
Grafos: fluxo em redes

SCC0216 Modelagem Computacional em Grafos

Thiago A. S. Pardo

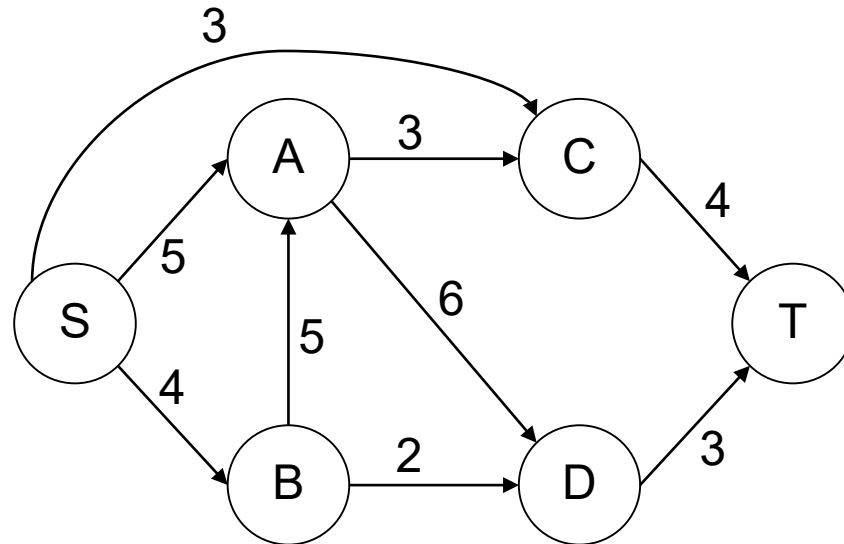
O problema

Um cenário...

- Sistema de **forneimento de água**
 - Arestas indicam tubulações com capacidades (positivas) de volume de água
 - Vértices são pontos de união e transferência de água entre tubulações
 - Há uma fonte (S - *source*) e um destino (T - *target*)
 - Origem/produtor vs. receptor/consumidor/usuário/sorvedouro/“ralo”
 - Tudo que sai da fonte, deve chegar ao destino
 - Água só flui em uma direção em cada tubulação

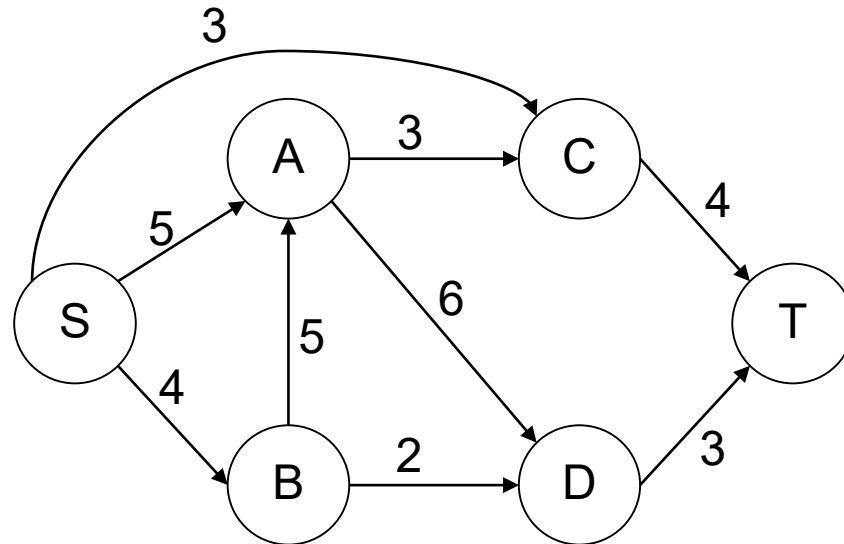
Um cenário...

- Sistema de fornecimento de água: grafo dirigido ponderado
 - Qual a capacidade máxima possível que chega a T?



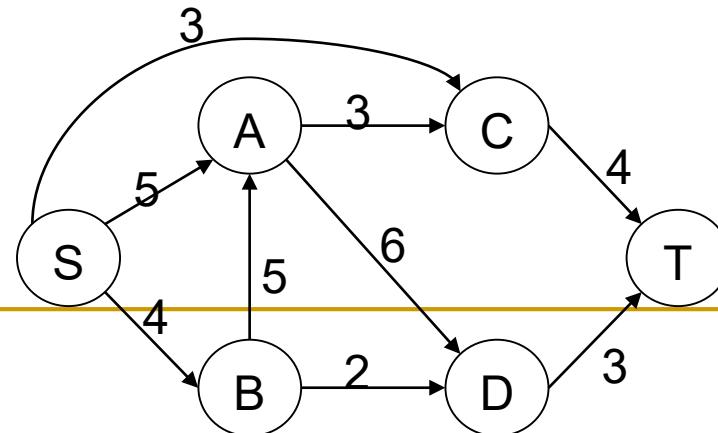
Um cenário...

- Sistema de fornecimento de água: grafo dirigido ponderado
 - Como maximizar a quantidade de água fluindo de S até T?



Problema

- Fluxo: taxa em que o material é transportado no sistema
- Conservação do fluxo
 - material passa pelos vértices sem acumulação
 - taxa de entrada = taxa de saída
- Cada aresta tem uma capacidade



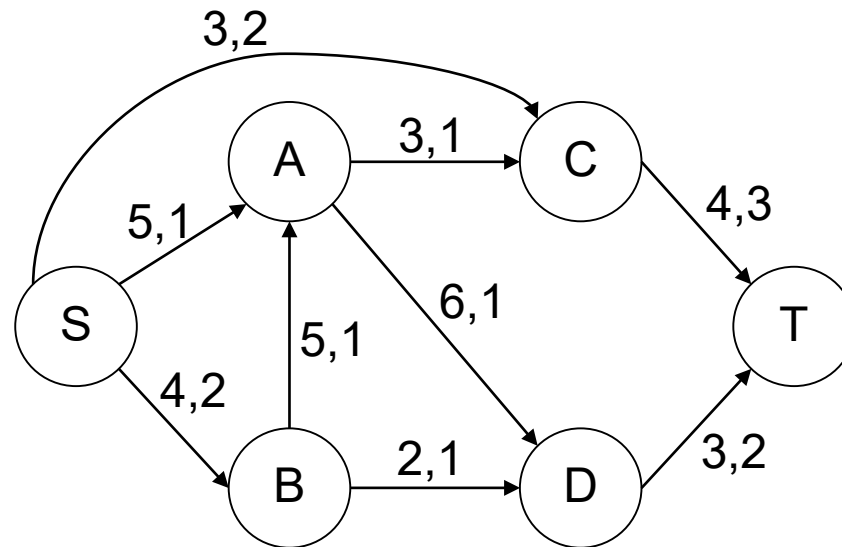
Problema

■ Problema do fluxo máximo

- Qual a maior taxa na qual o material pode ser enviado da origem ao destino
- Sujeito às restrições anteriores de capacidade e conservação

Problema

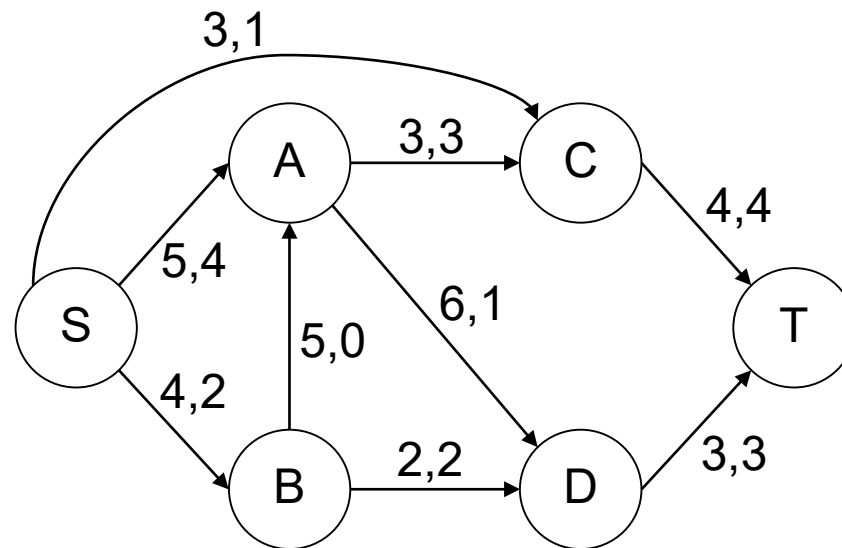
- Possibilidade de fluxo
 - A notação x,y nas arestas indica capacidade (x) e fluxo real (y)
 - Por exemplo, a aresta (S,A) tem capacidade 5 e fluxo 1



Dá para melhorar?
Se sim, como?

Problema

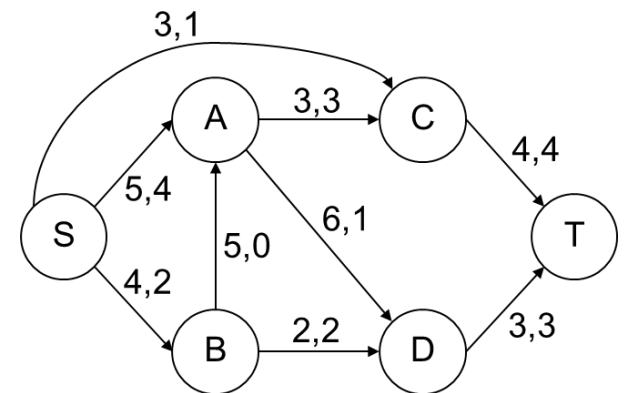
- Possibilidade de fluxo
 - A notação x,y nas arestas indica capacidade (x) e fluxo real (y)
 - Agora, a aresta (S,A) tem capacidade 5 e fluxo 4



Dá para melhorar?
Se sim, como?

Primeiras lições

- É possível perceber alguns **comportamentos** do sistema
 - O fluxo real $f(u,v)$ entre dois vértices u e v deve ser igual ou menor do que a capacidade máxima de fluxo $c(u,v)$: $f(u,v) \leq c(u,v)$
 - O fluxo mínimo possível é zero: $f(u,v) \geq 0$
 - Para dois vértices sem arestas entre si, então $c(u,v) = 0$ e $f(u,v) = 0$
 - Todo o fluxo que sai de S deve chegar a T:
 $\sum_{v \in \text{Vértices}} f(S,v) = \sum_{v \in \text{Vértices}} f(v,T)$
 - Com exceção de S e T, para cada vértice v , o total que sai (*outflow*) deve ser equivalente ao total que entra (*inflow*): $\text{outflow}(v) = \text{inflow}(v)$
 - Para S e T, $\text{outflow}(S) = \text{inflow}(T)$
 - **Somente S** produz água
 - **Somente T** consome água
 - Há mais de uma função de fluxo máxima (= ótima) possível
 - **Alternativa para o exemplo ao lado?**



O desafio

- Determinar uma **função de fluxo máximo**, ou seja, a melhor configuração de $f(v_i, v_j)$ para o grafo
 - Obedecendo-se as restrições anteriores, logicamente
- Diversas situações além do fornecimento de água
 - Corrente por uma rede elétrica
 - Trens em um sistema ferroviário
 - Bits em uma rede de comunicação
 - Peças em uma linha de montagem
 - Etc.

Soluções

Ford-Fulkerson

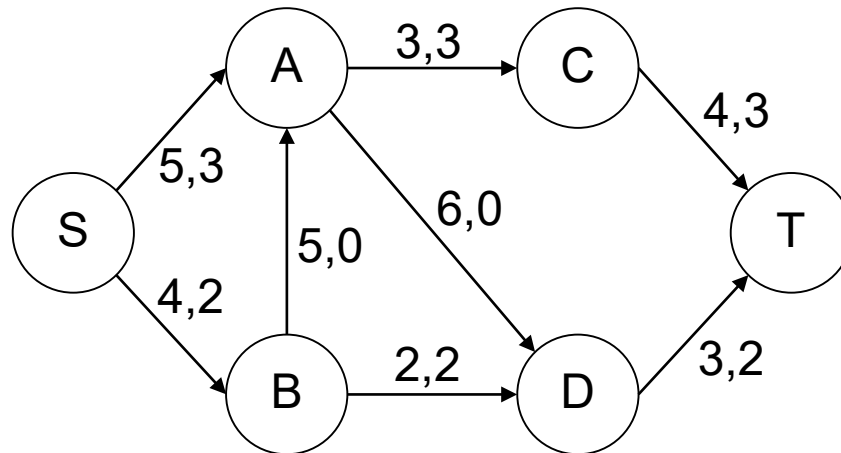
- Um método: engloba múltiplas implementações
- Método iterativo: parte de um fluxo nulo em cada vértice, e procura aumentar esse fluxo enquanto for possível
 - Caminho `aumentante`: um caminho de S a T ao longo do qual podemos empurrar mais fluxo

Melhorando uma função de fluxo

- Dada uma função de fluxo f (possivelmente inicializada com zero), encontrar um caminho $S = v_1, v_2, \dots, v_n = T$, tal que
 - O **fluxo em cada aresta** (v_i, v_j) no caminho seja **menor** do que a sua **capacidade**
 - Capacidade residual da aresta (v_i, v_j) : $c_f(v_i, v_j) = c(v_i, v_j) - f(v_i, v_j)$
 - Seja possível aumentar o fluxo pelo **menor valor** c_f no caminho
 - Dessa forma, ao final, pelo menos uma aresta do caminho terá $c(v_i, v_j) = f(v_i, v_j)$

Melhorando uma função de fluxo

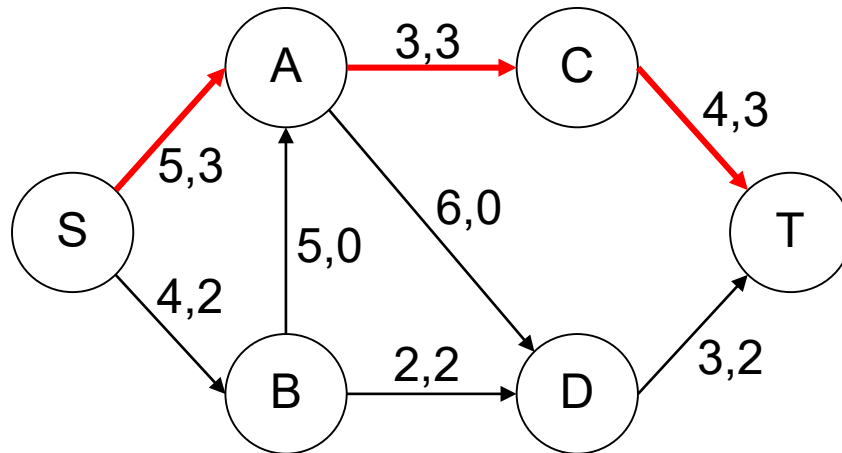
Exemplo



Melhorando uma função de fluxo

Exemplo

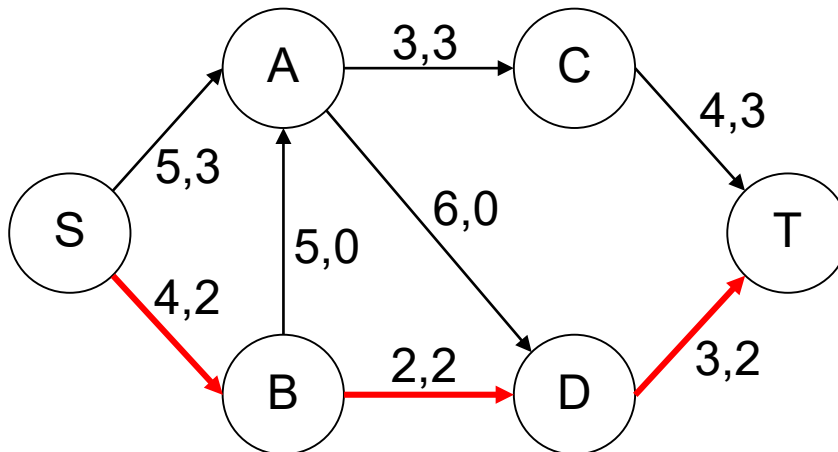
- Caminhos possíveis
 - (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo



Melhorando uma função de fluxo

Exemplo

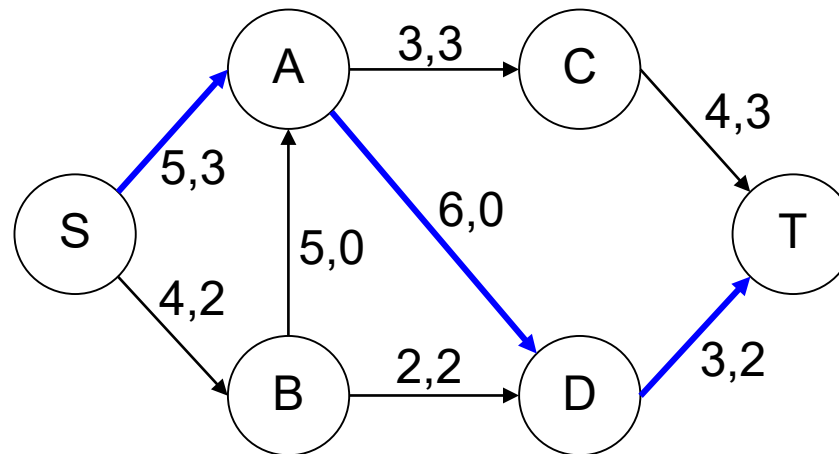
- Caminhos possíveis
 - (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo
 - (S,B,D,T): **infrutífero**, pois (B,D) já tem fluxo máximo



Melhorando uma função de fluxo

Exemplo

- Caminhos possíveis
 - (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo
 - (S,B,D,T): **infrutífero**, pois (B,D) já tem fluxo máximo
 - (S,A,D,T): **possível aumentar**, já que todas as arestas têm fluxo menor que capacidade, podendo-se aumentar pelo menor valor $c(v_i, v_j) - f(v_i, v_j)$ no caminho, que equivale a 1 para a aresta (D,T)

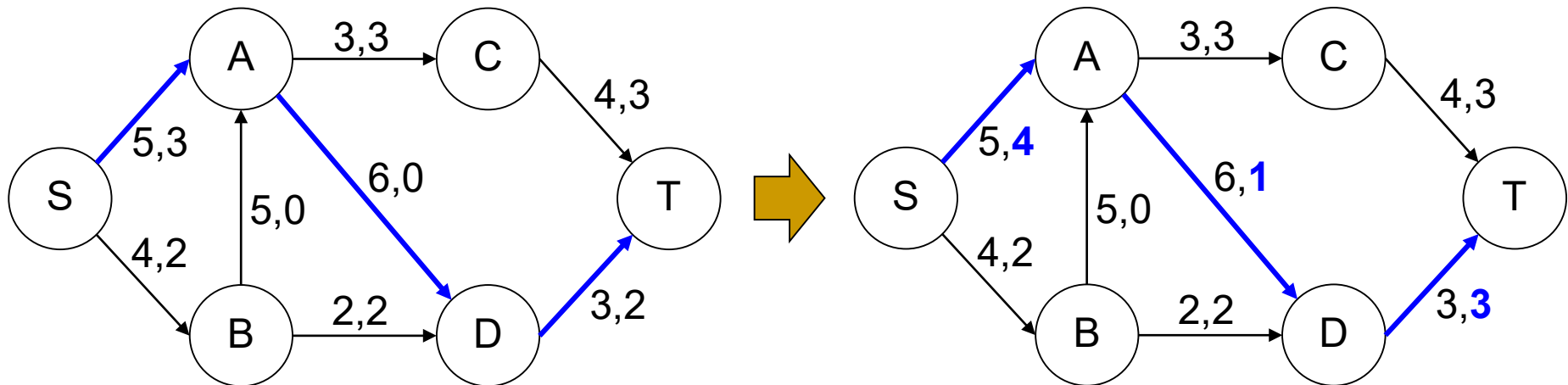


Melhorando uma função de fluxo

Exemplo

■ Caminhos possíveis

- (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo
- (S,B,D,T): **infrutífero**, pois (B,D) já tem fluxo máximo
- (S,A,D,T): **possível aumentar**, já que todas as arestas têm fluxo menor que capacidade, podendo-se aumentar pelo menor valor $c(v_i, v_j) - f(v_i, v_j)$ no caminho, que equivale a 1 para a aresta (D,T)

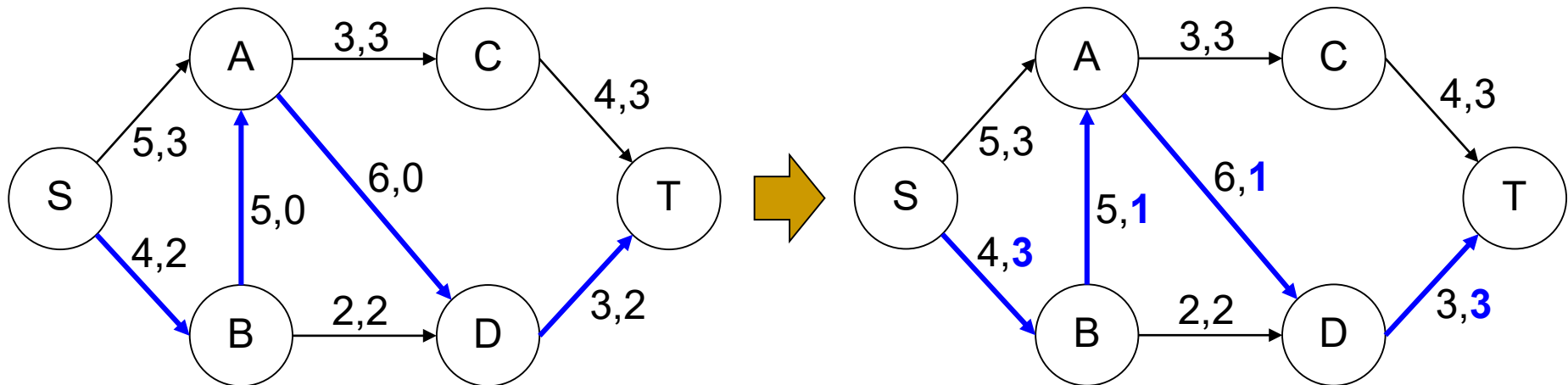


Fluxo total da rede aumentou de 5 para 6
Muito importante: as restrições são respeitadas

Melhorando uma função de fluxo

Exemplo

- Caminhos possíveis
 - (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo
 - (S,B,D,T): **infrutífero**, pois (B,D) já tem fluxo máximo
 - (S,B,A,D,T): **outro caminho possível**



Fluxo total da rede aumentou de 5 para 6
Muito importante: as restrições são respeitadas

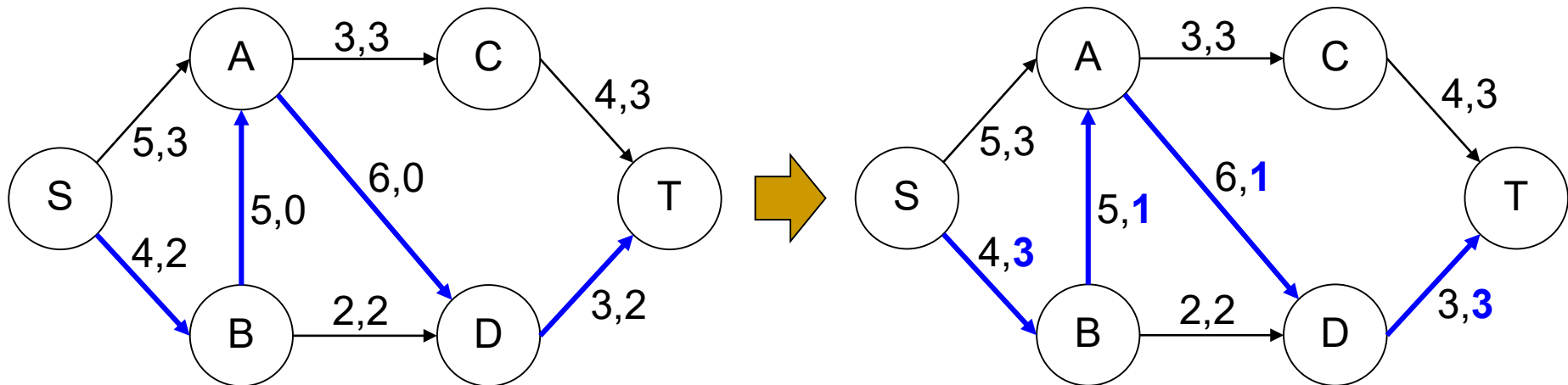
Melhorando uma função de fluxo

Exemplo

Chegamos ao
fluxo máximo!

■ Caminhos possíveis

- (S,A,C,T): **infrutífero**, pois (A,C) já tem fluxo máximo
- (S,B,D,T): **infrutífero**, pois (B,D) já tem fluxo máximo
- (S,B,A,D,T): **outro caminho possível**
- Há outro caminho que seja possível aumentar?

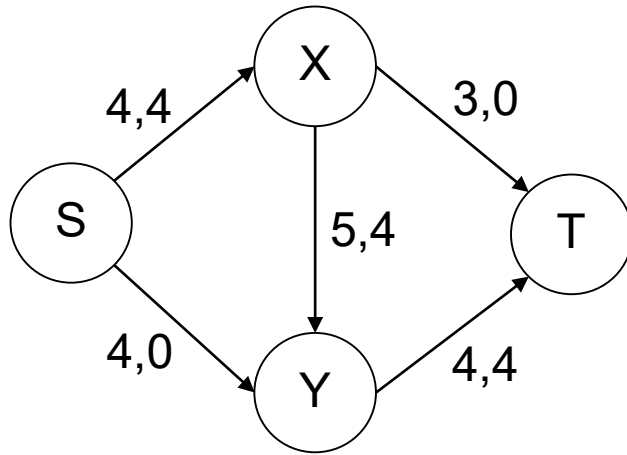


Fluxo total da rede aumentou de 5 para 6
Muito importante: as restrições são respeitadas

Melhorando uma função de fluxo

Outro Exemplo

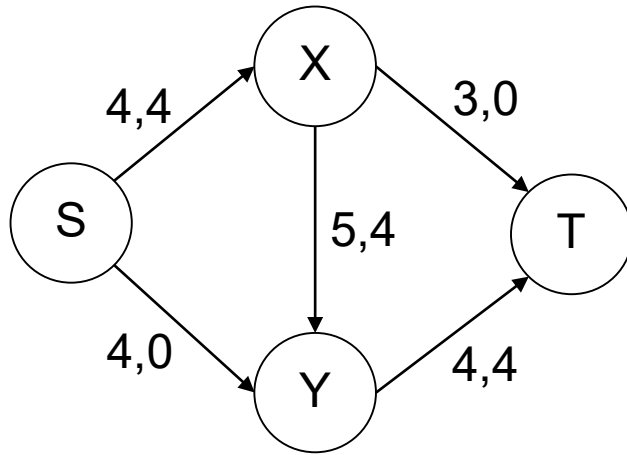
- Como aumentar o fluxo na rede abaixo?



Melhorando uma função de fluxo

Outro Exemplo

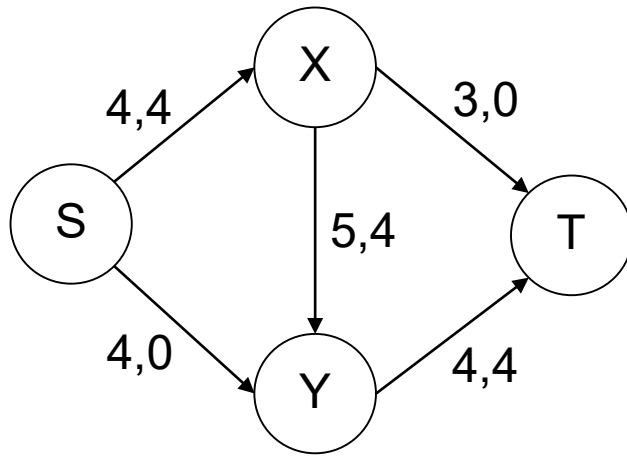
- Como aumentar o fluxo na rede abaixo?
 - Não há caminho possível de ser aumentado
 - (S,X,T) no limite possível
 - (S,X,Y,T) no limite possível
 - (S,Y,T) no limite possível



Melhorando uma função de fluxo

Outro Exemplo

- Como aumentar o fluxo na rede abaixo?
 - Não há caminho possível de ser aumentado
 - (S,X,T) no limite possível
 - (S,X,Y,T) no limite possível
 - (S,Y,T) no limite possível

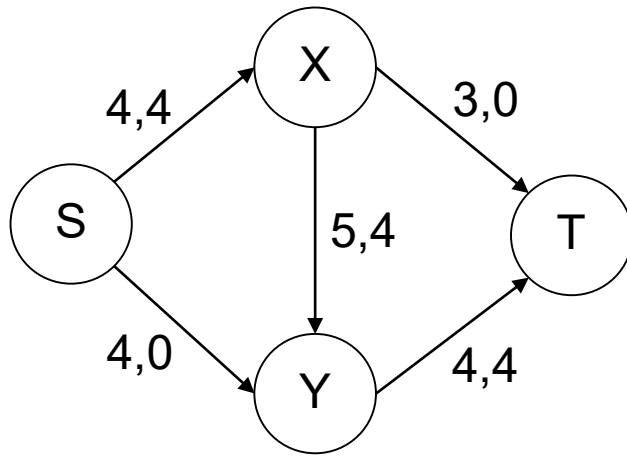


Mas, será que é mesmo impossível aumentar esse fluxo?

Melhorando uma função de fluxo

Outro Exemplo

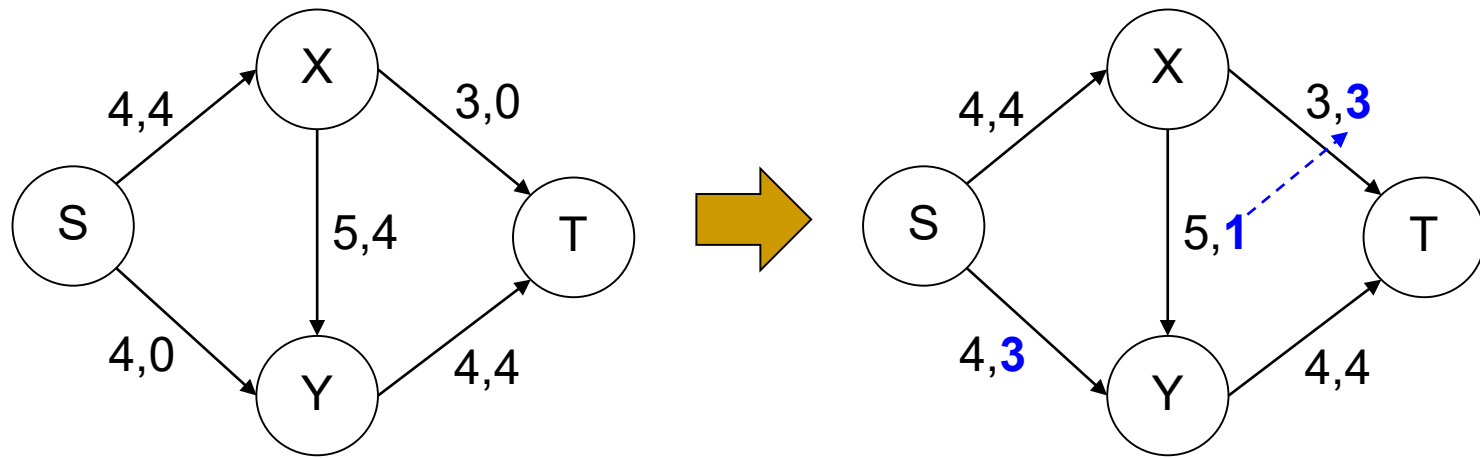
- Como aumentar o fluxo na rede abaixo?
 - Não há caminho possível de ser aumentado
 - Mas é possível redirecionar alguns fluxos e aumentar outros
 - Diminui-se o fluxo na aresta (X,Y)
 - Aumenta-se nas arestas (X,T) e (S,Y)



Melhorando uma função de fluxo

Outro Exemplo

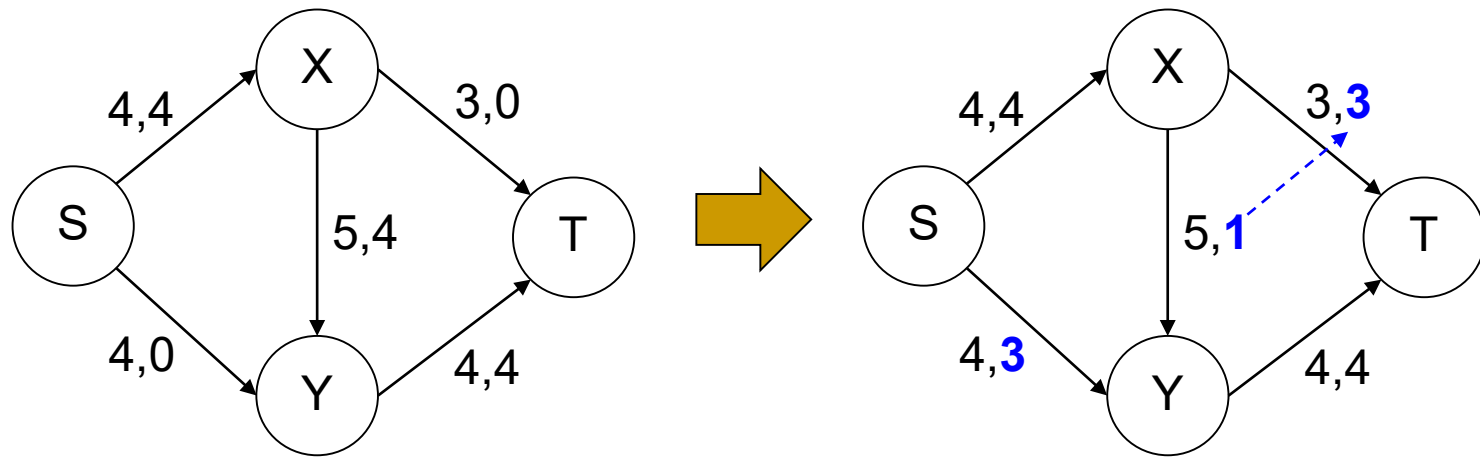
- Como aumentar o fluxo na rede abaixo?
 - Não há caminho possível de ser aumentado
 - Mas é possível redirecionar alguns fluxos e aumentar outros
 - Diminui-se o fluxo na aresta (X,Y)
 - Aumenta-se nas arestas (X,T) e (S,Y)



Melhorando uma função de fluxo

Outro Exemplo

- Como aumentar o fluxo na rede abaixo?
 - Não há caminho possível de ser aumentado
 - Mas é possível redirecionar alguns fluxos e aumentar outros
 - Diminui-se o fluxo na aresta (X,Y)
 - Aumenta-se nas arestas (X,T) e (S,Y)



Fluxo total aumenta de 4 para 7!

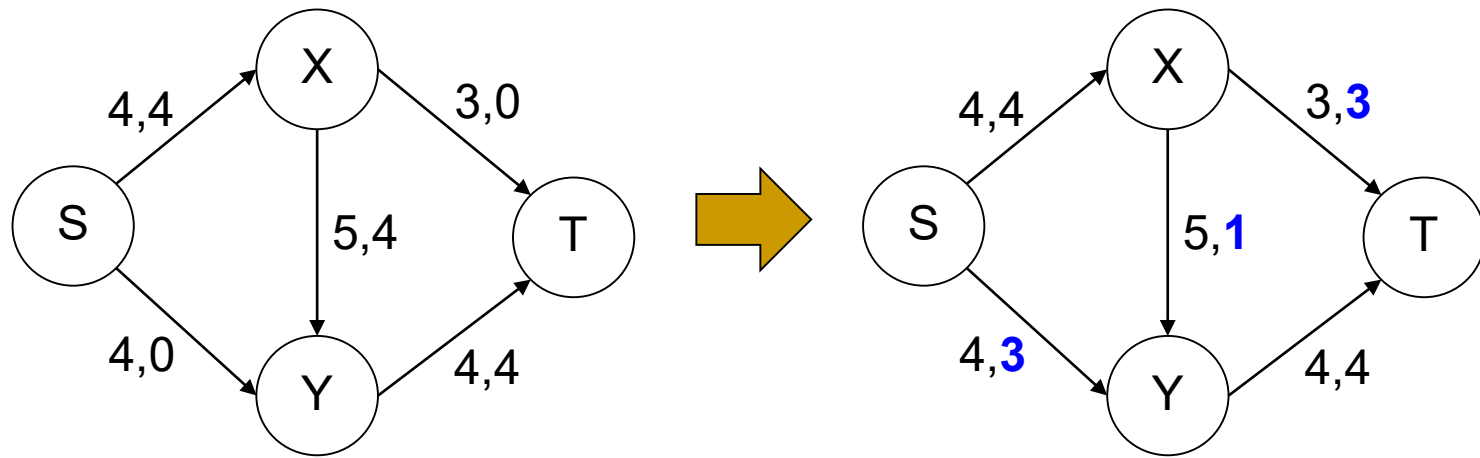
Melhorando uma função de fluxo

Outro Exemplo

■ Como aumentar o fluxo na rede abaixo?

□ Regra genérica

- Se houver (i) um caminho de S a Y, (ii) um caminho de X a T, e (iii) um caminho de X a Y com fluxo positivo, então o fluxo entre X e Y pode ser reduzido na mesma proporção em que os fluxos de S a Y e de X a T são aumentados



Fluxo total aumenta de 4 para 7!

Melhorando uma função de fluxo

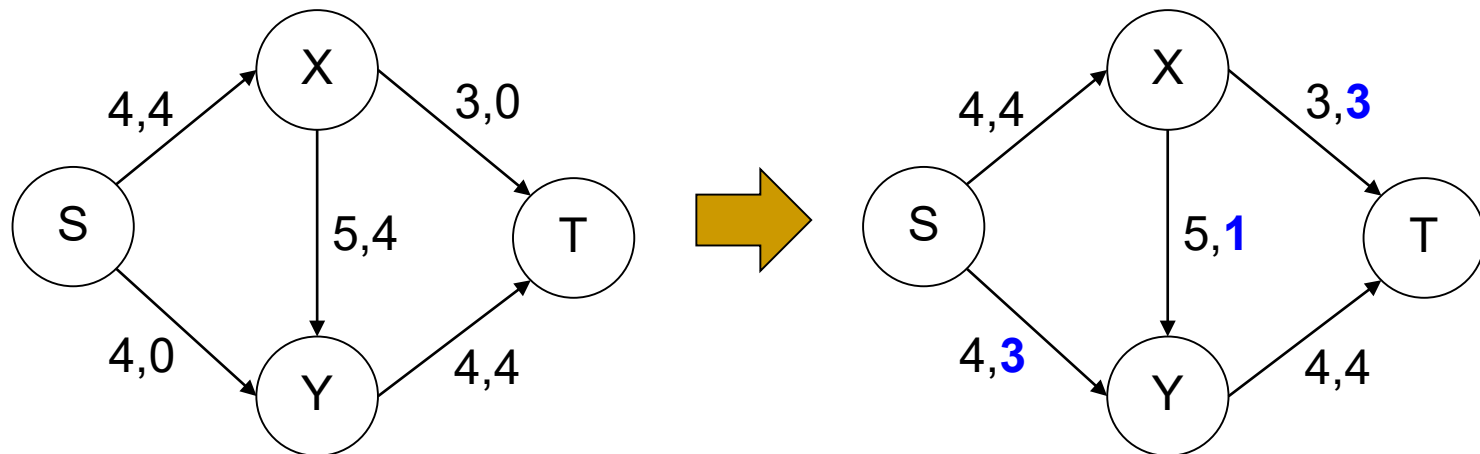
Outro Exemplo

■ Regra para aumentar o fluxo na rede abaixo

- Se houver (i) um caminho de S a Y, (ii) um caminho de X a T e (iii) um caminho de X a Y com fluxo positivo => o fluxo entre X e Y pode ser reduzido na mesma proporção em que os fluxos de S a Y e de X a T são aumentados

- $c_f(S,Y) = c(S,Y) - f(S,Y)$; $c_f(X,T) = c(X,T) - f(X,T)$

- Valor correspondente ao mínimo entre $f(X,Y)$, $c_f(S,Y)$ e $c_f(X,T)$



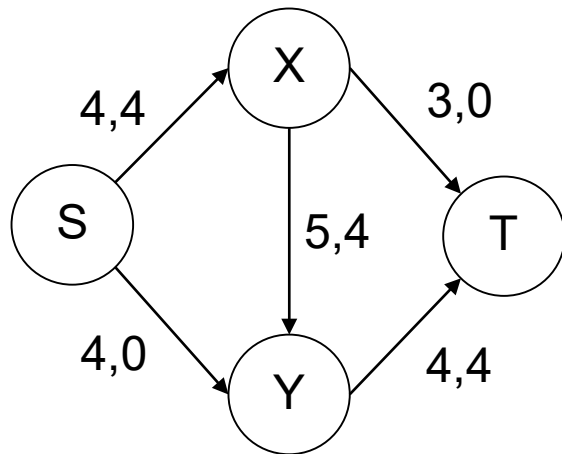
Fluxo total aumenta de 4 para 7!

Algoritmo

Melhorando uma função de fluxo

- **Estratégia:** considera um novo grafo (grafo “residual”)
- **Grafo residual** indica os caminhos possíveis com fluxo positivo (>0) “para frente” (fluxo adicional novo) e “para trás” (fluxo redirecionado)
 - todas as possibilidades de “caminhos aumentantes” para o fluxo entre S e T

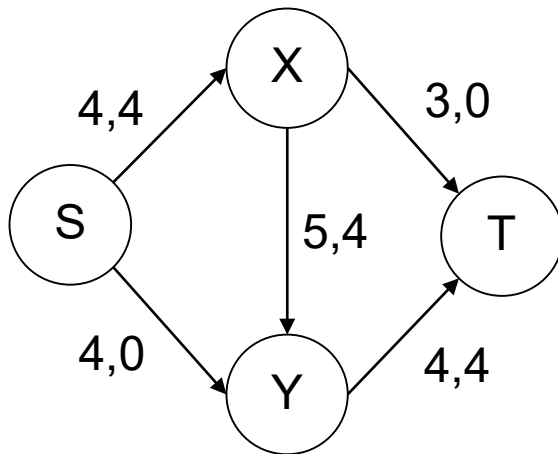
Fluxo inicial



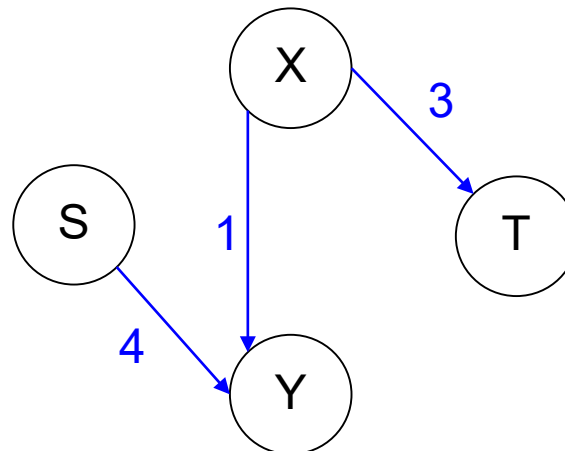
Melhorando uma função de fluxo

- **Estratégia:** considera um novo grafo (grafo “residual”)
- **Grafo residual** indica os caminhos possíveis com fluxo positivo (>0) “para frente” (fluxo adicional novo) e “para trás” (fluxo redirecionado)
 - todas as possibilidades de “caminhos aumentantes” para o fluxo entre S e T

Fluxo inicial



Grafo “residual”

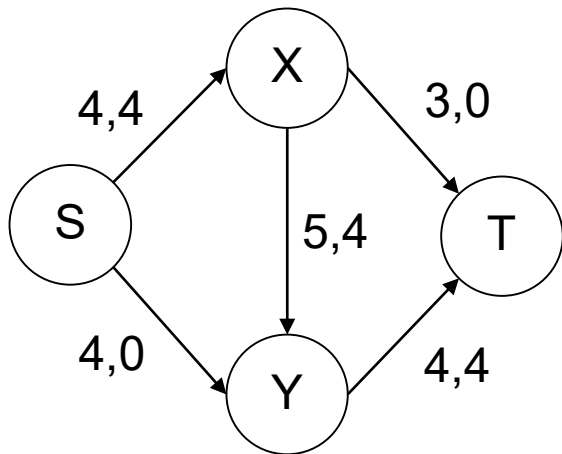


Com fluxo adicional novo possível

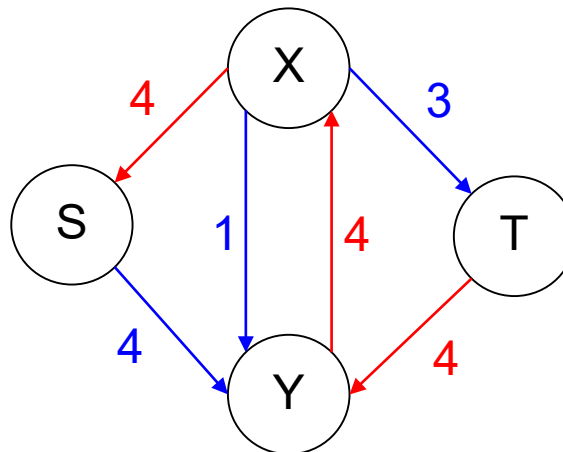
Melhorando uma função de fluxo

- **Estratégia:** considera um novo grafo (grafo “residual”)
- **Grafo residual** indica os caminhos possíveis com fluxo positivo (>0) “para frente” (fluxo adicional novo) e “para trás” (fluxo redirecionado)
 - todas as possibilidades de “caminhos aumentantes” para o fluxo entre S e T

Fluxo inicial



Grafo “residual”



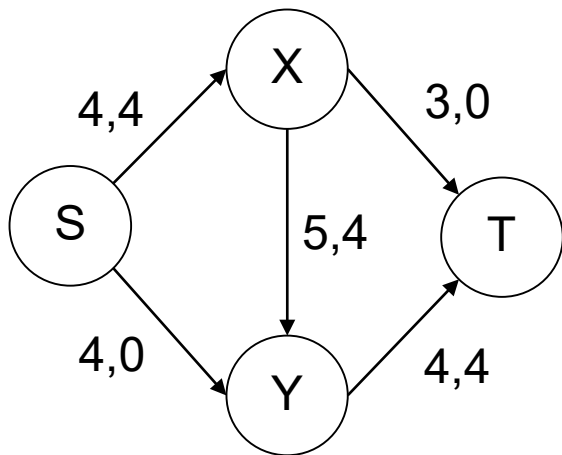
Com fluxo adicional novo possível

Com fluxo que pode ser ‘redirecionado’

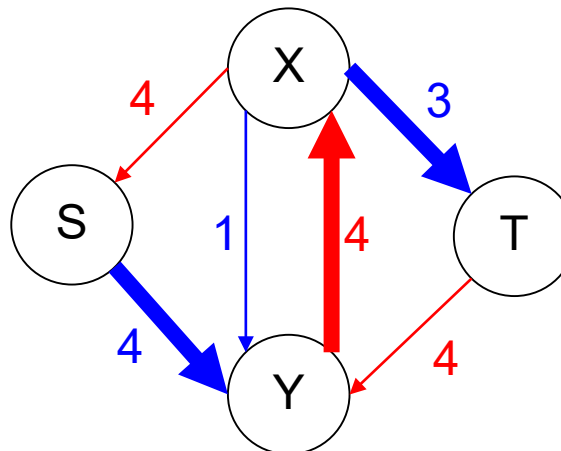
Melhorando uma função de fluxo

- **Estratégia usual:** representar no grafo os caminhos com fluxo positivo (>0) “**para frente**” (fluxo adicional novo) e “**para trás**” (fluxo redirecionado) possíveis, produzindo um grafo com todas as possibilidades de “caminhos aumentantes” do fluxo entre S e T

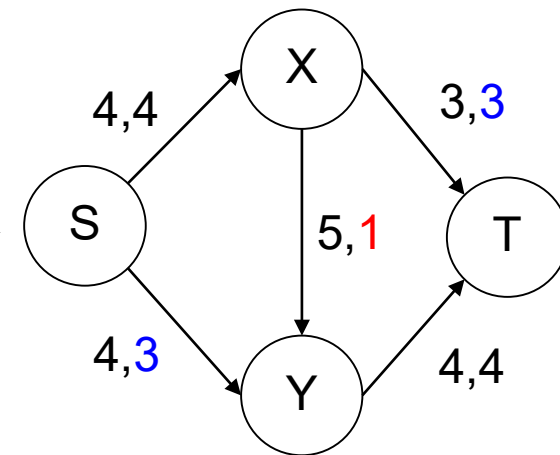
Fluxo inicial



Grafo “residual”



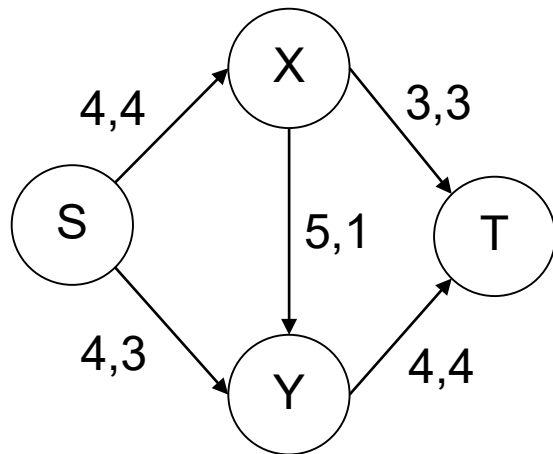
Fluxo atualizado



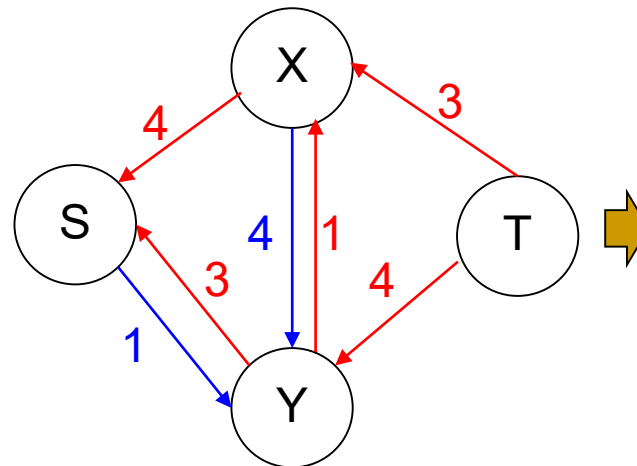
Melhorando uma função de fluxo

- **Estratégia usual:** representar no grafo os caminhos com fluxo positivo (>0) “**para frente**” (fluxo adicional novo) e “**para trás**” (fluxo redirecionado) possíveis, produzindo um grafo com todas as possibilidades de “caminhos aumentantes” do fluxo entre S e T

Fluxo atualizado



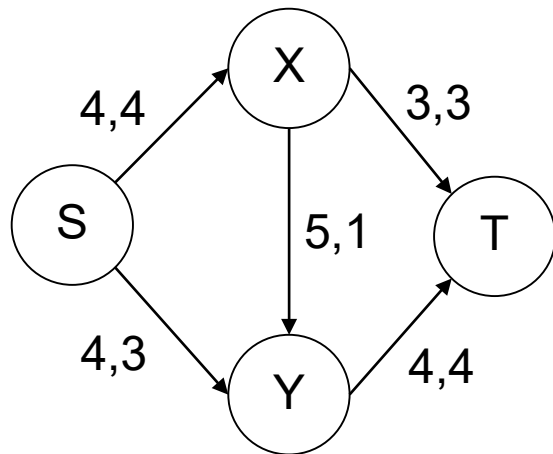
Grafo “residual” atualizado



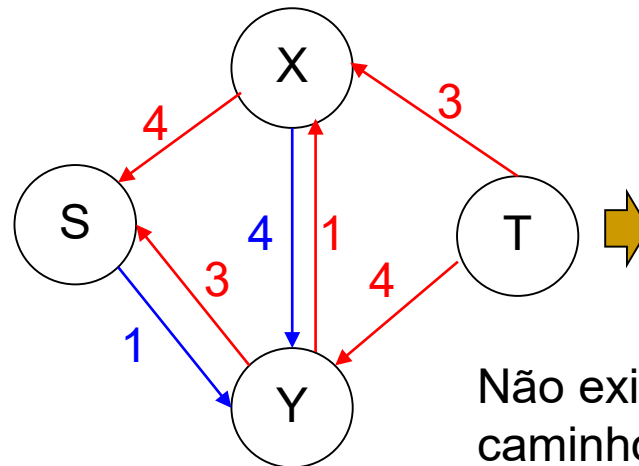
Melhorando uma função de fluxo

- **Estratégia usual:** representar no grafo os caminhos com fluxo positivo (>0) “**para frente**” (fluxo adicional novo) e “**para trás**” (fluxo redirecionado) possíveis, produzindo um grafo com todas as possibilidades de “caminhos aumentantes” do fluxo entre S e T

Fluxo atualizado



Grafo “residual” atualizado



Não existe outro caminho aumentante possível!

Algoritmo de Ford-Fulkerson

Entrada: grafo $G=(V,A)$ direcionado e ponderado, vértices S e T

para cada aresta (u,v) do grafo G , **faça**

$$f(u,v) = 0$$

construir o grafo residual G' a partir de G

enquanto existir em G' um caminho aumentante C , de S até T **faça**

 buscar aresta de peso mínimo no caminho aumentante C

 atualizar os valores de fluxo nas arestas correspondentes em G

 atualizar grafo residual G'

Algoritmo de Ford-Fulkerson

Entrada: grafo $G=(V,A)$ direcionado e ponderado, vértices S e T

para cada aresta (u,v) do grafo G , **faça**

$$f(u,v) = 0$$

construir o grafo residual G' a partir de G

enquanto existir em G' um caminho aumentante C , de S até T **faça**

 buscar aresta de peso mínimo no caminho aumentante C

 atualizar os valores de fluxo nas arestas correspondentes em G

 atualizar grafo residual G'

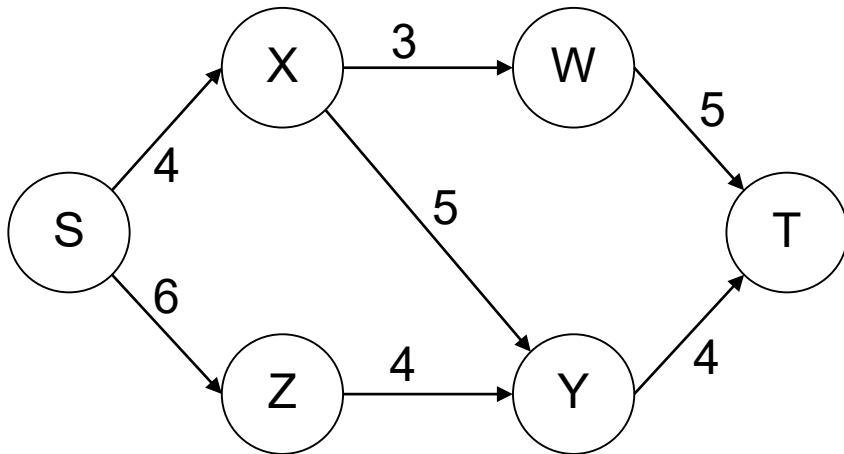
Obs.

(i) Um caminho aumentante **C** é um caminho simples de S a T no grafo residual G'

(ii) Um fluxo em G é máximo se e somente se o seu grafo residual G' não contém nenhum caminho aumentante

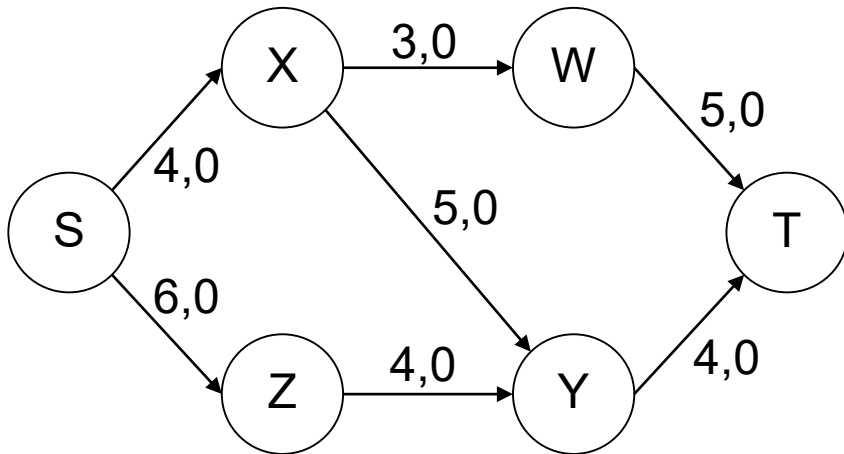
Exemplo

- Considere a rede abaixo



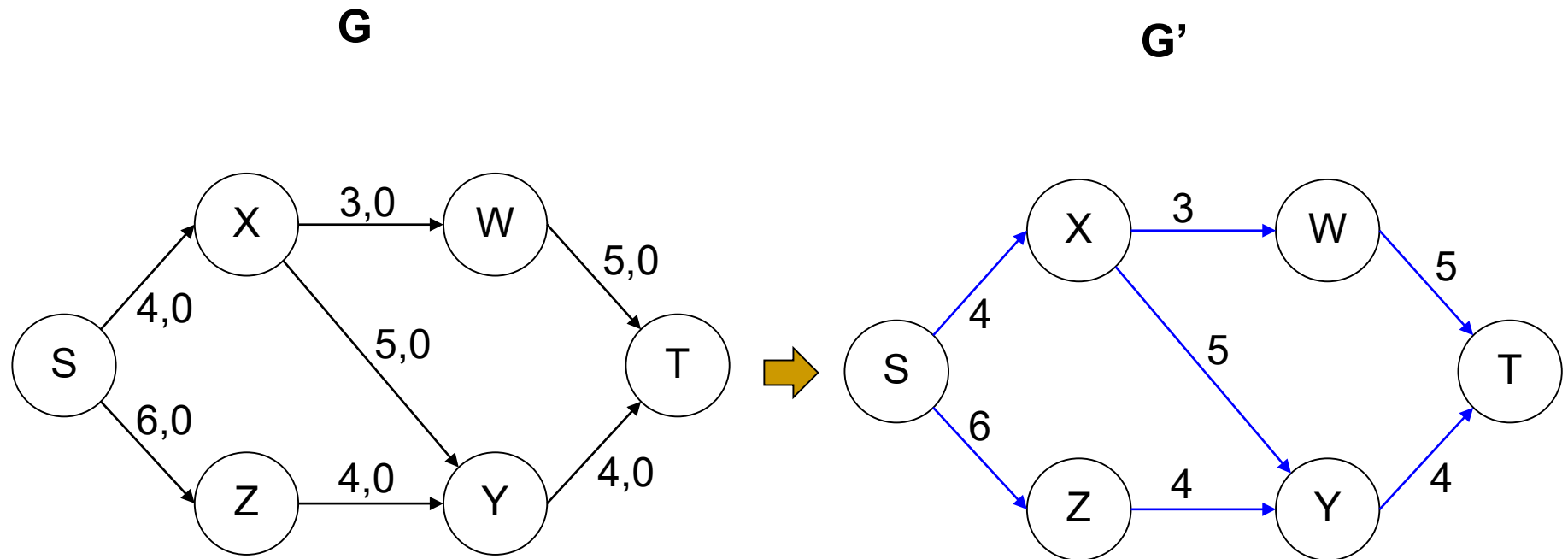
Exemplo

- Fluxos são zerados



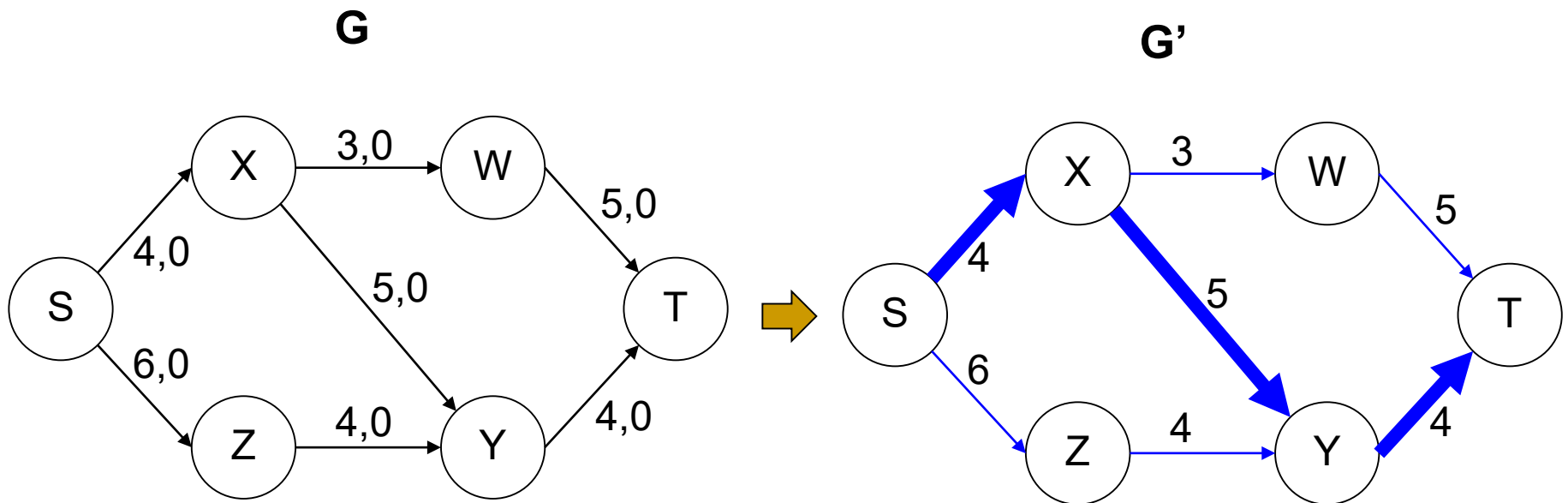
Exemplo

- Grafo residual é construído



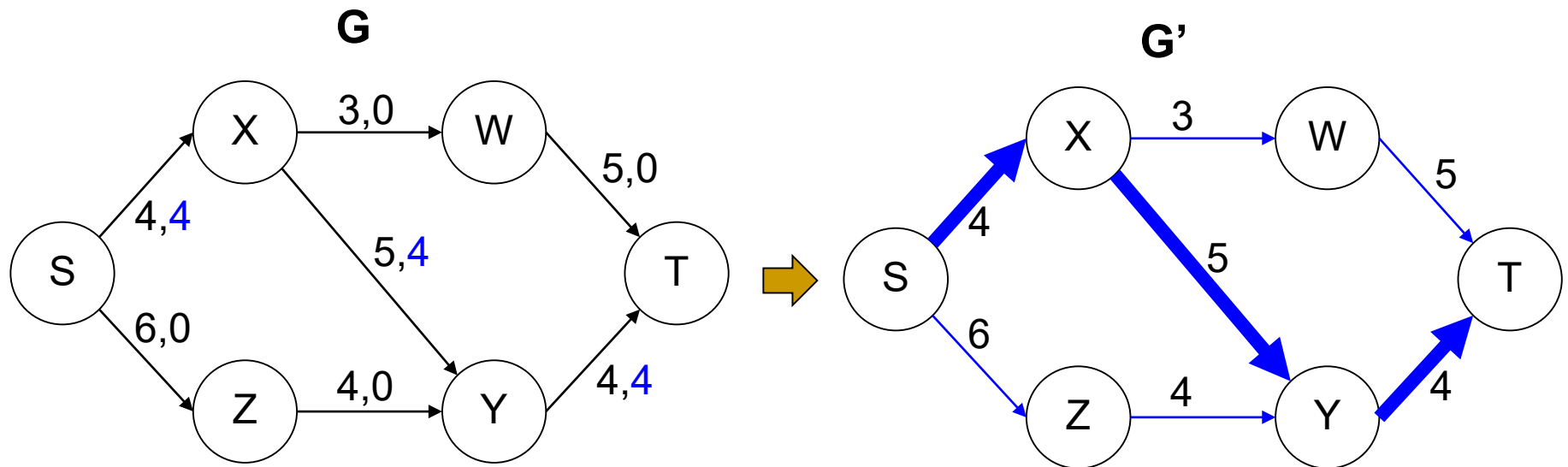
Exemplo

- Grafo residual é construído
 - Busca-se um caminho aumentante arbitrário



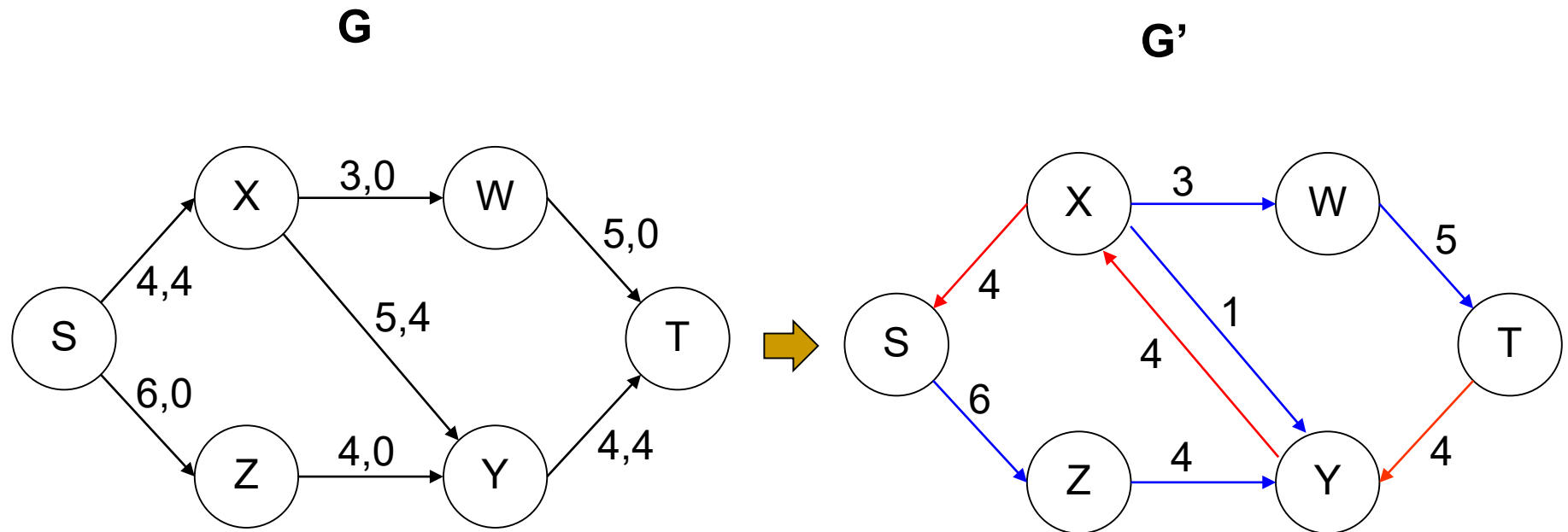
Exemplo

- Grafo residual é construído
 - Busca um caminho aumentante arbitrário
 - Busca o menor valor no caminho (=4) e atualiza os fluxos



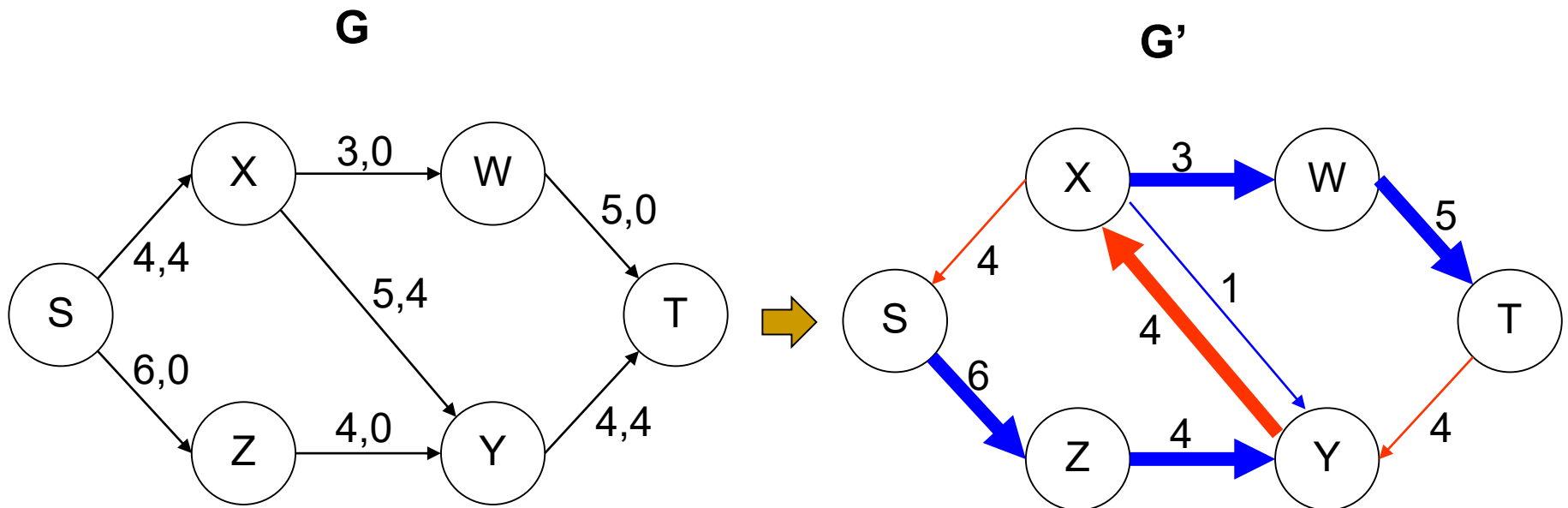
Exemplo

- Grafo residual é atualizado



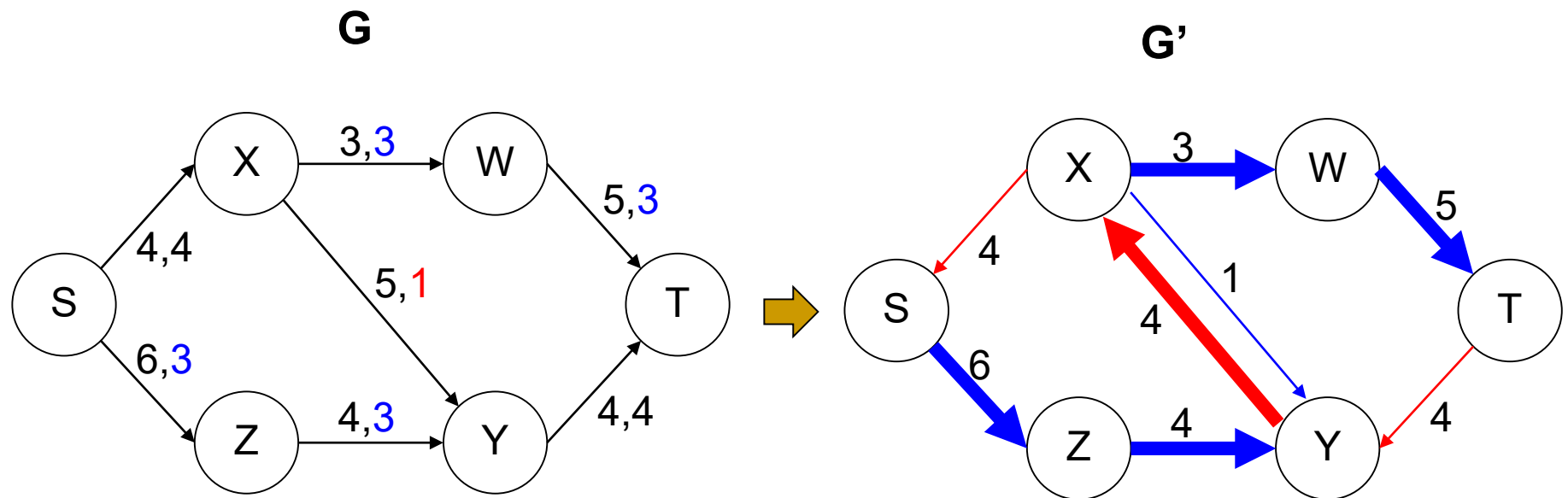
Exemplo

- Grafo residual é atualizado
 - Busca um novo caminho aumentante arbitrário



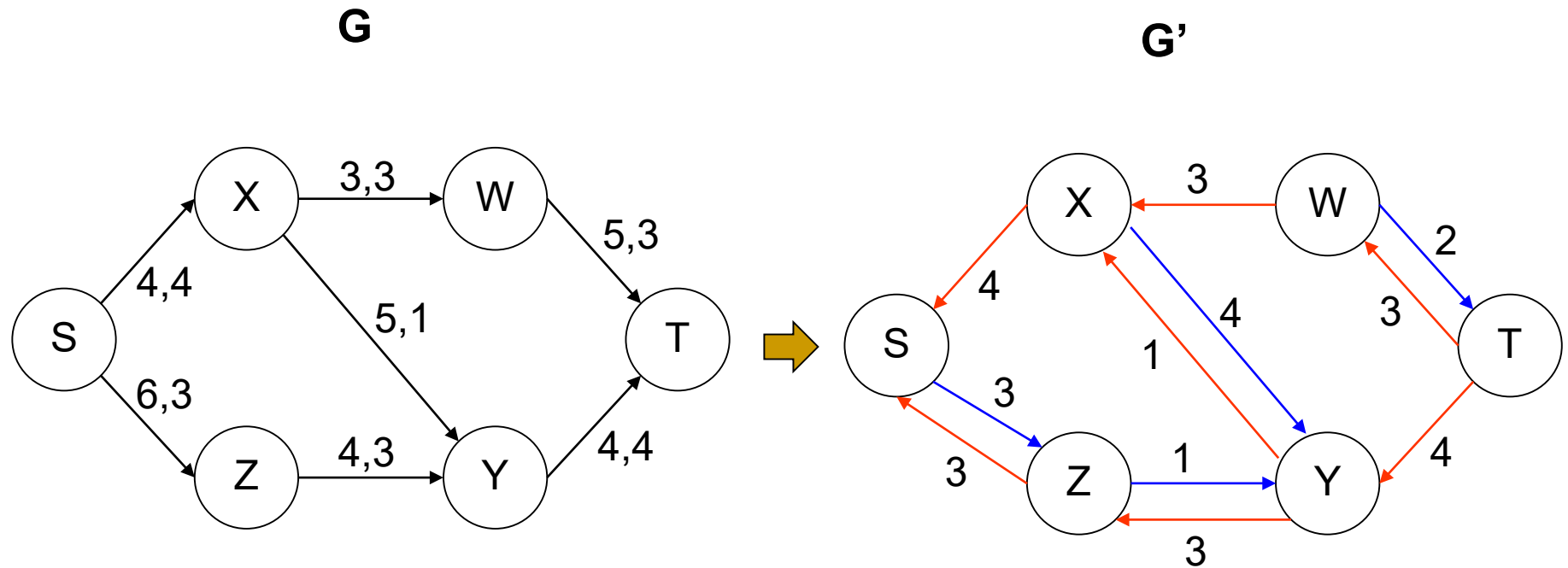
Exemplo

- Grafo residual é atualizado
 - Busca um novo caminho aumentante arbitrário
 - Busca o menor valor no caminho (=3) e atualiza os fluxos



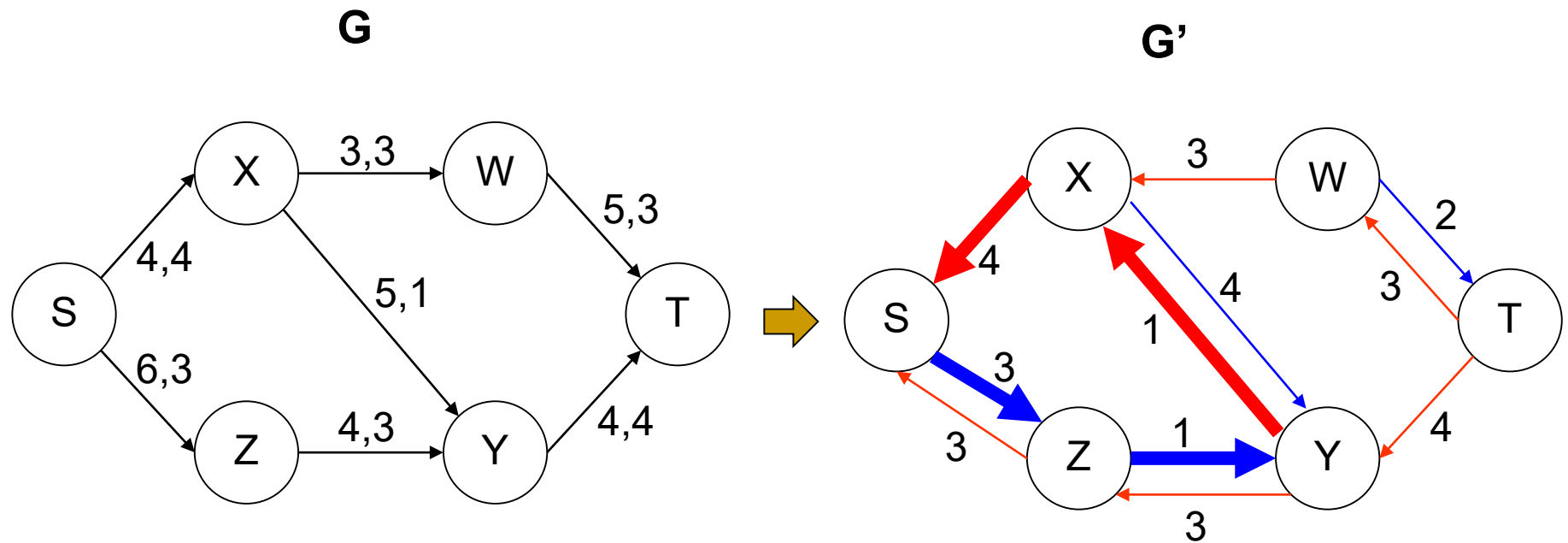
Exemplo

- Grafo residual é atualizado



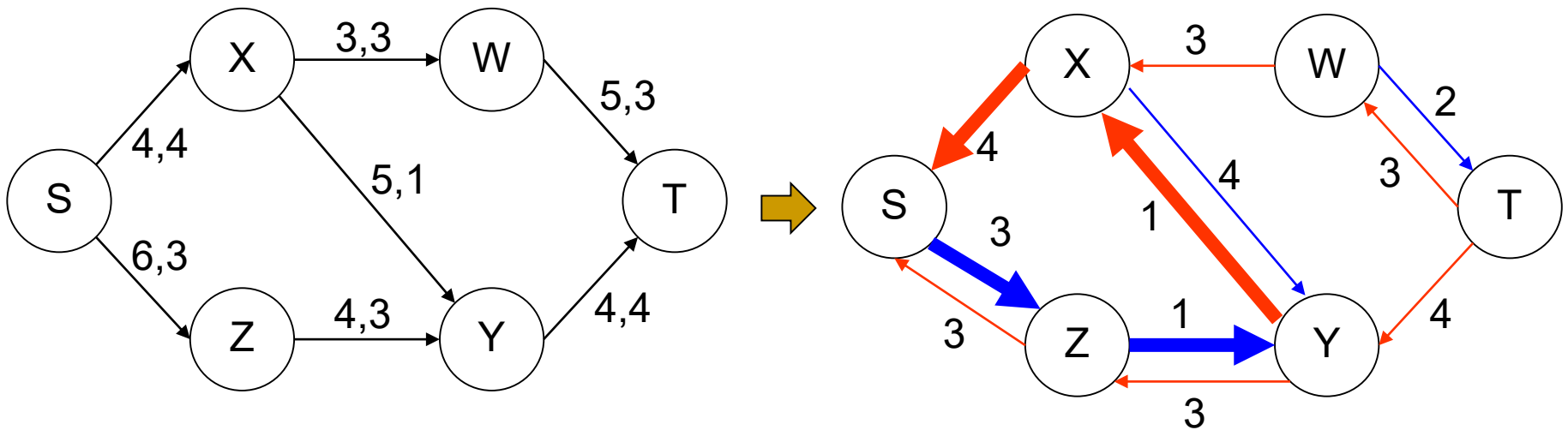
Exemplo

- Grafo residual é atualizado
 - Busca um novo caminho aumentante arbitrário



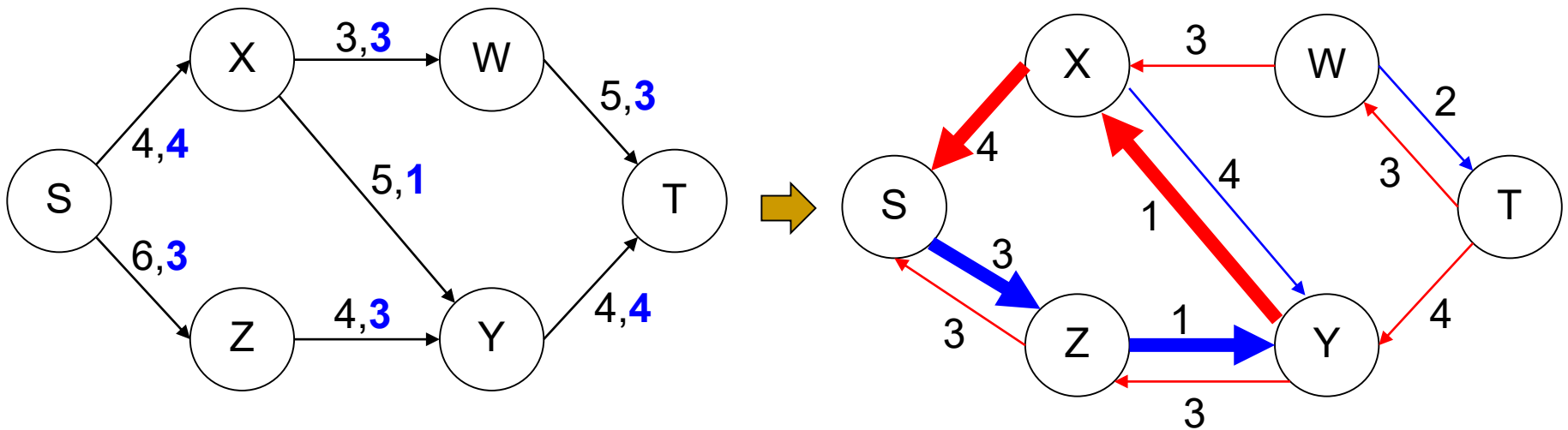
Exemplo

- Grafo residual é atualizado
 - Busca um novo caminho aumentante arbitrário
 - Não é possível encontrar um caminho de S a T 😞



Exemplo

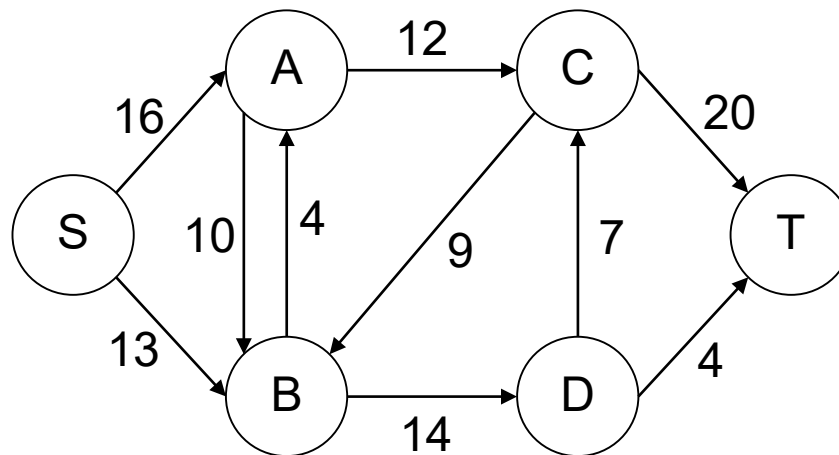
- Grafo residual é atualizado
 - Busca-se um novo caminho aumentante arbitrário
 - Não é possível encontrar um outro caminho aumentante de S a T ☹️
 - Portanto, encontramos o **fluxo máximo possível** 😊



Fluxo máximo = 7

Exercício

- Executar o algoritmo de Ford-Fulkerson sobre a rede abaixo



Algoritmo de Ford-Fulkerson

Entrada: grafo $G=(V,A)$ direcionado e ponderado, vértices S e T

para cada aresta (u,v) do grafo G , **faça**

$$f(u,v)=0$$

construir o grafo residual G' a partir de G

enquanto existir em G' um caminho aumentante C de S até T **faça**

buscar peso mínimo de aresta no caminho aumentante C

atualizar valores de fluxo nas arestas correspondentes em G

atualizar grafo residual G'

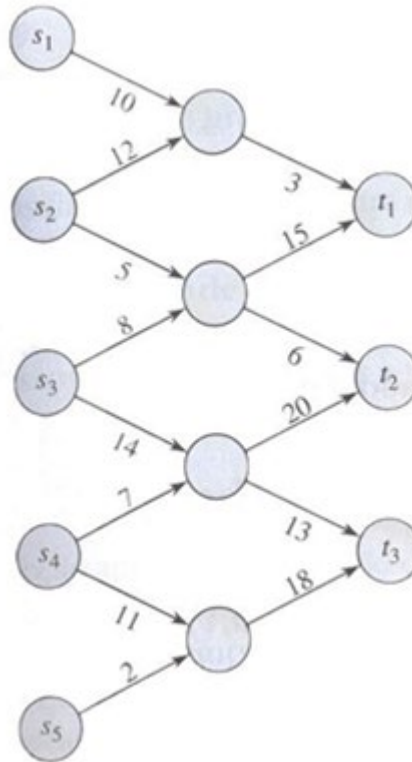
Complexidade do algoritmo?

Complexidade do algoritmo

- Supondo
 - Percurso em largura ou profundidade para encontrar os caminhos aumentantes, com $O(|V|+|A|)$
 - Pode-se considerar $O(|A|)$ em um grafo em que $|A| \gg |V|$ (usual)
 - Sendo possível aumentar o fluxo, o menor aumento possível é 1, no pior caso
 - Portanto, no pior caso, $O(|A| \cdot \text{fluxo máximo encontrado})$

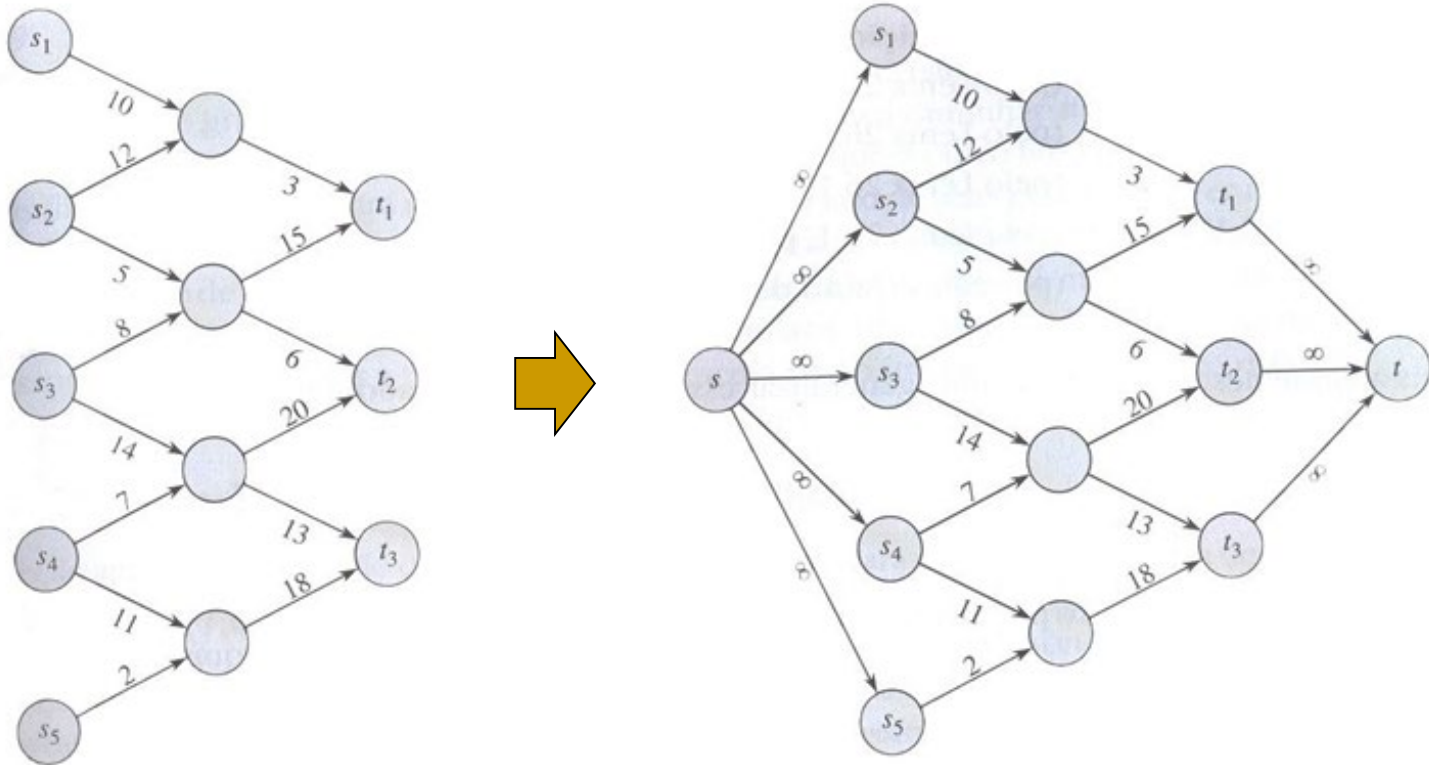
Questão

- *O que fazer em situação com várias origens e destinos?*
 - Imagine uma fábrica com várias filiais fazendo entregas variadas



Solução

- Transformamos o problema: incluímos *super-origem e superdestino*
 - As mesmas estratégias anteriores podem ser usadas



Por que isso funciona?

Desafio

- Implementar algoritmo de Ford-Fulkerson!