ALGORITMO DE FORD-FULKERSON

DCE529 - Algoritmos e Estruturas de Dados III

Atualizado em: 7 de outubro de 2024

Iago Carvalho

Departamento de Ciência da Computação

FLUXO EM GRAFOS

Problemas de fluxo abordam o processo de produção

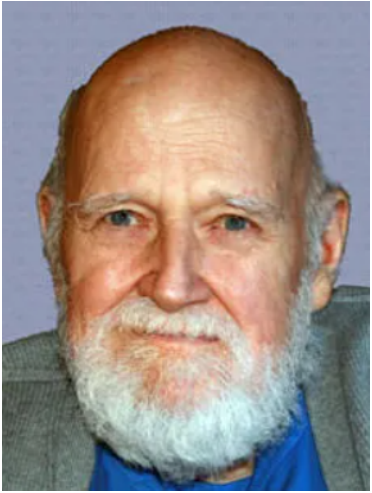
- Produtos tem origem em um ponto do grafo
- Produtos são consumidos em um outro ponto do grafo
- As arestas (ou arcos) indicam as possíveis ligações (caminhos) que o produto pode percorrer

Estes problemas ocorrem em diversas situações

- Plantas industriais
- Sistemas de comunicação e de transporte
- Distribuição de água e energia
- ...

Algoritmo desenvolvido em 1956 para o problema do fluxo máximo

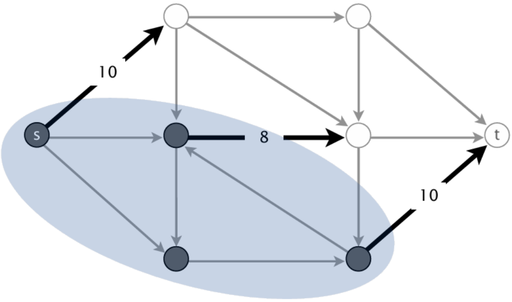
Lester Randolph Ford Jr.



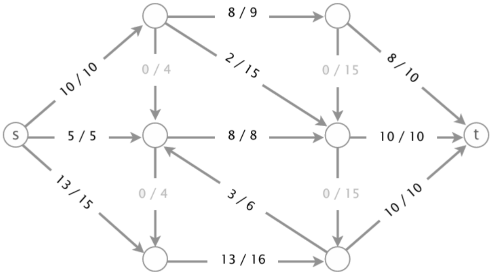
Delbert Ray Fulkerson

FLUXO MÁXIMO E CORTE MÍNIMO

Corte mínimo
10+8+10 = 28



Fluxo máximo
8+10+10 = 28



O **Algoritmo de Ford-Fulkerson** baseia-se em dois conceitos

- Pseudo-caminhos
- Pseudo-caminhos aumentadores

Inicia-se a solução com um pseudo-caminho

- Depois, iterativamente, encontra-se pseudo-caminhos aumentadores
- O algoritmo para quando não é possível encontrar outro pseudo-caminho

PSEUDO-CAMINHO

Um pseudo-caminho em um digrafo $G = (V, A)$ é uma sequência de vértices tal que, para cada par $i, j \in V$ de vértices consecutivos

- $(i, j) \in A$ ou $(j, i) \in A$

No primeiro caso, dizemos que (i, j) é um **arco direto**

Já no segundo caso, podemos dizer que (j, i) é um **arco reverso**

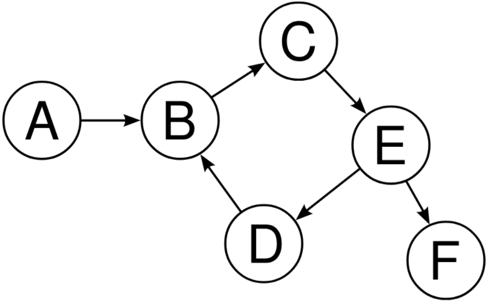
- Um pseudo-caminho que não possui arcos reversos é um caminho

PSEUDO-CAMINHO

Considere o pseudo-caminho A, B, D, E, F

- Os arcos (A, B) e (E, F) são diretos
- Os arcos (B, D) e (D, E) são reversos

Existe algum caminho entre A e F?



PSEUDO-CAMINHO AUMENTADORES

Considere uma rede $R = (V, A, F, U)$ como um **digrafo** (grafo direcionado) $G = (V, A)$ atravessado por um fluxo

- Vértice **s**: *source* (fonte), origem do fluxo
- Vértice **t**: *terminal* (sumidouro), destino do fluxo
- $\underline{u}(i,j) = 0 \, \forall (i,j) \in A$

Suponha que este grafo possua um fluxo F viável

- Arco (i,j) é **cheio** se $f_{ij} = \overline{u}(i,j)$
- Arco (i,j) é **vazio** se $f_{ij} = \underline{u}(i,j) = 0$

PSEUDO-CAMINHO AUMENTADORES

Em relação a esse fluxo, um pseudo-caminho é **aumentador** caso vá do vértice inicial ao final do digrafo tal que

- Nenhum arco direto do pseudo-caminho está cheio **e**
- Nenhum arco reverso do pseudocaminho está vazio

Dado um pseudo-caminho aumentador relativo a um fluxo F, a operação de enviar ϵ unidades de fluxo ao longo do pseudo-caminho consiste em calcular um novo fluxo

PSEUDO-CAMINHO AUMENTADORES

Este novo fluxo pode ser computado da seguinte maneira

Para cada arco direto (i, j) do pseudo-caminho, faça $f_{ij} = f_{ij} + \varepsilon$

- Ou seja, some ε ao fluxo no arco

Para cada arco reverso (j, i) do pseudo-caminho faça $f_{ji} = f_{ji} - \varepsilon$

- Ou seja, subtraia ε ao fluxo no arco

Esta operação aumenta ε ao fluxo total do grafo

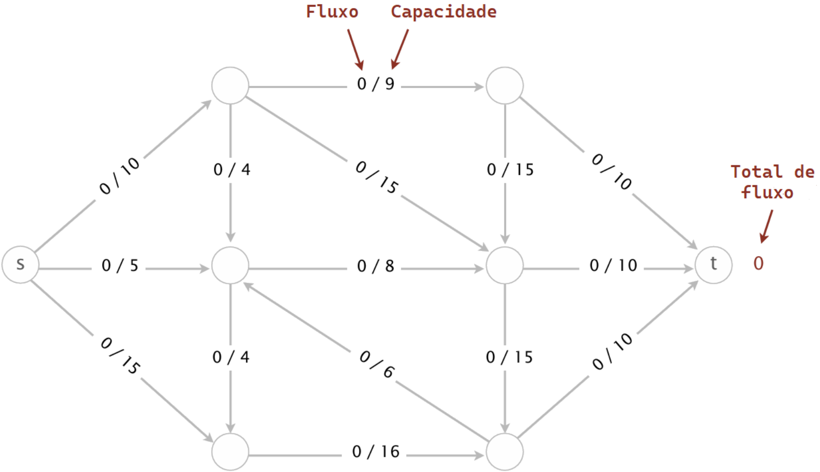
ALGORITMO DE FORD-FULKERSON

Fluxo $= 0$
while Existir pseudo-caminhos aumentadores **do**
    Encontrar um pseudo-caminho aumentador
    Computar o valor ε do arco limitante
    Alterar o valor dos arcos do caminho por ε
end while

Algorithm 1: Ford-Fulkerson

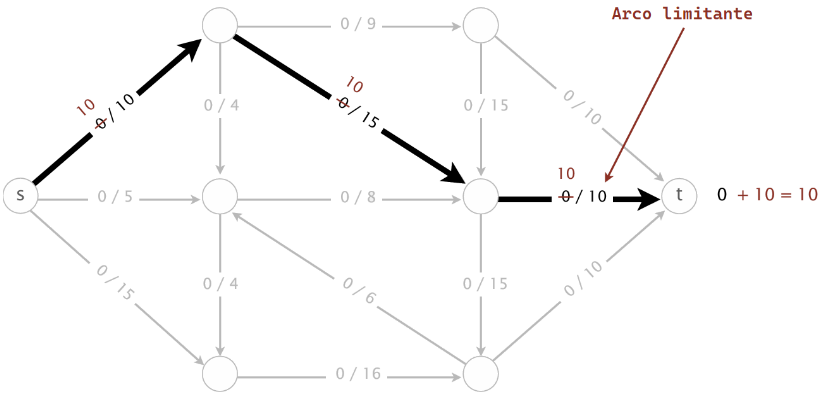
PSEUDO-CAMINHO AUMENTADORES

Considerar o grafo abaixo. **s** é o vértice inicial e **t** é o vértice final

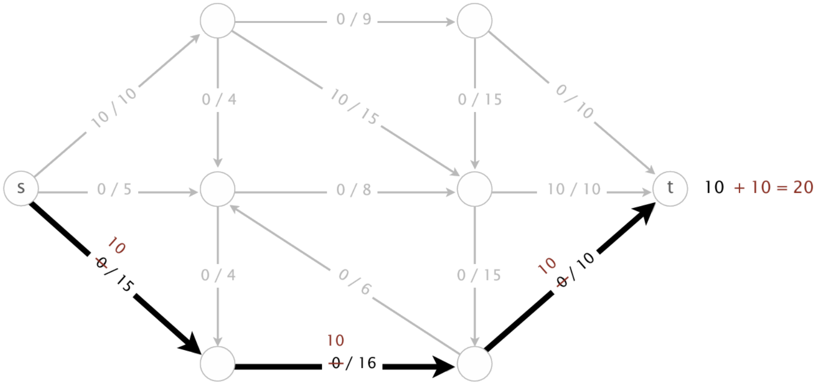


PSEUDO-CAMINHO AUMENTADORES

Vamos definir os pseudo-caminhos aumentadores neste grafo

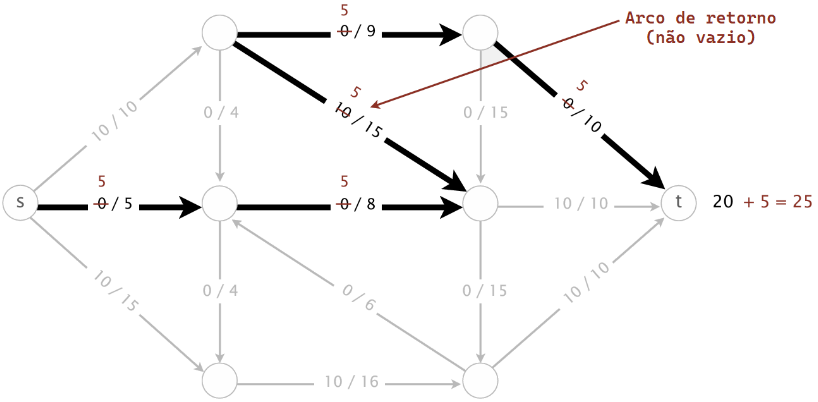


Segundo pseudo-caminho

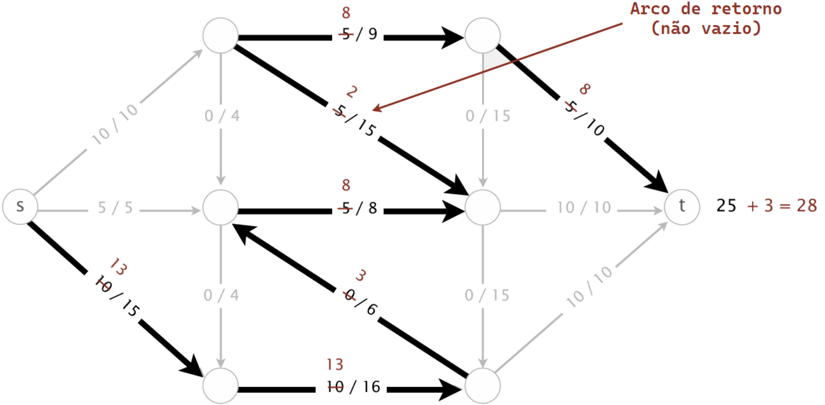


PSEUDO-CAMINHO AUMENTADORES

Terceiro pseudo-caminho



Quarto pseudo-caminho



O consumo de tempo do algoritmo de Ford-Fulkersonal é proporcional ao número de iterações

- Complexidade de $\mathcal{O}(|V|f)$, onde f é o fluxo máximo da rede

Quantos pseudo-caminhos aumentadores são necessários para produzir um fluxo máximo a partir do fluxo nulo?

Número de pseudo-caminhos aumentadores

Se todos os arcos do grafo têm capacidade inteira e menor que uma constante f então o número de pseudo-caminhos aumentadores necessário para atingir um fluxo máximo é menor que $|V|f$, sendo V o conjunto de vértices do grafo

COMPLEXIDADE DO FORD-FULKERSON

Como as capacidades dos arcos são números inteiros, o fluxo em cada arco é um número inteiro.

Portanto, ao longo da execução do algoritmo, a capacidade residual de cada arco em qualquer pseudo-caminho aumentador é um número inteiro positivo.

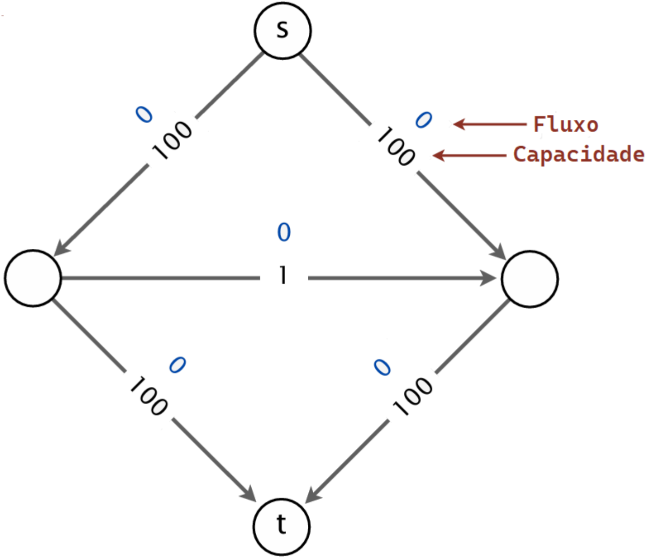
Logo, cada pseudocaminho aumentador tem capacidade residual pelo menos 1 e assim aumenta em pelo menos 1 a intensidade do fluxo.

Além disso, não existe nenhum fluxo de capacidade maior do que $|V|f$

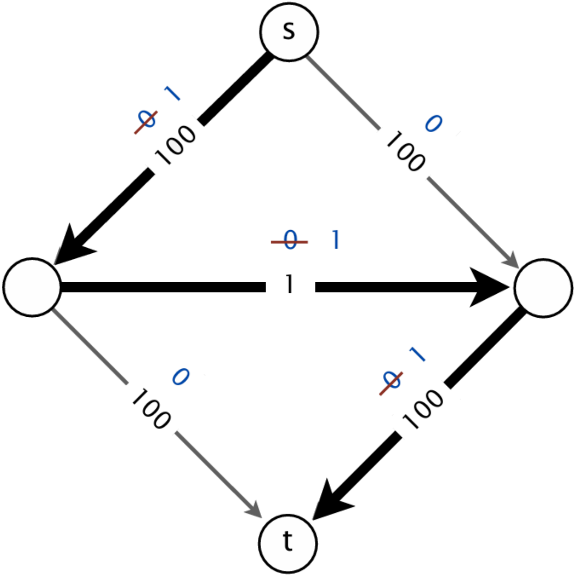
Como não temos arcos paralelos, o número de arcos que saem do vértice inicial é menor que $|V|$

- Portanto, qualquer fluxo tem intensidade menor que $|V|f$

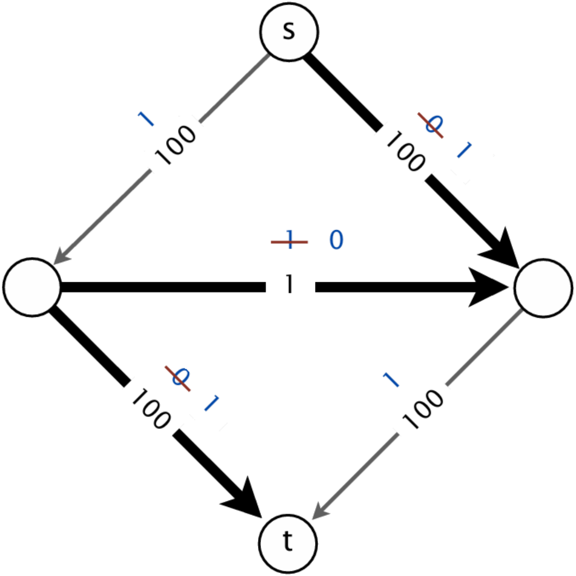
PIOR CASO DO FORD-FULKERSON



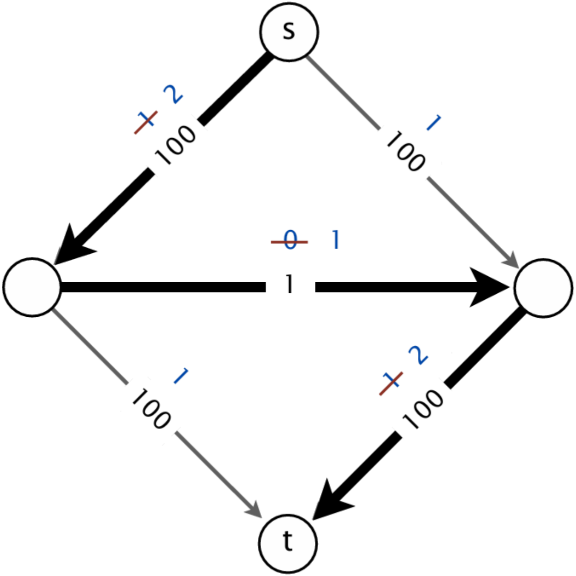
PIOR CASO DO FORD-FULKERSON



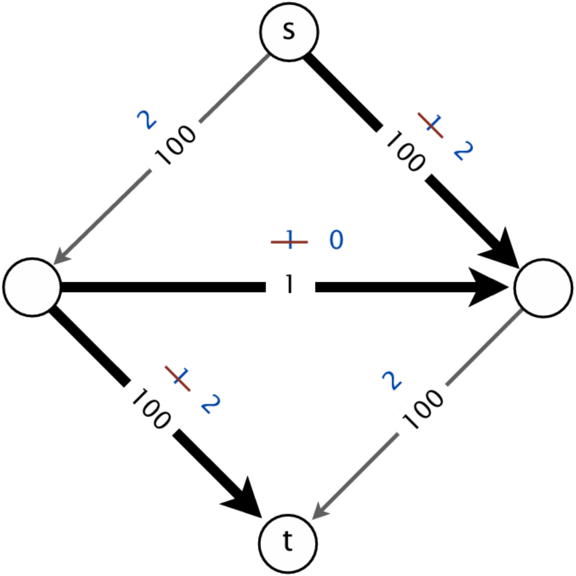
PIOR CASO DO FORD-FULKERSON



PIOR CASO DO FORD-FULKERSON



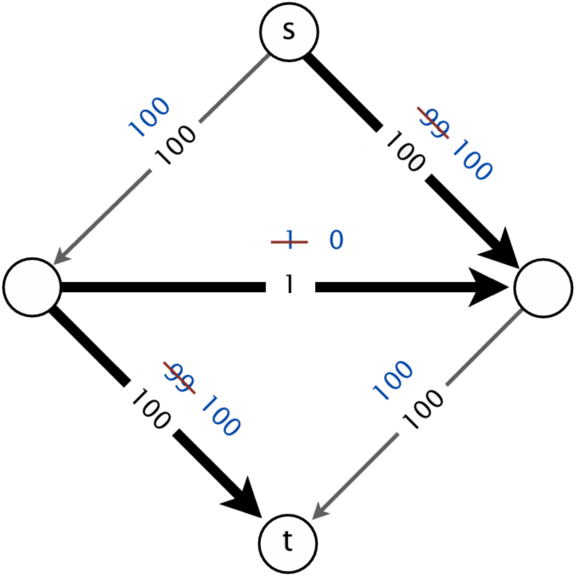
PIOR CASO DO FORD-FULKERSON



PIOR CASO DO FORD-FULKERSON

...

PIOR CASO DO FORD-FULKERSON



COMO MELHORAR O FORD-FULKERSON

Pode-se facilmente melhorar a complexidade do algoritmo de Ford-Fulkerson e evitar este pior caso

- Tudo depende da maneira como é encontrado um pseudo-caminho aumentador

Método	Número de caminhos
Aleatório	$\mathcal{O}(\lvert E\rvert\, U)$
Busca em profundidade	$\mathcal{O}(\lvert E\rvert\, U)$
Caminho mais curto	$\frac{1}{2}\mathcal{O}(\lvert V\rvert \lvert E\rvert)$
Caminho mais largo	$\mathcal{O}\left(\lvert E\rvert log(\lvert E\rvert\, U)\right)$

Algoritmo de Edmonds-Karp (Caminho mais curto)

- Complexidade $\mathcal{O}(|V||A|^2) = \mathcal{O}(|V|^5)$

Algoritmo de Dinic (Caminho mais largo)

- Complexidade $\mathcal{O}(|V|^2|A|) = \mathcal{O}(|V|^4)$

Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems

JACK EDMONDS

University of Waterloo, Waterloo, Ontario, Canada

AND

RICHARD M. KARP

University of California, Berkeley, California

ABSTRACT. This paper presents new algorithms for the maximum flow problem, the Hitchcock transportation problem, and the general minimum-cost flow problem. Upper bounds on the numbers of steps in these algorithms are derived, and are shown to compare favorably with upper bounds on the numbers of steps required by earlier algorithms.

Dokl. Akad. Nauk SSSR
Tom 194 (1970), No. 4

Soviet Math. Dokl.
Vol. 11 (1970), No. 5

ALGORITHM FOR SOLUTION OF A PROBLEM OF MAXIMUM FLOW IN A NETWORK WITH POWER ESTIMATION

UDC 518.5

E. A. DINIC

Different variants of the formulation of the problem of maximal stationary flow in a network and its many applications are given in [1]. There also is given an algorithm solving the problem in the case where the initial data are integers (or, what is equivalent, commensurable). In the general case this algorithm requires preliminary rounding off of the initial data, i.e. only an approximate solution of the problem is possible. In this connection the rapidity of convergence of the algorithm is inversely proportional to the relative precision.