Problemas de Fluxo

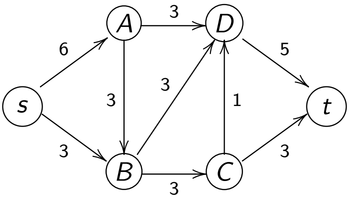
Ana Paula Tomás

Desenho e Análise de Algoritmos
Universidade do Porto

Abril 2024

Problema

Um grupo de pessoas pretende viajar de s para t. Os valores nos ramos indicam a **capacidade das ligações**, i.e., o número de lugares disponíveis em cada ligação.



Qual é o **número máximo de elementos** que o grupo pode ter se

1. **Caso 1:** o grupo não se separar?
2. **Caso 2:** o grupo se puder separar?

Como o **encaminhar** em cada caso?

Tópicos a abordar

1. Caminhos de capacidade máxima

2. Maximização de Fluxo (com fracionamento)
    - Teorema de Ford-Fulkerson
    - Rede residual e caminhos para aumento de fluxo
    - Prova do teorema de Ford-Fulkerson
    - Método de Ford-Fulkerson
    - Algoritmo de Edmonds-Karp

Caminhos de capacidade máxima (adaptação do Algoritmo de Dijkstra)

Seja $G = (V, E, c)$ um grafo dirigido, finito, $c(u, v) \geq 0$ indica a **capacidade do ramo** (u, v). A **capacidade de um percurso** é o **mínimo** das capacidades dos ramos que constituem o percurso.

Problema: Determinar um *percurso com capacidade máxima* de s para v, para cada $v \neq s$, para um nó origem s dado.

```
CaminhosCapacidadeMaxima(G, s)
  Para cada v ∈ V fazer { pai[v] ← NULL; cap[v] ← 0;}
  cap[s] ← ∞;
  Q ← Mk_PQ_HeapMax(cap, V);
  Enquanto (PQ_Not_Empty(Q)) fazer
      v ← ExtractMax(Q);
      Para cada w ∈ Adjs[v] fazer
          Se min(cap[v], c(v, w)) > cap[w] então
              cap[w] ← min(cap[v], c(v, w));
              pai[w] ← v;
              IncreaseKey(Q, w, cap[w]);
```

Complexidade temporal:

Se G for dado por **listas de adjacências** e a fila de prioridade Q for suportada por uma **heap de máximo**, tem complexidade temporal $O((|V| + |E|)\log_2 |V|)$, como o algoritmo de Dijkstra.

Caminhos de capacidade máxima (adaptação do Algoritmo de Dijkstra)

Complexidade temporal:
Se G for dado por **listas de adjacências** e a fila de prioridade Q for suportada por uma **heap de máximo**, tem complexidade temporal $O((|V| + |E|)\log_2 |V|)$, como o algoritmo de Dijkstra.

Correção (i.e., porque é que o algoritmo resolve corretamente o problema)?
O ciclo "Enquanto" preserva o **invariante** seguinte, para todo $k \geq 1$: sendo $\mathcal{Q}_k$ o conjunto de nós que estão na fila Q e $\mathcal{M}_k = V \setminus \mathcal{Q}_k$ o conjunto de nós que já sairam de Q, no final da iteração k, tem-se

1. para $r \in \mathcal{M}_k$, o valor $cap[r]$ é a capacidade máxima dos percursos de s para r em G, para todo $r \in \mathcal{M}_k$;
2. para $r \in Q_k$, o valor $cap[r]$ é a capacidade máxima dos percursos de s para r em G se os percursos só puderem passar por nós de $\mathcal{M}_k \cup \{r\}$.

Caminhos de capacidade máxima (adaptação do Algoritmo de Dijkstra)

Complexidade temporal:
Se G for dado por **listas de adjacências** e a fila de prioridade Q for suportada por uma **heap de máximo**, tem complexidade temporal $O((|V| + |E|) \log_2 |V|)$, como o algoritmo de Dijkstra.

Correção (i.e., porque é que o algoritmo resolve corretamente o problema)?
O ciclo "Enquanto" preserva o **invariante** seguinte, para todo $k \geq 1$: sendo $\mathcal{Q}_k$ o conjunto de nós que estão na fila Q e $\mathcal{M}_k = V \setminus \mathcal{Q}_k$ o conjunto de nós que já sairam de Q, no final da iteração k, tem-se

1. para $r \in \mathcal{M}_k$, o valor $cap[r]$ é a capacidade máxima dos percursos de s para r em G, para todo $r \in \mathcal{M}_k$;
2. para $r \in Q_k$, o valor $cap[r]$ é a capacidade máxima dos percursos de s para r em G se os percursos só puderem passar por nós de $\mathcal{M}_k \cup \{r\}$.

Propriedade que explora: um percurso γ_{st} de capacidade máxima **não tem de ter subestrutura ótima**. Mas, é verdade que se $\gamma_{st} = \gamma_{sv}\gamma_{vt}$, para algum v, **podemos substituir** cada um dos percursos γ_{sv} e γ_{vt} por caminhos $\gamma^\star_{sv}$ e $\gamma^\star_{vt}$ de capacidade máxima.

Caminhos de capacidade máxima em grafos não dirigidos

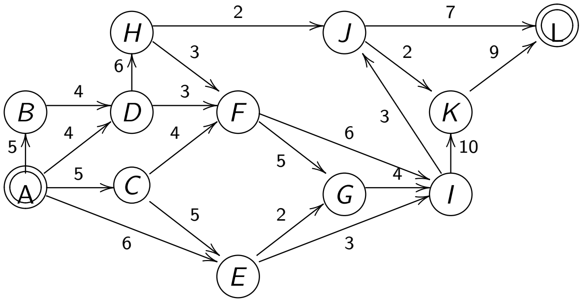
Propriedade

Se $G = (V, E, c)$ for um grafo não dirigido e conexo, a árvore geradora de **peso máximo criada a partir da raíz** s por adaptação do algoritmo de Prim contém um caminho de capacidade máxima de s para v, para cada $v \in V \setminus \{s\}$.

- Por isso, em instâncias deste tipo, o algoritmo de Prim (adaptado para obter árvores de peso máximo) seria uma alternativa ao que apresentámos acima.
- Esta propriedade resulta da definição de caminho de capacidade máxima e da seguinte propriedade estrutural das árvores de suporte de peso máximo:

  Seja T uma árvore geradora de peso **máximo** *de um grafo $G = (V, E, d)$ não dirigido e conexo. Qualquer que seja a partição $\{V_1, V_2\}$ do conjunto de vértices V, a árvore T tem algum ramo $\langle v_1, v_2\rangle$ com $v_1 \in V_1$ e $v_2 \in V_2$ e tal que $d(v_1, v_2) = \max\{d(x, y) \mid x \in V_1, y \in V_2, \langle x, y\rangle \in E\}$.*

Exemplo: Encaminhamento de chamadas



O grafo representa parte de uma rede de comunicações. Quando A recebe uma chamada, encaminha-a para L através de um conjunto de terminais de reencaminhamento. A capacidade de cada ligação está indicada na ligação. Enquanto estiver a decorrer, **uma chamada ocupa uma unidade de cada uma das linhas usadas** para a estabelecer. Não há corte das chamadas.

Quantas chamadas podem estar a decorrer no máximo num instante?

Fluxo máximo numa rede

- Rede: grafo dirigido $G = (V, A, c, \{s, t\})$ com valores nos ramos e um **nó origem** s (*source*) e um **nó destino** t (*target*).
- $c(u, v) \in \mathbb{R}_0^+$ diz-se capacidade de $(u, v) \in V \times V$.

  Assumimos que $\boxed{c(u, v) = 0 \text{ se } (u, v) \notin A}$.

Fluxo máximo numa rede

- Rede: grafo dirigido $G = (V, A, c, \{s, t\})$ com valores nos ramos e um **nó origem** s (*source*) e um **nó destino** t (*target*).
- $c(u, v) \in \mathbb{R}_0^+$ diz-se capacidade de $(u, v) \in V \times V$.

  Assumimos que $\boxed{c(u, v) = 0 \text{ se } (u, v) \notin A}$.
- Um fluxo na rede é uma função $f : V \times V \longrightarrow \mathbb{R}$ que satisfaz as condições seguintes:
  1. $f(u, v) = -f(v, u)$, para $u, v \in V$;
  2. $f(u, v) \leq c(u, v)$, para $u, v \in V$ (fluxo não excede capacidade)
  3. $\sum_{v \in V} f(u, v) = 0$, para $u \in V \setminus \{s, t\}$ (conservação do fluxo)

Fluxo máximo numa rede

- Rede: grafo dirigido $G = (V, A, c, \{s, t\})$ com valores nos ramos e um **nó origem** s (*source*) e um **nó destino** t (*target*).
- $c(u, v) \in \mathbb{R}_0^+$ diz-se capacidade de $(u, v) \in V \times V$.

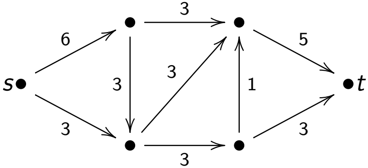
  Assumimos que $\boxed{c(u, v) = 0 \text{ se } (u, v) \notin A}$.
- Um fluxo na rede é uma função $f : V \times V \longrightarrow \mathbb{R}$ que satisfaz as condições seguintes:
  1. $f(u, v) = -f(v, u)$, para $u, v \in V$;
  2. $f(u, v) \leq c(u, v)$, para $u, v \in V$ (fluxo não excede capacidade)
  3. $\sum_{v \in V} f(u, v) = 0$, para $u \in V \setminus \{s, t\}$ (conservação do fluxo)
- O valor do fluxo f denota-se por $|f|$, e é igual ao *fluxo que sai da origem* s, sendo necessariamente igual ao *fluxo que chega ao destino* t:

$$|f| = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t).$$

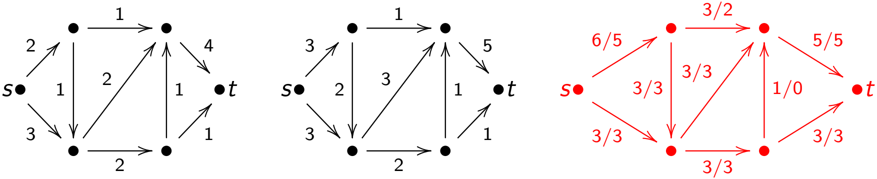
- **Problema:** determinar um **fluxo máximo** em G.

Exemplo de rede e de fluxos na rede

Rede



Três exemplos de fluxos na rede



Nos exemplos, representou-se $f(u, v)$ apenas para $(u, v) \in A$.

No exemplo à direita, colocou-se c/f nos ramos (pares capacidade/fluxo).

Definições

- Existe fluxo de u para v sse $f(u, v) > 0$
- Se $f(u, v) = c(u, v)$, o ramo (u, v) está saturado.
- Corte $\{S, T\}$ é qualquer partição $\{S, T\}$ de V tal que $s \in S$ e $t \in T$.
- Capacidade do corte $\{S, T\}$ é $c(S, T) = \sum_{u \in S, v \in T} c(u, v)$
- Corte mínimo é qualquer corte $\{S, T\}$ de capacidade mínima.
- Fluxo através do corte $\{S, T\}$ é $f(S, T) = \sum_{u \in S, v \in T} f(u, v)$

Teorema de Ford-Fulkerson

Teorema de Ford-Fulkerson (1956)

O valor do fluxo máximo numa rede é igual à capacidade do corte mínimo.

O teorema será provado à frente.

Teorema de Ford-Fulkerson

Teorema de Ford-Fulkerson (1956)

O valor do fluxo máximo numa rede é igual à capacidade do corte mínimo.

O teorema será provado à frente.

Alguns lemas úteis para a prova

- $f(X,Y) = -f(Y,X)$ e $f(X,X) = 0$, quaisquer que sejam $X, Y \subseteq V$.
- $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$, para $X, Y, Z \subseteq V$, com $X \cap Y = \emptyset$.
- Qualquer que seja o fluxo f, o fluxo através de qualquer corte $\{S, T\}$ é igual ao fluxo na rede, isto é $|f| = \sum_{u \in S, v \in T} f(u,v)$.
- $|f| \leq c(S,T)$, para todo o fluxo f e corte $\{S, T\}$, com $s \in S$ e $t \in T$.
- Se as capacidades são inteiras então existe um fluxo máximo inteiro, i.e., se $c(u,v) \in \mathbb{Z}_0^+$, existe um fluxo máximo f com $f(u,v) \in \mathbb{Z}$, para todo (u,v).

Provas dos lemas

- $f(X, Y) = -f(Y, X)$ e $f(X, X) = 0$, quaisquer que sejam $X, Y \subseteq V$.

$$f(X, Y) \stackrel{def}{=} \sum_{x \in X, y \in Y} f(x, y) = \sum_{x \in X, y \in Y} (-f(y, x)) \stackrel{def}{=} -f(Y, X)$$

Logo, se $X = Y$ tem-se $f(X, X) = -f(X, X)$. Portanto, $f(X, X) = 0$.

Provas dos lemas

- $f(X,Y) = -f(Y,X)$ e $f(X,X) = 0$, quaisquer que sejam $X, Y \subseteq V$.

$$f(X,Y) \stackrel{def}{=} \sum_{x\in X, y\in Y} f(x,y) = \sum_{x\in X, y\in Y} (-f(y,x)) \stackrel{def}{=} -f(Y,X)$$

  Logo, se $X = Y$ tem-se $f(X,X) = -f(X,X)$. Portanto, $f(X,X) = 0$.

- $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$, para $X, Y, Z \subseteq V$, com $X \cap Y = \emptyset$.
  Segue trivialmente da definição, usando $X \cap Y = \emptyset$.

Provas dos lemas

- $f(X, Y) = -f(Y, X)$ e $f(X, X) = 0$, quaisquer que sejam $X, Y \subseteq V$.

  $$f(X, Y) \stackrel{def}{=} \sum_{x \in X, y \in Y} f(x, y) = \sum_{x \in X, y \in Y} (-f(y, x)) \stackrel{def}{=} -f(Y, X)$$

  Logo, se $X = Y$ tem-se $f(X, X) = -f(X, X)$. Portanto, $f(X, X) = 0$.

- $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$, para $X, Y, Z \subseteq V$, com $X \cap Y = \emptyset$.
  Segue trivialmente da definição, usando $X \cap Y = \emptyset$.

- Para todo fluxo f e corte $\{S, T\}$ tem-se $|f| = f(S, T)$.

  $f(S, T) = f(S, T) + f(T, T) = f(S \cup T, T) = f(V, T) = |f|$ pois
  $f(V, T) = \sum_{y \in T \setminus \{t\}} f(V, \{y\}) + f(V, \{t\}) = 0 + |f| = |f|$, porque $f(V, \{y\}) = 0$ para $y \notin \{s, t\}$.

Provas dos lemas

- $f(X, Y) = -f(Y, X)$ e $f(X, X) = 0$, quaisquer que sejam $X, Y \subseteq V$.

$$f(X, Y) \stackrel{def}{=} \sum_{x \in X, y \in Y} f(x, y) = \sum_{x \in X, y \in Y} (-f(y, x)) \stackrel{def}{=} -f(Y, X)$$

  Logo, se $X = Y$ tem-se $f(X, X) = -f(X, X)$. Portanto, $f(X, X) = 0$.

- $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$, para $X, Y, Z \subseteq V$, com $X \cap Y = \emptyset$.
  Segue trivialmente da definição, usando $X \cap Y = \emptyset$.

- Para todo fluxo f e corte $\{S, T\}$ tem-se $|f| = f(S, T)$.

  $f(S, T) = f(S, T) + f(T, T) = f(S \cup T, T) = f(V, T) = |f|$ pois
  $f(V, T) = \sum_{y \in T \setminus \{t\}} f(V, \{y\}) + f(V, \{t\}) = 0 + |f| = |f|$, porque $f(V, \{y\}) = 0$ para $y \notin \{s, t\}$.

- $|f| \leq c(S, T)$ para todo o fluxo f e corte $\{S, T\}$.

  $|f| = f(S, T) = \sum_{u \in S, v \in T} f(u, v) \leq \sum_{u \in S, v \in T} c(u, v) = c(S, T)$

Rede residual associada a um fluxo

Dado um fluxo f, a **capacidade residual c_f** é definida $V \times V \to$ em $\mathbb{R}$, por

$$\boxed{c_f(u,v) = c(u,v) - f(u,v)}$$

A rede $G_f = (V, A_f, c_f, \{s,t\})$ com $A_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$ designa-se por **rede residual induzida pelo fluxo f em G**.

Um **caminho para aumento de f** é qualquer caminho de s para t em G_f.
A **capacidade residual** de um caminho para aumento γ é a capacidade residual mínima dos ramos que constituem tal caminho γ.

Rede residual associada a um fluxo

Dado um fluxo f, a **capacidade residual c_f** é definida $V \times V \to$ em $\mathbb{R}$, por

$$\boxed{c_f(u,v) = c(u,v) - f(u,v)}$$

A rede $G_f = (V, A_f, c_f, \{s,t\})$ com $A_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$ designa-se por **rede residual induzida pelo fluxo f em G**.

Um **caminho para aumento de f** é qualquer caminho de s para t em G_f.
A **capacidade residual** de um caminho para aumento γ é a capacidade residual mínima dos ramos que constituem tal caminho γ.

Da prova do teorema de Ford-Fulkerson, que veremos à frente, conclui-se que

Caraterização do fluxo máximo

O fluxo f é máximo se e só se não existe caminho de s para t na rede residual G_f.

Fluxos na rede e na rede residual

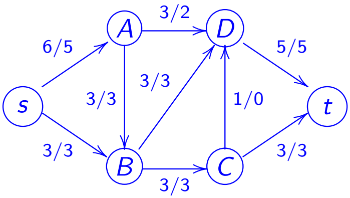
Lema (relaciona fluxos na rede e na rede residual)

Seja $G = (V, E, c, \{s, t\})$ uma rede com origem s e destino t. Seja f um fluxo em G e sejam G_f o grafo residual induzido por esse fluxo em G e f' um fluxo em G_f.

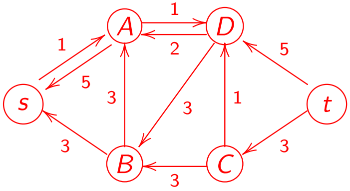
1. f' é um fluxo em G_f se e só se $f + f'$ é um fluxo em G.
2. O valor do fluxo $f + f'$ em G é $|f| + |f'|$ quaisquer que sejam $x, y \in V$ (por definição, $(f + f')(x, y) = f(x, y) + f'(x, y)$).
3. f' é um fluxo máximo em G_f se e só se $f + f'$ é um fluxo máximo em G.

Exemplo: fluxo e rede residual associada

Fluxo f



Rede residual G_f



Omitiu-se fluxo $f(x, y)$ para $(x, y) \notin A$.

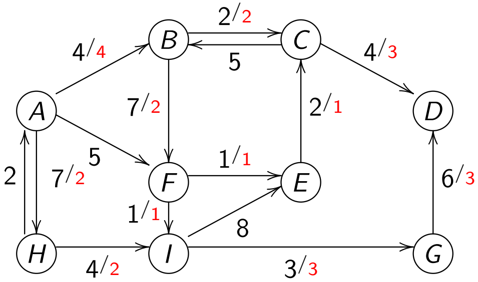
Na rede residual, $c_f(s, A) = 1$ e $c_f(A, s) = 5$ pois

$$\begin{aligned} c_f(A, s) &= c(A, s) - f(A, s) = 0 - (-f(s, A)) = 5 \\ c_f(s, A) &= c(s, A) - f(s, A) = 6 - 5 = 1 \end{aligned}$$

O fluxo f é máximo porque não há caminho de s para t na rede residual G_f.
Exemplo de corte $\{S, T\}$ mínimo: $S = \{s, A, D, B\}$, $T = \{C, t\}$ com capacidade $c(D, t) + c(B, C) = 8 = |f|$.

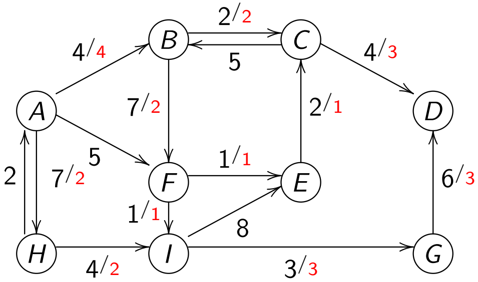
Exemplo: Como é a rede residual? O fluxo é máximo?

Os valores indicados a vermelho representam o fluxo no ramo. A origem da rede é o nó A e o destino é o nó D. Não estão indicados valores de $f(x, y) \leq 0$.



Exemplo: Como é a rede residual? O fluxo é máximo?

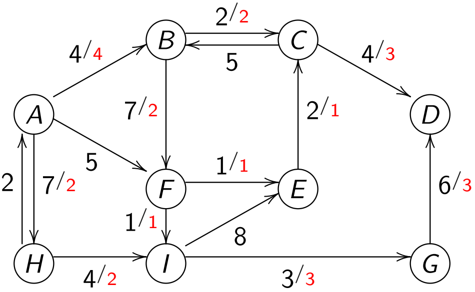
Os valores indicados a vermelho representam o fluxo no ramo. A origem da rede é o nó A e o destino é o nó D. Não estão indicados valores de $f(x, y) \leq 0$.



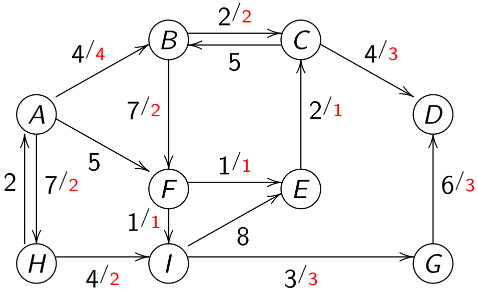
$f(A, H) =?$ $c(A, H) =?$ $c_f(A, H) =?$ $f(H, A) =?$ $c(H, A) =?$ $c_f(H, A) =?$
$f(G, I) =?$ $c(G, I) =?$ $c_f(G, I) =?$
$f(I, E) =?$ $f(E, I) =?$ $c_f(E, I) =?$ $c_f(I, E) =?$

Observar a **conservação de fluxo nos nós internos e na rede** . Quanto é $|f|$?

Como é a rede residual? O fluxo é máximo?



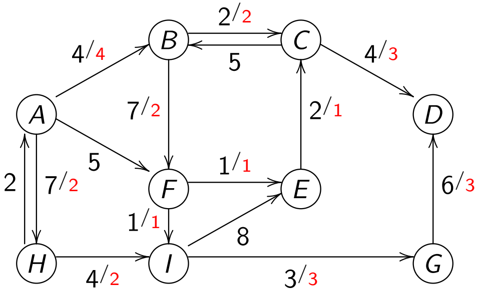
Como é a rede residual? O fluxo é máximo?



$$
\begin{array}{lllll}
|f| = 6 & f(A,H) = 2 & c(A,H) = 7 & c_f(A,H) = 5 & \\
 & f(H,A) = -2 & c(H,A) = 2 & c_f(H,A) = 4 & \\
 & f(G,I) = -3 & c(G,I) = 0 & c_f(G,I) = 3 & \\
 & f(I,E) = 0 & f(E,I) = 0 & c_f(E,I) = 0 & c_f(I,E) = 8
\end{array}
$$

Como é a rede residual G_f? Notar que G_f contém (x,y) sse $c_f(x,y) > 0$.

Como é a rede residual? O fluxo é máximo?



$|f| = 6 \quad f(A,H) = 2 \quad c(A,H) = 7 \quad c_f(A,H) = 5$

$f(H,A) = -2 \quad c(H,A) = 2 \quad c_f(H,A) = 4$

$f(G,I) = -3 \quad c(G,I) = 0 \quad c_f(G,I) = 3$

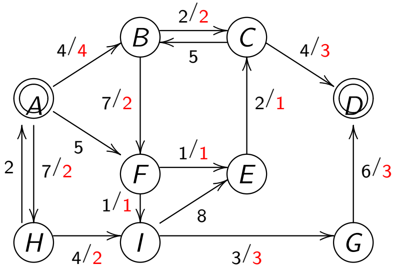
$f(I,E) = 0 \quad f(E,I) = 0 \quad c_f(E,I) = 0 \quad c_f(I,E) = 8$

Como é a rede residual G_f? Notar que G_f contém (x,y) sse $c_f(x,y) > 0$.

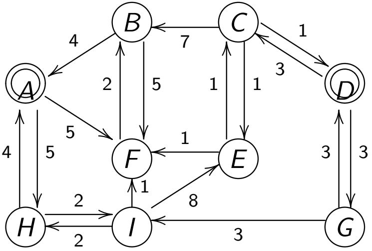
O fluxo f é máximo? Nao é máximo sse existir caminho para aumento em G_f. Verificar!

Como é a rede residual? O fluxo é máximo?

Fluxo f

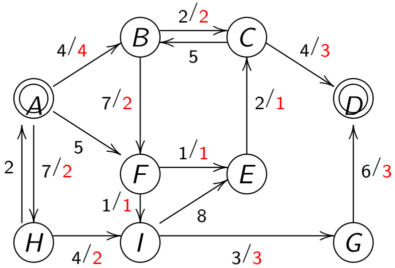


Rede residual G_f

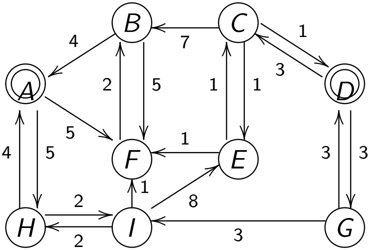


Como é a rede residual? O fluxo é máximo?

Fluxo f



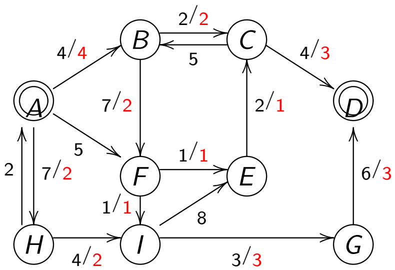
Rede residual G_f



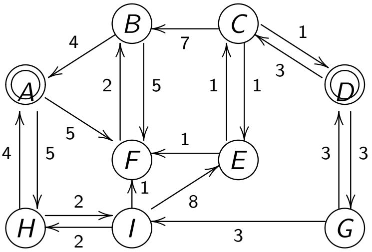
O fluxo não é máximo pois existe caminho da origem para o destino na rede residual: $\gamma = (A, H, I, E, C, D)$, com $c_\gamma = 1$. O fluxo pode aumentar de c_γ.

Como é a rede residual? O fluxo é máximo?

Fluxo f



Rede residual G_f



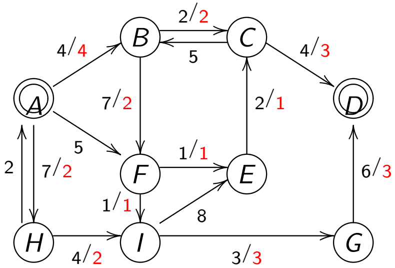
O fluxo não é máximo pois existe caminho da origem para o destino na rede residual: $\gamma = (A, H, I, E, C, D)$, com $c_\gamma = 1$. O fluxo pode aumentar de c_γ.

Alteração do fluxo ao longo de γ (restantes valores de f não mudam):

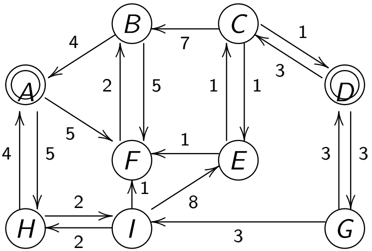
$f(A,H) = 2 + c_\gamma = 3$, $f(H,A) = -3$; $f(H,I) = 2 + c_\gamma = 3$, $f(I,H) = -3$;
$f(I,E) = 0 + c_\gamma = 1$, $f(E,I) = -1$; $f(E,C) = 1 + c_\gamma = 2$, $f(C,E) = -2$;
$f(C,D) = 3 + c_\gamma = 4$, $f(D,C) = -4$.

Como é a rede residual? O fluxo é máximo?

Fluxo f



Rede residual G_f



O fluxo não é máximo pois existe caminho da origem para o destino na rede residual: $\gamma = (A, H, I, E, C, D)$, com $c_\gamma = 1$. O fluxo pode aumentar de c_γ.

Alteração do fluxo ao longo de γ (restantes valores de f não mudam):
$f(A,H) = 2 + c_\gamma = 3$, $f(H,A) = -3$; $f(H,I) = 2 + c_\gamma = 3$, $f(I,H) = -3$;
$f(I,E) = 0 + c_\gamma = 1$, $f(E,I) = -1$; $f(E,C) = 1 + c_\gamma = 2$, $f(C,E) = -2$;
$f(C,D) = 3 + c_\gamma = 4$, $f(D,C) = -4$.
Se voltarmos a calcular a rede residual, já não há caminho em G_f. $|f| = 7$ é máximo.

Prova do Teorema de Ford-Fulkerson

Teorema de Ford-Fulkerson (1956)

O valor do fluxo máximo numa rede é igual à capacidade do corte mínimo.

O teorema de Ford-Fulkerson resulta da prova de que **as três condições seguintes são equivalentes**:

1. f é um fluxo máximo em G.
2. Na rede residual G_f não existe caminho de s para t.
3. $|f| = c(S, T)$, para algum corte (S, T) de G, com $s \in S$ e $t \in T$.

Prova do Teorema de Ford-Fulkerson

Teorema de Ford-Fulkerson (1956)

O valor do fluxo máximo numa rede é igual à capacidade do corte mínimo.

O teorema de Ford-Fulkerson resulta da prova de que **as três condições seguintes são equivalentes**:

1. f é um fluxo máximo em G.
2. Na rede residual G_f não existe caminho de s para t.
3. $|f| = c(S, T)$, para algum corte (S, T) de G, com $s \in S$ e $t \in T$.

Para provar as três equivalências basta provar três implicações:

$$
\begin{tikzcd}
\text{f é fluxo máximo} \arrow[dr, Rightarrow] & \\
\begin{array}{l}|f| = c(S,T)\\ \text{para algum corte } \{S,T\}\end{array} \arrow[u, Rightarrow] & \begin{array}{l}\text{não existe caminho}\\ \text{para aumento de f em } G_f\end{array} \arrow[l, Rightarrow]
\end{tikzcd}
$$

Prova do Teorema de Ford-Fulkerson

Prova de que $\boxed{1} \Rightarrow \boxed{2} \Rightarrow \boxed{3} \Rightarrow \boxed{1}$

1. *f é um fluxo máximo em G.*
2. *Na rede residual G_f não existe caminho de s para t.*
3. $|f| = c(S, T)$, *para algum corte* (S, T) *de* G, *com* $s \in S$ *e* $t \in T$.

$\boxed{1} \Rightarrow \boxed{2}$ Se existisse caminho de s para t em G_f, então um fluxo f' ao longo desse caminho aumentaria $|f|$ de $|f'|$ (ver os lemas anteriores).

Prova do Teorema de Ford-Fulkerson

Prova de que $\boxed{1} \Rightarrow \boxed{2} \Rightarrow \boxed{3} \Rightarrow \boxed{1}$

1. *f é um fluxo máximo em G.*
2. *Na rede residual G_f não existe caminho de s para t.*
3. *$|f| = c(S, T)$, para algum corte (S, T) de G, com $s \in S$ e $t \in T$.*

$\boxed{1} \Rightarrow \boxed{2}$ Se existisse caminho de s para t em G_f, então um fluxo f' ao longo desse caminho aumentaria $|f|$ de $|f'|$ (ver os lemas anteriores).

$\boxed{2} \Rightarrow \boxed{3}$ Sejam $S = \{$nós acessíveis de s em $G_f\}$ e $T = V \setminus S$ (notar que $t \in T$). Os ramos que existiam de S para T na rede inicial, não existem em G_f. Isso quer dizer que f os saturou. Logo, $c(S, T) = |f|$.

Prova do Teorema de Ford-Fulkerson

Prova de que $\boxed{1} \Rightarrow \boxed{2} \Rightarrow \boxed{3} \Rightarrow \boxed{1}$

1. *f é um fluxo máximo em G.*
2. *Na rede residual G_f não existe caminho de s para t.*
3. *$|f| = c(S,T)$, para algum corte (S,T) de G, com $s \in S$ e $t \in T$.*

$\boxed{1} \Rightarrow \boxed{2}$ Se existisse caminho de s para t em G_f, então um fluxo f' ao longo desse caminho aumentaria $|f|$ de $|f'|$ (ver os lemas anteriores).

$\boxed{2} \Rightarrow \boxed{3}$ Sejam $S = \{$nós acessíveis de s em $G_f\}$ e $T = V \setminus S$ (notar que $t \in T$). Os ramos que existiam de S para T na rede inicial, não existem em G_f. Isso quer dizer que f os saturou. Logo, $c(S,T) = |f|$.

$\boxed{3} \Rightarrow \boxed{1}$ Como $|f| = c(S,T)$, então $|f|$ é máximo pois nenhum fluxo excede a capacidade de um corte $\{S,T\}$ da rede, qualquer que seja o corte $\{S,T\}$ (cf, lemas). Assim, se $|f| = c(S,T)$, então $\{S,T\}$ tem capacidade mínima. Logo, $|f|$ é máximo.

Prova de que $\boxed{1} \Rightarrow \boxed{2} \Rightarrow \boxed{3} \Rightarrow \boxed{1}$

1. *f é um fluxo máximo em G.*
2. *Na rede residual G_f não existe caminho de s para t.*
3. *$|f| = c(S, T)$, para algum corte (S, T) de G, com $s \in S$ e $t \in T$.*

$\boxed{1} \Rightarrow \boxed{2}$ Se existisse caminho de s para t em G_f, então um fluxo f' ao longo desse caminho aumentaria $|f|$ de $|f'|$ (ver os lemas anteriores).

$\boxed{2} \Rightarrow \boxed{3}$ Sejam $S = \{$nós acessíveis de s em $G_f\}$ e $T = V \setminus S$ (notar que $t \in T$). Os ramos que existiam de S para T na rede inicial, não existem em G_f. Isso quer dizer que f os saturou. Logo, $c(S, T) = |f|$.

$\boxed{3} \Rightarrow \boxed{1}$ Como $|f| = c(S, T)$, então $|f|$ é máximo pois nenhum fluxo excede a capacidade de um corte $\{S, T\}$ da rede, qualquer que seja o corte $\{S, T\}$ (cf, lemas). Assim, se $|f| = c(S, T)$, então $\{S, T\}$ tem capacidade mínima. Logo, $|f|$ é máximo.

NB: a prova de $\boxed{2} \Rightarrow \boxed{3}$ indica como encontrar um corte $\{S, T\}$ mínimo, i.e., tal que $|f| = c(S, T)$.

Método de Ford-Fulkerson para determinar fluxo máximo

Da prova resulta um método para cálculo do fluxo máximo, que termina sempre para capacidades $\in \mathbb{Q}_0^+$.

Metodo_Ford-Fulkerson(G)

Para cada $(u, v) \in G.A$ fazer $f(u, v) \leftarrow 0$; $f(v, u) \leftarrow 0$;
Determinar a rede residual G_f; /* neste caso $G_f = G$ pois f é nulo */
Enquanto existir um caminho γ de s para t em G_f fazer:
  $c_\gamma \leftarrow \min\{c_f(u, v) \mid (u, v) \in \gamma\}$;
  Para cada $(u, v) \in \gamma$ fazer
    $f(u, v) \leftarrow f(u, v) + c_\gamma$;
    $f(v, u) \leftarrow -f(u, v)$;
  Actualizar G_f; /* afeta apenas os ramos de γ e simétricos */

NB: em vez de partir de f nulo, podemos **partir de um fluxo f já conhecido**, **calcular G_f, e prosseguir...**

Método de Ford-Fulkerson para determinar fluxo máximo

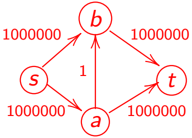
- O método de Ford-Fulkerson não indica como se escolhe γ.

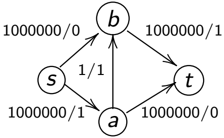
Método de Ford-Fulkerson para determinar fluxo máximo

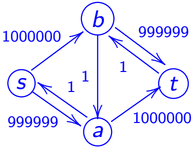
- O método de Ford-Fulkerson não indica como se escolhe γ. Tal causa problemas.
  - Prova-se que termina sempre para capacidades $\in \mathbb{Z}_0^+$ (e $\mathbb{Q}_0^+$) mas **pode não terminar** se forem irracionais. $\therefore$ Não é *algoritmo*.

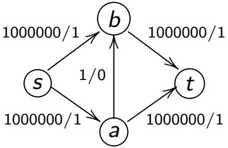
Método de Ford-Fulkerson para determinar fluxo máximo

- O método de Ford-Fulkerson não indica como se escolhe γ. Tal causa problemas.
  - Prova-se que termina sempre para capacidades $\in \mathbb{Z}_0^+$ (e $\mathbb{Q}_0^+$) mas **pode não terminar** se forem irracionais. $\therefore$ Não é *algoritmo*.
  - O número de iterações pode ser exponencial no tamanho da instância.



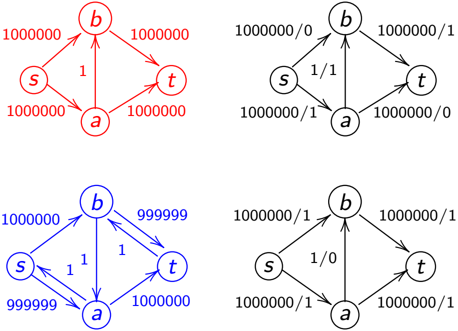






Método de Ford-Fulkerson para determinar fluxo máximo

- O método de Ford-Fulkerson não indica como se escolhe γ. Tal causa problemas.
  - Prova-se que termina sempre para capacidades $\in \mathbb{Z}_0^+$ (e $\mathbb{Q}_0^+$) mas **pode não terminar** se forem irracionais. $\therefore$ Não é *algoritmo*.
  - O número de iterações pode ser exponencial no tamanho da instância.



Se escolher alternadamente os caminhos (s, a, b, t) e (s, b, a, t) são necessárias 2000000 **iterações**. Mas, **bastam duas se for** (s, a, t) **e** (s, b, t).

Método de Ford-Fulkerson para determinar fluxo máximo

Para capacidades inteiras...

- Complexidade de cada iteração é $O(m)$, sendo $|A| = m$.
- O fluxo aumenta de uma unidade pelo menos em cada iteração. Portanto, o número de iterações é menor ou igual ao fluxo ótimo $|f^*|$.
- Complexidade do método é $\mathcal{O}(m^2\,C)$, onde C é capacidade máxima dos ramos. Notar que é $O(m|f^*|)$ e $|f^*| \leq mC$.

Método de Ford-Fulkerson para determinar fluxo máximo

Para capacidades inteiras...

- Complexidade de cada iteração é $O(m)$, sendo $|A| = m$.
- O fluxo aumenta de uma unidade pelo menos em cada iteração. Portanto, o número de iterações é menor ou igual ao fluxo ótimo $|f^*|$.
- Complexidade do método é $\mathcal{O}(m^2\, C)$, onde C é capacidade máxima dos ramos. Notar que é $O(m|f^*|)$ e $|f^*| \leq mC$. De facto, $|f^*| \leq nC$, porque não excede a capacidade do corte $(\{s\}, V \setminus \{s\})$, sendo n o número de nós, ou seja, **a complexidade do método de Ford-Fulkerson para capacidades inteiras é $O(mnC)$**.
- $O(mnC)$ não é polinomial no tamanho do *input*. $C \in O(2^{\log_2 C})$.

Método de Ford-Fulkerson para determinar fluxo máximo

Para capacidades inteiras...

- Complexidade de cada iteração é $O(m)$, sendo $|A| = m$.
- O fluxo aumenta de uma unidade pelo menos em cada iteração. Portanto, o número de iterações é menor ou igual ao fluxo ótimo $|f^*|$.
- Complexidade do método é $\mathcal{O}(m^2\, C)$, onde C é capacidade máxima dos ramos. Notar que é $O(m|f^*|)$ e $|f^*| \leq mC$. De facto, $|f^*| \leq nC$, porque não excede a capacidade do corte $(\{s\}, V \setminus \{s\})$, sendo n o número de nós, ou seja, **a complexidade do método de Ford-Fulkerson para capacidades inteiras é $O(mnC)$**.
- $O(mnC)$ não é polinomial no tamanho do *input*. $C \in O(2^{\log_2 C})$.

Algoritmo de Edmonds & Karp

Análogo ao método de Ford-Fulkerson, mas escolhe caminho γ em G_f com menor número de ramos, **aplicando BFS**. Prova-se que o número de iterações não excede mn. O algoritmo é polinomial, com complexidade $O(m^2 n)$.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$. Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$. Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.
Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.
Note-se agora que, se (u, v) estiver em G_f então $d(v) = k$ em G_f (porque seria $d(u) + 1$). Mas, como $d(v)$ decresceu e $d(v) = k$ em $G_{f'}$, então $d(v) \geq k + 1$ em G_f.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.
Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.
Note-se agora que, se (u, v) estiver em G_f então $d(v) = k$ em G_f (porque seria $d(u) + 1$). Mas, como $d(v)$ decresceu e $d(v) = k$ em $G_{f'}$, então $d(v) \geq k + 1$ em G_f.
Logo, o ramo (u, v) não estava em G_f e reapareceu em $G_{f'}$, o que implica que (v, u) pertencia ao caminho γ usado para aumentar o fluxo.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.
Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.
Note-se agora que, se (u, v) estiver em G_f então $d(v) = k$ em G_f (porque seria $d(u) + 1$). Mas, como $d(v)$ decresceu e $d(v) = k$ em $G_{f'}$, então $d(v) \geq k + 1$ em G_f.
Logo, o ramo (u, v) não estava em G_f e reapareceu em $G_{f'}$, o que implica que (v, u) pertencia ao caminho γ usado para aumentar o fluxo.
Como γ tem comprimento $\geq k + 1$ até v, teria comprimento $\geq k + 2$ até u.

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.
Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.
Note-se agora que, se (u, v) estiver em G_f então $d(v) = k$ em G_f (porque seria $d(u) + 1$). Mas, como $d(v)$ decresceu e $d(v) = k$ em $G_{f'}$, então $d(v) \geq k + 1$ em G_f.
Logo, o ramo (u, v) não estava em G_f e reapareceu em $G_{f'}$, o que implica que (v, u) pertencia ao caminho γ usado para aumentar o fluxo.
Como γ tem comprimento $\geq k + 1$ até v, teria comprimento $\geq k + 2$ até u. Então, γ não é um caminho mínimo de s para t em G_f, pois inclui um sub-caminho até u que não é mínimo (já que o caminho mínimo de s até u em G_f tem $k - 1$ ramos).

Prova da complexidade do algoritmo Edmonds-Karp

Lema 1

Seja $d(v)$ a distância mínima de s a v na rede residual correspondente ao fluxo. No algoritmo de Edmonds-Karp, $d(v)$ não decresce para nenhum v.

Prova (por redução ao absurdo): Suponhamos que $d(v)$ decresce numa iteração em que se passa de f a f', sendo v o nó nessas condições mais próximo de s em $G_{f'}$.
Seja k igual ao comprimento do caminho mais curto de s para v em $G_{f'}$ e (u, v) o seu último ramo. Então, $d(u)$ não decresceu (se não, v não seria o mais próximo!). Logo, $d(u) = k - 1$ nas redes residuais G_f e $G_{f'}$.
Note-se agora que, se (u, v) estiver em G_f então $d(v) = k$ em G_f (porque seria $d(u) + 1$). Mas, como $d(v)$ decresceu e $d(v) = k$ em $G_{f'}$, então $d(v) \geq k + 1$ em G_f.
Logo, o ramo (u, v) não estava em G_f e reapareceu em $G_{f'}$, o que implica que (v, u) pertencia ao caminho γ usado para aumentar o fluxo.
Como γ tem comprimento $\geq k + 1$ até v, teria comprimento $\geq k + 2$ até u. Então, γ não é um caminho mínimo de s para t em G_f, pois inclui um sub-caminho até u que não é mínimo (já que o caminho mínimo de s até u em G_f tem $k - 1$ ramos).
O absurdo resultou de se ter suposto que $d(v)$ decresceu. Logo, não decresceu. (cqd)

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova: Suponhamos que (u, v) é crítico em G_f e (u, v) reaparece de novo num passo em que um fluxo f' é aumentado para f''.

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova: Suponhamos que (u, v) é crítico em G_f e (u, v) reaparece de novo num passo em que um fluxo f' é aumentado para f''. Seja k o valor de $d(u)$ em G_f.

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova: Suponhamos que (u, v) é crítico em G_f e (u, v) reaparece de novo num passo em que um fluxo f' é aumentado para f''. Seja k o valor de $d(u)$ em G_f. Então, em G_f, tem-se $d(v) = k + 1$, pois o caminho γ contém (u, v) e é mínimo em G_f.

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova: Suponhamos que (u, v) é crítico em G_f e (u, v) reaparece de novo num passo em que um fluxo f' é aumentado para f''. Seja k o valor de $d(u)$ em G_f. Então, em G_f, tem-se $d(v) = k + 1$, pois o caminho γ contém (u, v) e é mínimo em G_f. O caminho para aumento γ' em $G_{f'}$ tem de incluir o ramo (v, u), para fazer reaparecer (u, v). Como é mínimo e $d(v)$ não decresce, $d(v) \geq k + 1$ em $G_{f'}$, Logo, $d(u) \geq k + 2$ em $G_{f'}$.

Prova da complexidade do algoritmo Edmonds-Karp

Um ramo (u, v) na rede residual G_f diz-se **crítico** se o caminho para aumento γ inclui (u, v) e $c_f(u, v) = c_\gamma$. Os ramos críticos (u, v) no caminho γ são removidos de G_f: na iteração seguinte, a nova rede residual tem (v, u) mas não (u, v). Contudo, (u, v) pode reaparecer numa iteração posterior. Quantas vezes (u, v) pode ser crítico?

Lema 2

Entre duas iterações em que (u, v) seja crítico, $d(u)$ aumenta de pelo menos duas unidades. Logo, (u, v) não pode ser crítico mais do que $n/2$ vezes, sendo $n = |V|$.

Prova: Suponhamos que (u, v) é crítico em G_f e (u, v) reaparece de novo num passo em que um fluxo f' é aumentado para f''. Seja k o valor de $d(u)$ em G_f. Então, em G_f, tem-se $d(v) = k + 1$, pois o caminho γ contém (u, v) e é mínimo em G_f. O caminho para aumento γ' em $G_{f'}$ tem de incluir o ramo (v, u), para fazer reaparecer (u, v). Como é mínimo e $d(v)$ não decresce, $d(v) \geq k + 1$ em $G_{f'}$, Logo, $d(u) \geq k + 2$ em $G_{f'}$.

Qualquer caminho para aumento tem comprimento $< n$ pois, por definição de caminho, não repete nós. Se $d(u)$ cresce de pelo menos 2 unidades de cada vez que (u, v) é crítico, (u, v) não pode ser crítico mais do que $n/2$ vezes. (cdq)

Prova da complexidade do algoritmo Edmonds-Karp

Complexidade do algoritmo de Edmonds-Karp:

Seja $G = (V, A, c, \{s, t\})$, a rede inicial, com $|V| = n$ e $|A| = m$. O número de iterações do algoritmo de Edmonds-Karp não excede mn. O tempo de execução do algoritmo de Edmonds-Karp é $O(m^2 n)$.

Prova da complexidade do algoritmo Edmonds-Karp

Complexidade do algoritmo de Edmonds-Karp:

Seja $G = (V, A, c, \{s, t\})$, a rede inicial, com $|V| = n$ e $|A| = m$. O número de iterações do algoritmo de Edmonds-Karp não excede mn. O tempo de execução do algoritmo de Edmonds-Karp é $O(m^2 n)$.

Prova: Os **caminhos para aumento** são percursos de s para t em G_f que não repetem nós e têm sempre algum ramo (u, v) crítico. O conjunto de ramos de qualquer rede residual G_f é um subconjunto de $\{(x, y) \mid (x, y) \in A \text{ ou } (y, x) \in A\}$. Portanto, o número de iterações não excede mn.

Cada iteração envolve a pesquisa de caminho para aumento (por BFS) e a atualização do fluxo e da rede residual, tendo complexidade temporal $O(m + n)$.

Prova da complexidade do algoritmo Edmonds-Karp

Complexidade do algoritmo de Edmonds-Karp:

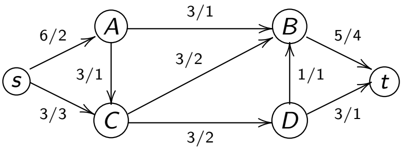
Seja $G = (V, A, c, \{s, t\})$, a rede inicial, com $|V| = n$ e $|A| = m$. O número de iterações do algoritmo de Edmonds-Karp não excede mn. O tempo de execução do algoritmo de Edmonds-Karp é $O(m^2 n)$.

Prova: Os **caminhos para aumento** são percursos de s para t em G_f que não repetem nós e têm sempre algum ramo (u, v) crítico. O conjunto de ramos de qualquer rede residual G_f é um subconjunto de $\{(x, y) \mid (x, y) \in A \text{ ou } (y, x) \in A\}$. Portanto, o número de iterações não excede mn.

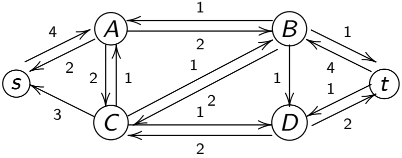
Cada iteração envolve a pesquisa de caminho para aumento (por BFS) e a atualização do fluxo e da rede residual, tendo complexidade temporal $O(m + n)$. Se assumirmos $n \in O(m)$, então a complexidade de cada iteração é $O(m)$ e, portanto, a complexidade do algoritmo de Edmonds-Karp é $O(m^2 n)$. cqd

Algoritmo de Edmonds-Karp: usa BFS para encontrar caminho γ em G_f

Fluxo f dado



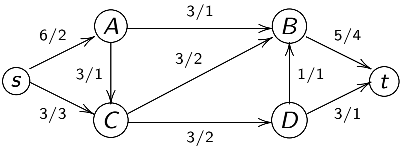
Rede residual G_f correspondente



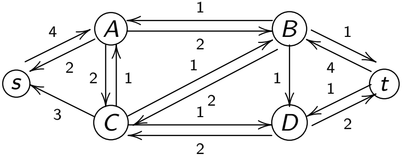
Algoritmo de Edmonds-Karp a partir de f já na rede

Algoritmo de Edmonds-Karp: usa BFS para encontrar caminho γ em G_f

Fluxo f dado

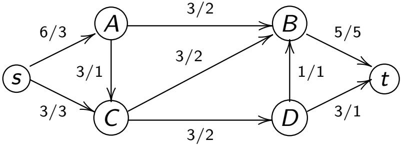


Rede residual G_f correspondente

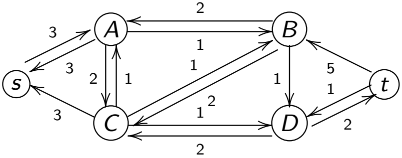


Existe caminho de s para t em G_f: (s, A, B, t), com capacidade 1, permite aumentar o fluxo de 1 unidade.

Novo fluxo f

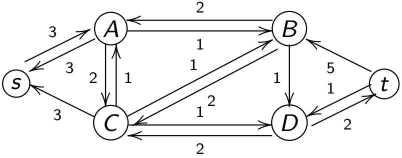


Rede residual G_f correspondente



O fluxo ainda não é máximo.

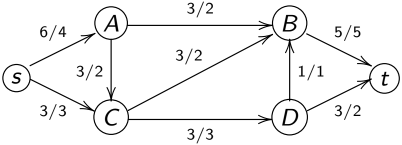
Exemplo (cont.)



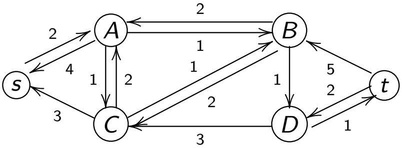
O fluxo ainda não é máximo. Existe caminho de s para t na rede residual: (s, A, C, D, t), que tem capacidade 1, permite aumentar o fluxo de 1 unidade.

Após essa iteração (como vemos abaixo à direita), o fluxo não é ainda máximo pois (s, A, B, D, t) é um caminho para aumento na nova rede residual.

Novo fluxo f



Rede residual G_f correspondente



O fluxo ainda não é máximo.

É necessário aumentar novamente o fluxo, ao longo de (s, A, B, D, t), para finalmente obter o fluxo máximo $|f^\star| = 8$.