introduction to the Design and Analysis of Algorithms** (3 ed), Pearson; 2011.