Melhores momentos

AULA PASSADA

Algoritmos

- KLEIN: mantém um fluxo que respeita u e satisfaz b e em cada iteração procura ciclos negativos.
- JEWEEL: mantém um fluxo que respeita u e tem custo mínimo (dentre as fluxos que respeita u e satisfazem e) e em cada iteração procura um caminho de incremento de custo mínimo.
- COST-SCALING: mantém um fluxo viável e uma função potencial que têm folgas ϵ-complementares. Em cada iteração resolve um problema de fluxo usando o algoritmo PREFLOW-PUSH$_\epsilon$.

Consumos de tempo

Algoritmo	consumo de tempo
KLEIN	$\mathrm{O}(nm^2UC)$
JEWEEL	$\mathrm{O}(n^3B)$
COST-SCALING	$\mathrm{O}(n^2m\lg(nC))$

AULA 21

Ciclo de Custo Médio Mínimo

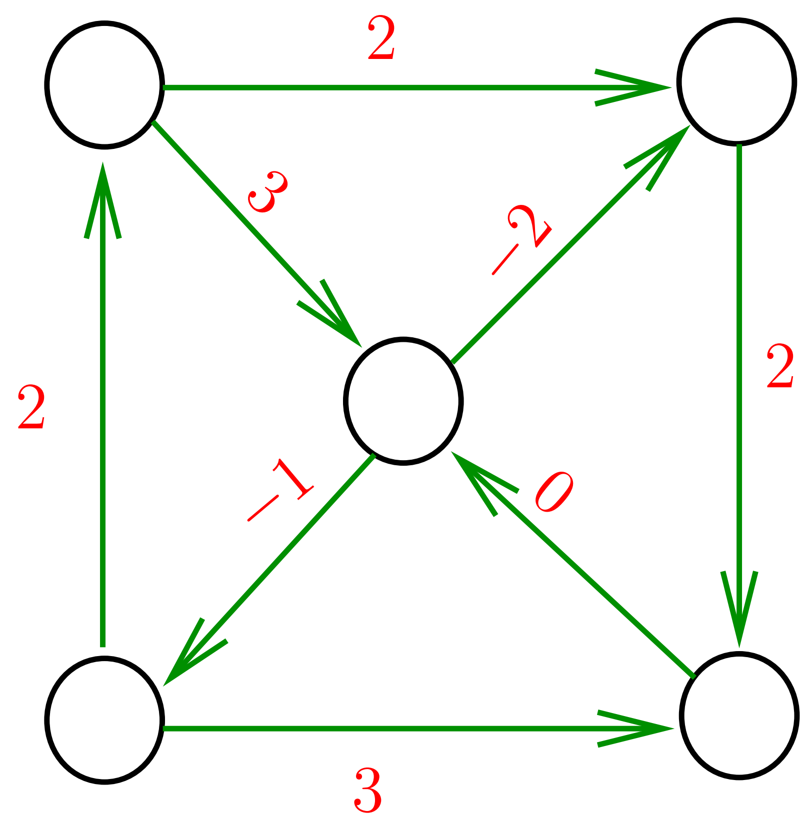
PF 25.1

Custo médio de um ciclo

Dada uma rede (N, A, c) com função-custo c, o custo médio de um ciclo O é o número

$$\mu(O) = \frac{c(O)}{|O|} \ .$$

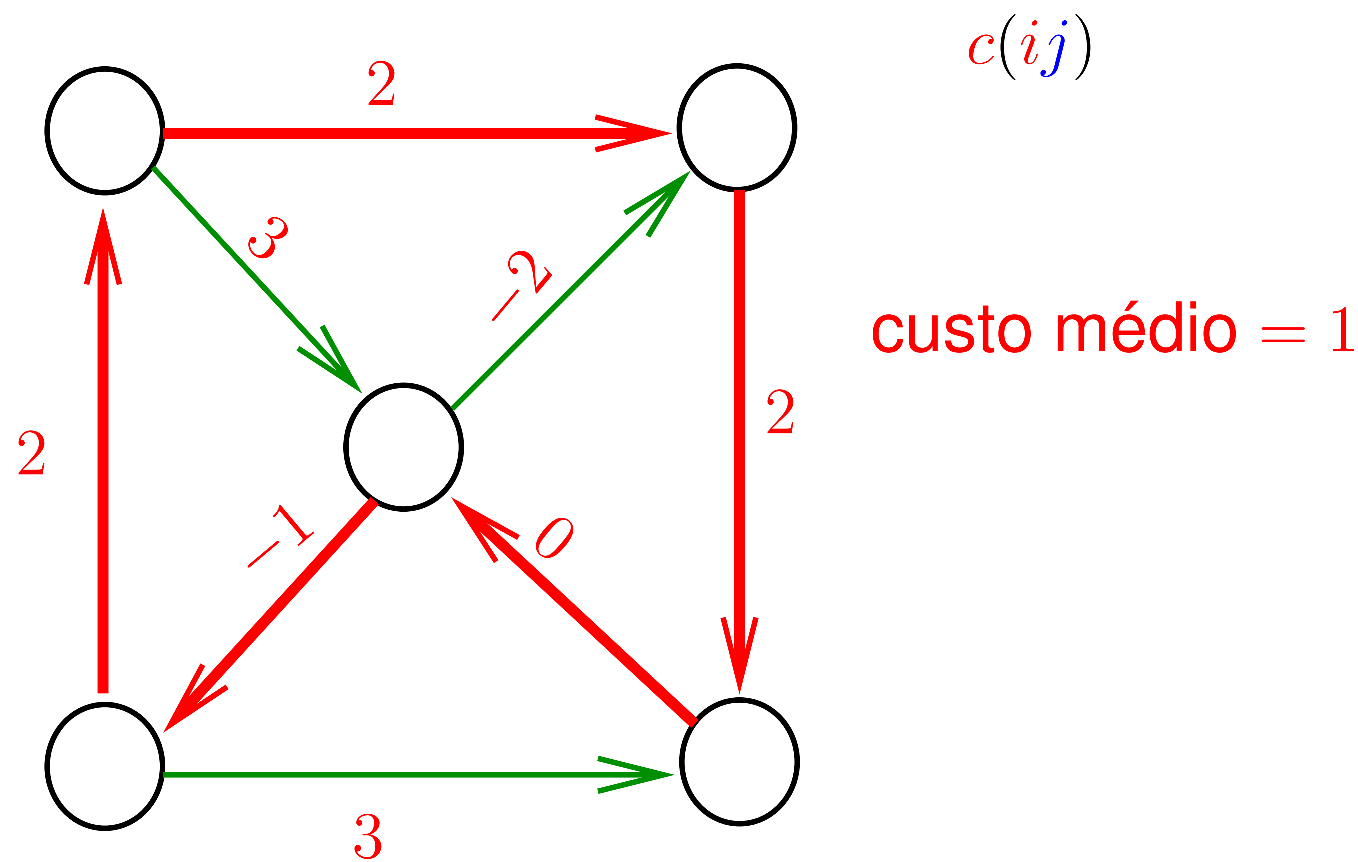
$c(ij)$



Custo médio de um ciclo

Dada uma rede (N, A, c) com função-custo c, o custo médio de um ciclo O é o número

$$\mu(O) = \frac{c(O)}{|O|}.$$

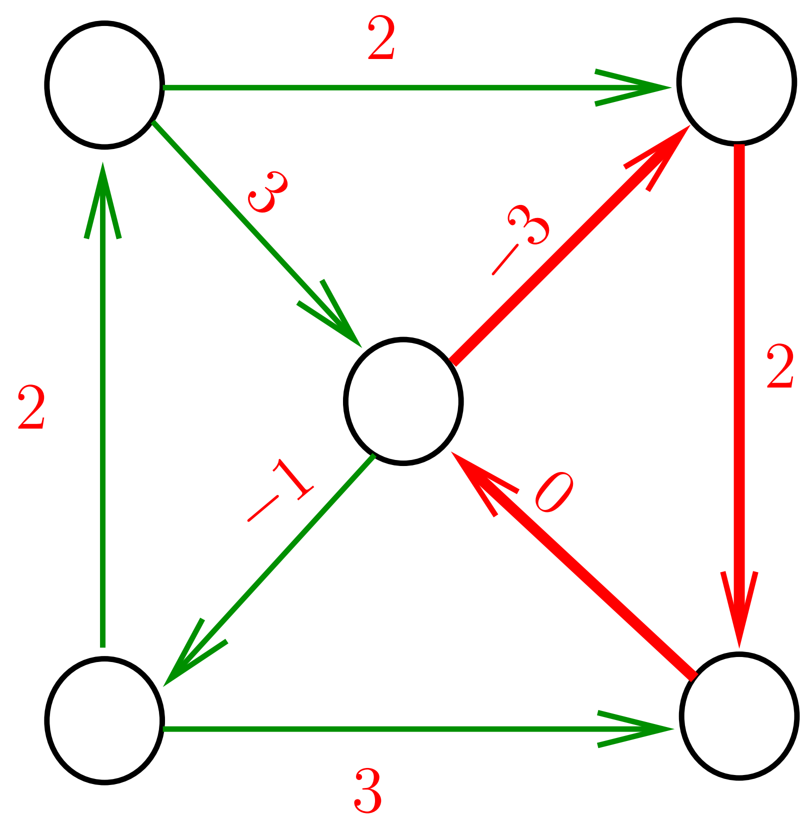


Custo médio de um ciclo

Dada uma rede (N, A, c) com função-custo c, o custo médio de um ciclo O é o número

$$\mu(O) = \frac{c(O)}{|O|} .$$

$c(ij)$

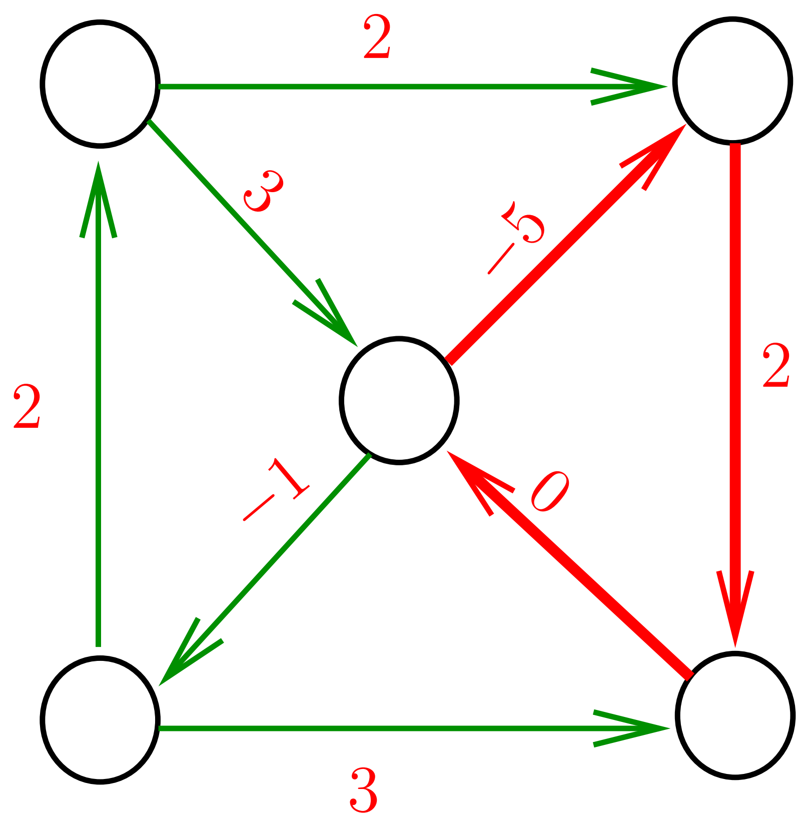


custo médio $= -1/3$

Problema do custo médio mínimo

Dada uma rede (N, A, c) encontrar um ciclo de custo médio mínimo.

$$\mu_*(c) := \min\left\{\frac{c(O)}{|O|} : O \text{ é um ciclo}\right\}$$



$c(ij)$

$\mu_*(c) = -1$

Idéia do algoritmo (1)

Seja Δ um número.

Passamos a consider 3 casos.

$c + \Delta$ representa a função definida por

$$(c+\Delta)(ij) = c(ij) + \Delta\ .$$

Caso 1. (N, A, c) tem ciclo de custo negativo

Seja O um ciclo de custo negativo

$$\begin{aligned}(c+\Delta)(O) < 0 &\Rightarrow c(O) + \Delta|O| < 0 \\ &\Rightarrow c(O) < -\Delta|O| \\ &\Rightarrow c(O)/|O| < -\Delta \\ &\Rightarrow \mu_*(c) < -\Delta\end{aligned}$$

Idéia do algoritmo (2)

Caso 2. (N, A, c) não tem ciclo de custo negativo

Caso 2A. $(c + \Delta)(O) > 0$ para todo ciclo O

Para todo ciclo O temos que

$$\begin{aligned}
(c + \Delta)(O) > 0 &\Rightarrow c(O) + \Delta|O| > 0 \\
&\Rightarrow c(O) > -\Delta|O| \\
&\Rightarrow c(O)/|O| > -\Delta \\
&\Rightarrow \mu_*(c) > -\Delta
\end{aligned}$$

Idéia do algoritmo (3)

Caso 2. (N, A, c) não tem ciclo de custo negativo

Caso 2B. $(c + \Delta)(O') = 0$ para algum ciclo O'

Se O' é um ciclo tal que $(c + \Delta)(O) = 0$, então

$$\begin{aligned}(c + \Delta)(O') = 0 &\Rightarrow c(O') + \Delta|O'| = 0\\ &\Rightarrow c(O') = -\Delta|O'|\\ &\Rightarrow c(O')/|O'| = -\Delta \qquad \Rightarrow \mu_*(c) \geq -\Delta\end{aligned}$$

Por outro lado, para todo ciclo O vale que

$$\begin{aligned}(c + \Delta)(O) \geq 0 &\Rightarrow c(O) + \Delta|O| \geq 0\\ &\Rightarrow \mu_*(c) \geq -\Delta\end{aligned}$$

Logo, $\mu_*(c) = -\Delta$.

Conseqüência

Uma simples conseqüência da análise anterior é a seguinte.

Seja O um ciclo de custo médio α. As seguintes afirmações são equivalentes:

- O é um ciclo de custo médio mínimo;
- para todo ciclo O' temos que

$$(c-\alpha)(O') \geq 0$$

e $(c-\alpha)(O) = 0$.

Algoritmo Min-Mean-Cycle

O algoritmo abaixo supõe que o (N, A, c) possui um ciclo negativo e que $c(ij)$ é multiplo de n^2 para todo ij.

MIN-MEAN-CYCLE (N, A, c)

$\Delta_1 \leftarrow 0$
$\Delta_2 \leftarrow C$
enquanto $\Delta_2 - \Delta_1 > 1$ **faça**
    $\Delta \leftarrow \lfloor(\Delta_1 + \Delta_2)/2\rfloor$
    $\langle O, y\rangle \leftarrow$ CICLO-NEGATIVO $(N, A, c + \Delta)$
    se O está definido
        então $\Delta_1 \leftarrow \Delta$
        senão $\Delta_2 \leftarrow \Delta$
$\langle O, y\rangle \leftarrow$ CICLO-NEGATIVO$(N, A, c + \Delta_1)$
devolva O

Invariantes

Na linha 3, antes do "**enquanto** $\Delta_1 \ldots$", vale que:

(i1) Δ_1 e Δ_2 são números inteiros e $0 \leq \Delta_1 < \Delta_2$;

(i2) $c(O)/|O| \geq -\Delta_2$ para todo ciclo O;

(i2)' $(c + \Delta_2)(O) \geq 0$ para todo ciclo O;

(i3) $c(O)/|O| < -\Delta_1$ para algum ciclo O;

(i3)' $(c + \Delta_1)(O) < 0$ para algum ciclo O.

Correção (1)

Quando o algoritmo pára temos que $\Delta_2 - \Delta_1 \leq 1$.
Sejam O_1 e O_2 ciclos negativos em $(N, c + \Delta_1)$.
Seja $-\alpha_1 := c(O_1)/|O_1|$ e $-\alpha_2 := c(O_2)/|O_2|$.

$$\begin{aligned}(c+\alpha_1)(O_1) &= c(O_1) + |O_1|\alpha_1 = 0 \\ &> (c+\Delta_1)(O_1) \\ &= c(O_1) + |O_1|\Delta_1\end{aligned}$$

$$\begin{aligned}(c+\alpha_1)(O_1) &= c(O_1) + |O_1|\alpha_1 = 0 \\ &\leq (c+\Delta_2)(O_1) \\ &= c(O_1) + |O_1|\Delta_2\end{aligned}$$

Assim, $\Delta_1 < \alpha_1 \leq \Delta_2$.

Correção (2)

Analogamente, $\Delta_1 < \alpha_2 \leq \Delta_2$.

Se $\alpha_1 \neq \alpha_2$, então

$$\begin{aligned}
1 &\geq \Delta_2 - \Delta_1 \\
&> |\alpha_2 - \alpha_1| \\
&= \left| \frac{c(O_1)}{|O_1|} - \frac{c(O_2)}{|O_2|} \right| \\
&= \left| \frac{|O_2|c(O_1) - |O|c(O_2)}{|O_1||O_2|} \right| \\
&\geq \frac{n^2}{n^2} = 1
\end{aligned}$$

Logo, todo ciclo negativo em $(N, A, c + \Delta_1)$ tem o mesmo custo médio $-\alpha = -\alpha_1 = -\alpha_2$ em (N, A, c).

Correção (3)

- para todo ciclo negativo em (N, A, c) tem-se que

$$(c+\alpha)(O) = 0.$$

- para todo ciclo não-negativo O em (N, A, c) tem-se que

$$\begin{aligned}(c+\alpha)(O) &= c(O) + |O|\alpha \\ &> c(O) + |O|\Delta_1 \\ &= (c+\Delta_1)(O) \\ &\geq 0,\end{aligned}$$

onde a desigualdade estrita se verifica pois $\alpha > \Delta_1$.

Portanto, $\mu_*(c) = \alpha$.

Consumo de tempo

O consumo de tempo do algoritmo
MIN-MEAN-CYCLE é $O(nm \lg C)$.

Condição de parada

Se $\mu(O) \neq \mu(O')$, então

$$\left|\frac{c(O)}{|O|} - \frac{c(O')}{|O'|}\right| = \left|\frac{|O'|c(O) - |O|c(O')}{|O||O'|}\right|$$
$$\geq \frac{1}{n^2}$$

Sem a hipótese de que c é multiplo de n^2, o consumo de tempo do algoritmo é $\mathrm{O}(nm\lg(nC))$.

Algoritmo Min-Mean-Cycle-Dyn-Prog

O algoritmo a seguir recebe um rede (N, A, c) e devolve um ciclo O de custo médio mínimo.

MIN-MEAN-CYCLE-DYN-PROG (N, A, c)

1 $\quad \mu \leftarrow$ MIN-MEAN-CYCLE-VALUE(N, A, c)

2 $\quad \langle O, y\rangle \leftarrow$ CICLO-NEGATIVO$(N, A, c - \mu)$

3 $\quad A' \leftarrow \{ij : y(j) - y(i) = c(ij) - \mu\}$

4 $\quad \langle O', y\rangle \leftarrow$ DAG(N, A')

5 $\quad$ **devolva** O'