Projeto e Análise de Algoritmos

Fluxo máximo

Diane Castonguay
diane@inf.ufg.br

Instituto de Informática
Universidade Federal de Goiás

Fluxo em rede

Um ***fluxo em rede*** $G = (V, E)$ é um grafo orientado em que cada aresta $(u,v) \in E$ tem uma ***capacidade*** não negativa $c(u,v) \geq 0$. Se $(u,v) \notin E$, supomos que $c(u,v) = 0$.

Dois vértices distinguidos: uma ***origem*** s e um ***sorvedor*** t.

Conveniência
Cada vértice reside em algum caminho de s até t

Observações: O grafo é conectado e $|E| \geq |V| - 1$

Fluxo

Um ***fluxo*** em G é uma função f: V x V → $\mathbb{R}$
que satisfaz às propriedades seguintes:

Restrição de capacidade

$f(u,v) \leq c(u,v)$

Anti-simetria oblíqua

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

Conservação de fluxo

$u \neq s, t$ temos que $\sum_{v \in V} f(u,v) = 0$

Fluxo entre vértices e valor de fluxo

A quantidade f(u,v) pode ser positiva, 0 ou negativa, é chamada ***fluxo*** do vértice u até o vértice v.

O ***valor*** de um fluxo f é definido como

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

Problema de fluxo máximo

Dados de entrada
Um fluxo em rede G

Saída
Um fluxo de valor máximo

Propriedades de Fluxo

sobre conjunto de vértices

Seja G = (V, E) um fluxo em rede, e seja f um fluxo em G. São válidas as igualdades a seguir para todo $X, Y, Z \subseteq V$.

- $f(X, X) = 0$
- $f(X, Y) = - f(Y, X)$
- Se $X \cap Y = \emptyset$, temos as somas
  - $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$
  - $f(Z, X \cup Y) = f(Z, X) + f(Z, Y)$

$|f| = f(V, t)$

$$
\begin{aligned}
|f| &= \sum_{v \in V} f(s,v) \\
&= f(s, V) \\
&= f(V, V) - f(V - s, V) && \text{(terceira propriedade)} \\
&= 0 - f(V - s, V) && \text{(primeira propriedade)} \\
&= f(V, V - s) && \text{(segunda propriedade)} \\
&= f(V, t) + f(V, V - s - t) && \text{(terceira propriedade)} \\
&= f(V, t) + 0 && \text{(conservação de fluxo)*}
\end{aligned}
$$

*Detalhado

$$
\begin{aligned}
f(V, V - s - t) &= \sum_{u \in V} \sum_{v \in V-s-t} f(u,v) = \sum_{v \in V-s-t} \left(\sum_{u \in V} f(u,v) \right) \\
&= \sum_{v \in V-s-t} (0) = 0
\end{aligned}
$$

O método de Ford-Fulkerson

FORD-FULKERSON-METHOD(G, s, t)
1. inicializar fluxo f como 0
2. enquanto existir um caminho aumentante p faça
3. ampliar fluxo f ao longo de p
4. retorna f

Redes residuais

Dados um fluxo em rede $G = (V, E)$ e um fluxo f,

a ***capacidade residual*** de (u, v) é dada por

$$c_f(u, v) = c(u, v) - f(u, v)$$

a ***rede residual*** de G induzida por f é $G_f = (V, E_f)$, onde

$$E_f = \{ (u,v) \in V \times V : c_f(u, v) > 0 \}$$

Caminhos aumentantes

Dados um fluxo em rede $G = (V, E)$ e um fluxo f,

um caminho aumentante p é um caminho simples desde s até t na rede residual G_f.

A capacidade residual de p é dada por
$c_f(p) = \min\{ c_f(u,v) : (u, v) \text{ está em } p \}$

Fluxo definido a partir de um caminho aumentante

Dados um fluxo em rede $G = (V, E)$ e um fluxo f, e seja p um caminho aumentante em G_f.

Defina uma função $f_p(u, v) = \begin{cases} c_f(p) \text{ se } (u, v) \text{ está em } p \\ -c_f(p) \text{ se } (v, u) \text{ está em } p \\ 0 \quad \text{em caso contrário} \end{cases}$

Observação: f_p é um fluxo em G_f com valor $|f_p| = c_f(p) > 0$.

Propriedade do fluxo definido a partir de um caminho aumentante

Dados um fluxo em rede $G = (V, E)$ e um fluxo f, e seja p um caminho aumentante em G_f.

Defina uma função $f' = f + f_p$.

Então, f' é um fluxo em G com valor $|f'| = |f| + |f_p| > |f|$.

Corte de fluxo em redes

Um ***corte*** (S, T) de um fluxo em rede G = (V, E) é uma partição de V em S e T = V – S tal que $s \in S$ e $t \in T$.

Se f é um fluxo em G, então
o ***fluxo líquido*** pelo corte (S, T) é definido como f(S,T).

A ***capacidade*** do corte (S, T) é c(S, T).

Um ***corte mínimo*** de uma rede é um corte
cuja capacidade é mínima dentre todos os cortes da rede.

Propriedade: corte versus fluxo

Seja f um fluxo de um fluxo em rede G e
seja (S, T) um corte de G.

$$f(S, T) = |f|$$

$$|f| \leq c(S, T)$$

Teorema de fluxo máximo e corte mínimo

Seja f um fluxo de um fluxo em rede G = (V, E)
com origem s e sorvedor t
então as condições a seguir são equivalentes:

- f é um fluxo máximo em G
- A rede residual G_f não contém nenhum caminho aumentante.
- |f| = c(S, T) para algum corte (S, T) de G

(como consequência temos que este corte é mínimo)

Algoritmo básico de Ford-Fulkerson

FORD-FULKERSON(G, s, t)

1. para cada aresta (u, v) ∈ E[G] faça
2. f[u, v] ← 0
3. f[v, u] ← 0
4. fim-para
5. enquanto existir um caminho p de s até t em G_f faça
6. $c_f(p)$ ←MIN { $c_f(u, v)$: (u, v) está em p }
7. para cada aresta (u, v) em p faça
8. f[u, v] ← f[u, v] + $c_f(p)$
9. f[v, u] ← - f[u, v]
10. fim-para
11. fim-enquanto

Análise de Ford-Fulkerson

FORD-FULKERSON(G, s, t)

1. para cada aresta (u, v) ∈ E[G] faça — tempo (E)
2. f[u, v] ← 0 — tempo (E)
3. f[v, u] ← 0 — tempo (E)
4. fim-para

Análise de Ford-Fulkerson
O loop enquanto

Implementação de G_f

Ao lugar de G e G_f, implementamos um grafo G' = (V, E')

onde E' ={(u, v): (u, v) ∈ E ou (v, u) ∈ E}

Tempo de calcular $c_f(u, v) = c(u, v) – f(u, v)$ i.e (E)

Quantas vezes???

Dependendo como é feita a escolha do caminho p,
o algortimo pode não parar....

Se as capacidades são inteiras,
o enquanto será executado no máximo |f*|
onde f* é o fluxo máximo encontrado pelo algortimo.

Análise de Ford-Fulkerson dentro do loop

6. $c_f(p) \leftarrow \text{MIN} \{ c_f(u, v) : (u, v) \text{ está em } p \}$ tempo O(E)
7. para cada aresta (u, v) em p faça tempo O(E)
8. $f[u, v] \leftarrow f[u, v] + c_f(p)$ tempo O(E)
9. $f[v, u] \leftarrow - f[u, v]$ tempo O(E)
10. fim-para

Análise de Ford-Fulkerson capacidades inteiras

Capacidades inteiras

O algoritmo é O(E |f*|)

Mesmo assim pode ser muito ruim...

Algoritmo de Edmonds-Karp

EDMONDS-KARP(G, s, t)
1. para cada aresta (u, v) ∈ E[G] faça
2. f[u, v] ← 0
3. f[v, u] ← 0
4. fim-para
/* modificar BFS(G_f, s) no equanto se u = t por cor preto e parar*/
5. BFS*(G_f, s)
6. enquanto cor[t] = PRETO faça
6. $c_f(p)$ ←MIN { $c_f(u, v)$: (u, v) está em p }
7. para cada aresta (u, v) em p faça
8. f[u, v] ← f[u, v] + $c_f(p)$
9. f[v, u] ← - f[u, v]
10. fim-para
11. fim-enquanto

Análise de Edmonds-Karp

A distância do caminho mais curto na rede residual aumentará monotonicamente com cada ampliação de fluxo.

O número total de ampliações de fluxo executadas pelo algoritmo é no máximo O(VE)

O algortimo é $O(VE^2)$