Redução AP (a principal redução para aproximabilidade)

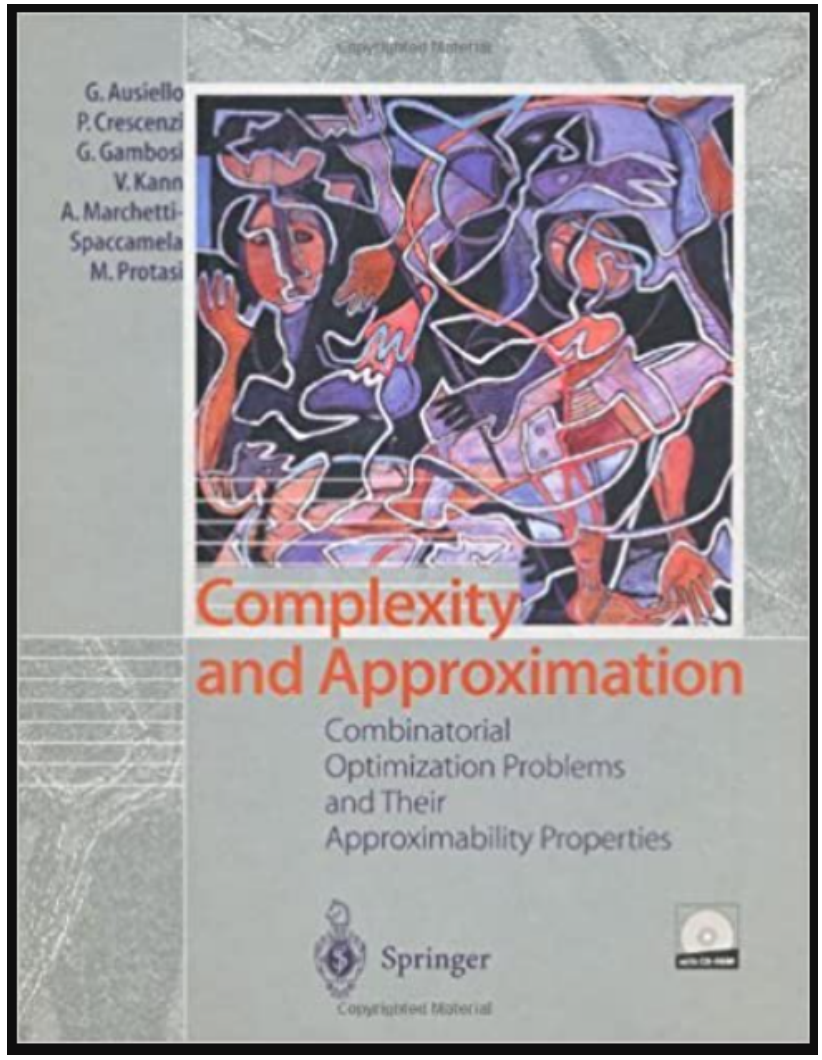


Figure: Livro "Complexity and Approximation" de Ausiello et al.

Algoritmo Aproximativo (Def. padrão versus alternativa)

DEFINIÇÃO: Dada uma instancia I e uma solução $S \in Sol(\mathrm{I})$, seja

$$\mathcal{R}(\mathrm{I}, S) = \max\left\{\frac{opt(\mathrm{I})}{val(S)}, \frac{val(S)}{opt(\mathrm{I})}\right\} \geq 1 \quad (\textit{performance ratio})$$

(*razão de aproximação*)

Um algoritmo $\mathcal{A}$ para um problema de otimização é **$\alpha_{\geq 1}$-aproximativo** se produz uma solução $\mathcal{A}(\mathrm{I})$ tal que $\mathcal{R}(\mathrm{I}, \mathcal{A}(\mathrm{I})) \leq \alpha$ para toda instância I.

Ou seja, um algoritmo tem fator de aproximação α se sempre produz uma solução com razão de aproximação $\leq \alpha$ (nessa definição alternativa).

- Problemas de minimização $\Rightarrow$ Definições coincidem (mesmo fator de aproximação)
- Problemas de maximização $\Rightarrow$ Definições divergem (fatores de aproximação inversos)
- Alg. 0.5-aprox MaxSAT-Johnson $\Rightarrow$ 2 - aprox (definição alternativa)
- Alg. 0.75-aprox MaxSAT-GW $\Rightarrow$ (1.333)-aprox (definição alternativa)
- Alg. 0.878-aprox MaxCut-GW $\Rightarrow$ (1.138)-aprox (definição alternativa)
- Alg. 0.931-aprox Max2SAT-FG $\Rightarrow$ (1.074)-aprox (definição alternativa)

Classes de Aproximabilidade

NPO: Problemas de otimização tais que toda solução tem um tamanho limitado por um polinômio no tamanho da instância e nos quais podemos, em tempo polinomial, reconhecer instâncias e soluções de instâncias, bem como calcular valores de soluções.

PO: Problemas NPO com algoritmo exato polinomial.

APX: Problemas NPO com algoritmo poli α-aprox. para α constante.

poly-APX: Problemas NPO com algoritmo de tempo poli $\alpha(n)$-aprox. para função **polinomial** $\alpha(n) = O(n^k)$, onde $k = const$ e $n = \langle \mathrm{I} \rangle$.

log-APX: idem, mas função **logarítmica** $\alpha(n) = O(\log n)$.

PTAS: Problemas NPO que tem PTAS (esquema de aproximação em tempo polinomial): algoritmo $\mathcal{A}_\varepsilon$ polinomial em $\langle \mathrm{I} \rangle$ que retorna solução satisf. $\mathcal{R}(\mathrm{I}, \mathcal{A}_\varepsilon(\mathrm{I})) \leq 1 + \varepsilon$, p/ cada racional $\varepsilon > 0$ e instância I.

FPTAS: Problemas NPO que tem FPTAS (esquema de aproximação em tempo completamente polinomial): PTAS polinomial também em $1/\varepsilon$.

PO $\subseteq$ **FPTAS** $\subseteq$ **PTAS** $\subseteq$ **APX** $\subseteq$ **log-APX** $\subseteq$ **poly-APX** $\subseteq$ **NPO**

Classes de Aproximabilidade - Exemplos

NPO: Caixeiro Viajante (TSP)

PO: Menor Caminho, Árvore Geradora Mínima (MST).

APX: Bin Packing, Vertex Cover, Caixeiro Viajante Métrico (TSPM).

poly-APX: MaxClique, MinColor, pois têm alg. n-aprox.

log-APX: Set Cover, pois tem alg. $(\ln n + 1)$-aprox.

PTAS: Escalonamento Mínimo, Caixeiro Viajante Euclidiano (TSPE).

FPTAS: Problema da Mochila.

$$\textbf{PO} \subseteq \textbf{FPTAS} \subseteq \textbf{PTAS} \subseteq \textbf{APX} \subseteq \textbf{log-APX} \subseteq \textbf{poly-APX} \subseteq \textbf{NPO}$$

Classes de Aproximabilidade - Relações, se P $\neq$ NP

NPO: TSP $\notin$ poly-APX (já visto)

PO: Menor Caminho, Árvore Geradora Mínima (MST).

APX: Bin Packing $\notin$ PTAS, pois é $(1.5 - \varepsilon)$-inaproximável.

poly-APX: MaxClique, MinColor $\notin$ log-APX, pois são $n^{1-\varepsilon}$-inaprox. (Zuckerman'06)

log-APX: Set Cover $\notin$ APX, pois é (1-ε) ln n-inaprox. (Moshkovitz'15)

PTAS: Escalonamento Mínimo $\notin$ FPTAS.

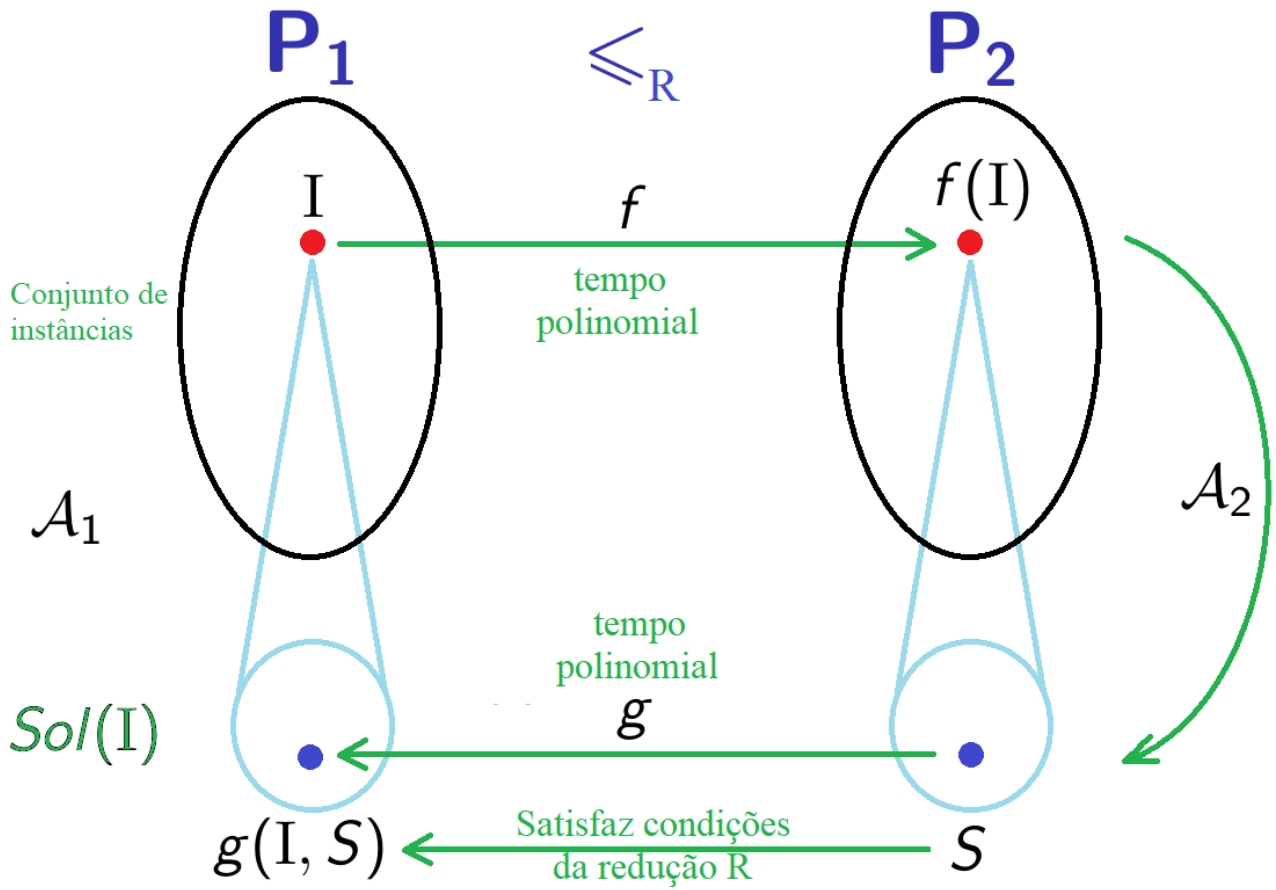
FPTAS: Problema da Mochila $\notin$ PO, pois é NP-Difícil.

PO $\subsetneq$ **FPTAS** $\subsetneq$ **PTAS** $\subsetneq$ **APX** $\subsetneq$ **log-APX** $\subsetneq$ **poly-APX** $\subsetneq$ **NPO**

Redução entre Problemas NPO P_1 e P_2

Redução $P_1 \leq_R P_2$: par (f, g)

- f e g: funções computáveis tempo polin, no tam de suas instâncias
- Instância I de P_1 $\Rightarrow$ $f(\mathrm{I})$ é uma instância de P_2
- $g(\mathrm{I}, S)$ é solução de I em P_1 $\Leftarrow$ Solução S de $f(\mathrm{I})$ em P_2



Redução entre Problemas NPO P_1 e P_2

- f e g: funções computáveis tempo polin. no tam de suas instâncias
- Instância I de P_1 $\Rightarrow$ $f(\mathrm{I})$ é uma instância de P_2
- $g(\mathrm{I}, S)$ é solução de I em P_1 $\Leftarrow$ Solução S de $f(\mathrm{I})$ em P_2
- tal que, para toda instância I de P_1 e toda solução S de $f(\mathrm{I})$ em P_2

$P_1 \leq_{strict} P_2$: Redução strict (f, g)

- $\boxed{\mathcal{R}_{P_2}(f(\mathrm{I}), S) \leq r \quad \Rightarrow \quad \mathcal{R}_{P_1}(\mathrm{I}, g(\mathrm{I}, S)) \leq r}$

$P_1 \leq_C P_2$: Redução contínua $(f, g, \alpha_{\geq 1})$

- $\boxed{\mathcal{R}_{P_2}(f(\mathrm{I}), S) \leq r \quad \Rightarrow \quad \mathcal{R}_{P_1}(\mathrm{I}, g(\mathrm{I}, S)) \leq \alpha \cdot r}$

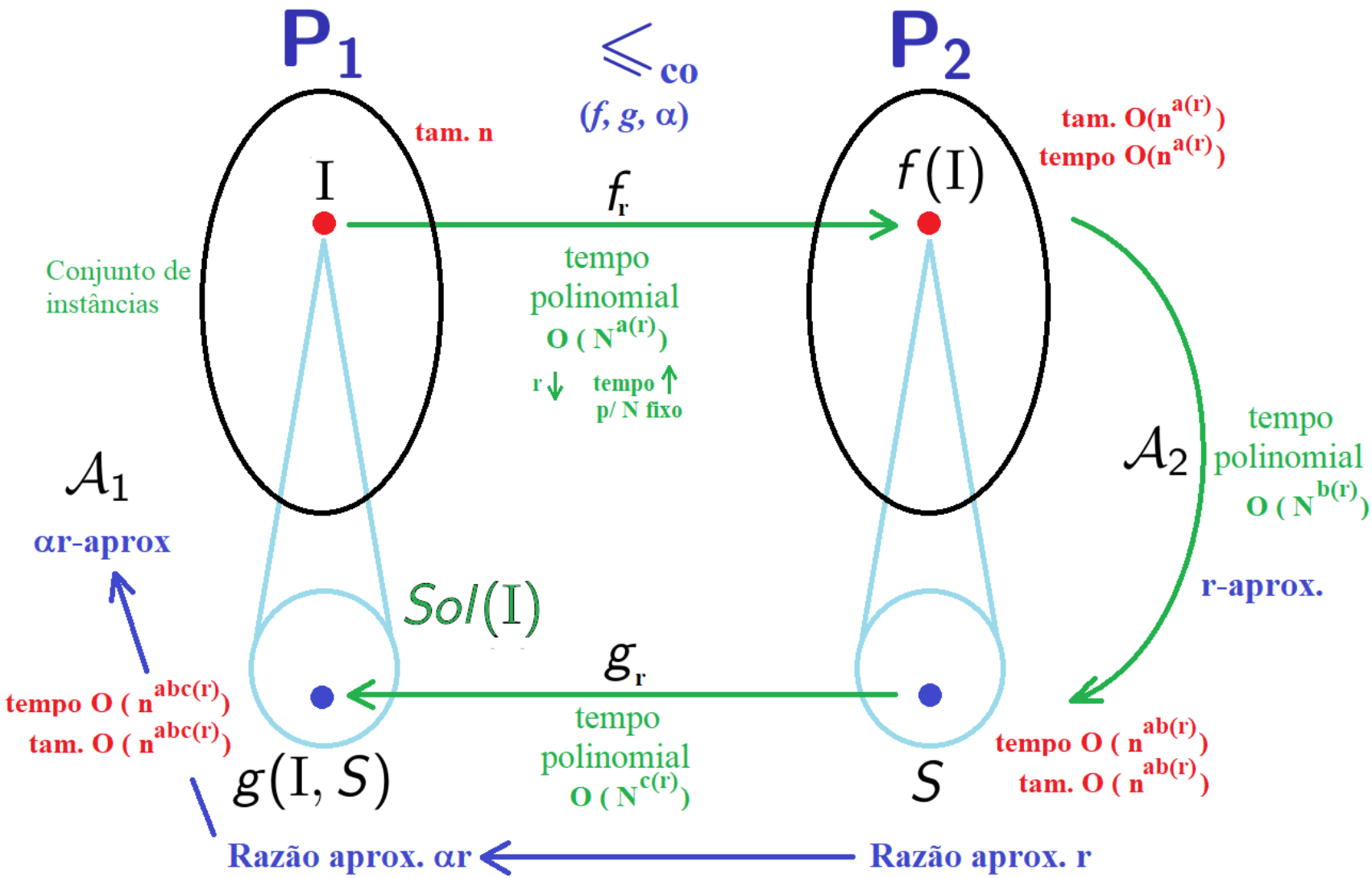
$P_1 \leq_E P_2$: Redução E (relativa ao "*erro*") $(f, g, \alpha_{\geq 1})$

- $\boxed{\mathcal{R}_{P_2}(f(\mathrm{I}), S) \leq r \quad \Rightarrow \quad \mathcal{R}_{P_1}(\mathrm{I}, g(\mathrm{I}, S)) - 1 \leq \alpha \cdot (r-1)}$
- $\mathbf{P_1 \leq_{strict} P_2} \quad \Rightarrow \quad \mathbf{P_1 \leq_E P_2} \quad \Rightarrow \quad \mathbf{P_1 \leq_C P_2}$

Redução entre Problemas NPO P_1 e P_2

- $\mathbf{P_1 \leq_C P_2}$ e $P_2 \in APX \Rightarrow P_1 \in APX$
  - $\mathbf{P_1 \leq_C P_2}$ e $P_1 \notin APX \Rightarrow P_2 \notin APX$
- $\mathbf{P_1 \leq_E P_2}$ e $P_2 \in PTAS \Rightarrow P_1 \in PTAS$
  - $\mathbf{P_1 \leq_E P_2}$ e $P_1 \notin PTAS \Rightarrow P_2 \notin PTAS$
- $\mathbf{P_1 \leq_E P_2}$ e $P_2 \in FPTAS \Rightarrow P_1 \in FPTAS$
  - $\mathbf{P_1 \leq_E P_2}$ e $P_1 \notin FPTAS \Rightarrow P_2 \notin FPTAS$
- $\mathbf{P_1 \leq_C P_2}$ e $\mathbf{P_2 \leq_C P_3} \Rightarrow \mathbf{P_1 \leq_C P_3}$
- $\mathbf{P_1 \leq_E P_2}$ e $\mathbf{P_2 \leq_E P_3} \Rightarrow \mathbf{P_1 \leq_E P_3}$

PROVA: $\mathbf{P_1} \leq_{\mathbf{co}} \mathbf{P_2}$ e $\mathbf{P_2} \in$ APX $\Rightarrow$ $\mathbf{P_1} \in$ APX



Redução entre Problemas NPO P_1 e P_2

Redução (f, g): para todo $r \geq 1$ racional

- f_r e g_r: funções computáveis tempo polin. no tam de suas instâncias
- Instância I de $P_1 \Rightarrow f_r(\mathrm{I})$ é uma instância de P_2
- $g_r(\mathrm{I}, S)$ é solução de I em $P_1 \Leftarrow$ Solução S de $f_r(\mathrm{I})$ em P_2
- tal que, para toda instância I de P_1 e toda solução S de $f_r(\mathrm{I})$ em P_2

$P_1 \leq_{co} P_2$: Redução Co (contínua forte) $(f, g, \alpha_{\geq 1})$

- $\boxed{\mathcal{R}_{P_2}(f_r(\mathrm{I}), S) \leq r \Rightarrow \mathcal{R}_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq \alpha \cdot r}$

$P_1 \leq_{ap} P_2$: Redução AP $(f, g, \alpha_{\geq 1})$

- $\boxed{\mathcal{R}_{P_2}(f_r(\mathrm{I}), S) \leq r \Rightarrow \mathcal{R}_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) - 1 \leq \alpha \cdot (r-1)}$

$$\mathbf{P_1 \leq_{strict} P_2} \Rightarrow \begin{array}{ccc} \mathbf{P_1 \leq_E P_2} & \Rightarrow & \mathbf{P_1 \leq_C P_2} \\ \Downarrow & & \Downarrow \\ \mathbf{P_1 \leq_{ap} P_2} & \Rightarrow & \mathbf{P_1 \leq_{co} P_2} \end{array}$$

Redução entre Problemas NPO P_1 e P_2

- $\mathbf{P_1 \leq_{co} P_2}$ e $P_2 \in$ APX $\Rightarrow$ $P_1 \in$ APX
- $\mathbf{P_1 \leq_{co} P_2}$ e $P_2 \in$ poly-APX $\Rightarrow$ $P_1 \in$ poly-APX
- $\mathbf{P_1 \leq_{co} P_2}$ e $P_2 \in$ log-APX $\Rightarrow$ $P_1 \in$ log-APX

- $\mathbf{P_1 \leq_{ap} P_2}$ e $P_2 \in$ PTAS $\Rightarrow$ $P_1 \in$ PTAS

- $\mathbf{P_1 \leq_{co} P_2}$ e $\mathbf{P_2 \leq_{co} P_3} \Rightarrow \mathbf{P_1 \leq_{co} P_3}$
- $\mathbf{P_1 \leq_{ap} P_2}$ e $\mathbf{P_2 \leq_{ap} P_3} \Rightarrow \mathbf{P_1 \leq_{ap} P_3}$

- $\mathbf{P_1 \leq_{strict} P_2} \Rightarrow \mathbf{P_1 \leq_{E} P_2} \Rightarrow \mathbf{P_1 \leq_{C} P_2}$

$$\begin{array}{ccc} \mathbf{P_1 \leq_{E} P_2} & \Rightarrow & \mathbf{P_1 \leq_{C} P_2} \\ \Downarrow & & \Downarrow \\ \mathbf{P_1 \leq_{ap} P_2} & \Rightarrow & \mathbf{P_1 \leq_{co} P_2} \\ \Downarrow & & \\ \mathbf{P_1 \leq_{ptas} P_2} & & \end{array}$$

NPO-completude e APX-completude

NPO-difícil e APX-difícil

- Problema $\mathbf{P_2}$ é **NPO-difícil** se $P_1 \leq_{co} \mathbf{P_2}$, $\forall P_1 \in$ NPO
- Problema $\mathbf{P_2}$ é **poly-APX-difícil** se $P_1 \leq_{co} \mathbf{P_2}$, $\forall P_1 \in$ poly-APX
- Problema $\mathbf{P_2}$ é **log-APX-difícil** se $P_1 \leq_{co} \mathbf{P_2}$, $\forall P_1 \in$ log-APX
- Problema $\mathbf{P_2}$ é **APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ APX
- Problema $\mathbf{P_2}$ é **APX-completo** se é APX e APX-difícil
- Problema $\mathbf{P_2}$ é **NPO-completo** se é NPO e NPO-difícil

Conclusões (se P $\neq$ NP)

- $\mathbf{P_2}$ é **NPO-difícil** $\Rightarrow$ $\mathbf{P_2} \notin$ poly-APX
- $\mathbf{P_2}$ é **poly-APX-difícil** $\Rightarrow$ $\mathbf{P_2} \notin$ log-APX
- $\mathbf{P_2}$ é **log-APX-difícil** $\Rightarrow$ $\mathbf{P_2} \notin$ APX
- $\mathbf{P_2}$ é **APX-difícil** $\Rightarrow$ $\mathbf{P_2} \notin$ PTAS

NPO-completude e APX-completude

NPO-difícil e APX-difícil

- Problema $\mathbf{P_2}$ é **NPO-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ NPO
- Problema $\mathbf{P_2}$ é **poly-APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ poly-APX
- Problema $\mathbf{P_2}$ é **log-APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ log-APX
- Problema $\mathbf{P_2}$ é **APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ APX
- Problema $\mathbf{P_2}$ é **APX-completo** se é APX e APX-difícil
- Problema $\mathbf{P_2}$ é **NPO-completo** se é NPO e NPO-difícil

Se considerarmos só r grande: $\leq_{co} \Rightarrow \leq_{ap}$

- $\mathcal{R}_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq \alpha \cdot r \Rightarrow \mathcal{R}_{P_1} - 1 \leq \alpha r - 1 \leq \alpha r$
- $\mathcal{R}_{P_1} - 1 \leq \alpha r \leq k\alpha(r-1)$ **para** $k \geq \frac{r}{r-1} = 1 + \frac{1}{r-1}$
- $\mathcal{R}_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq 1 + (\quad 2 \quad) \cdot \alpha \cdot (r-1)$, se $r \geq 2$
- $\mathcal{R}_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq 1 + \left(1 + \frac{1}{\varepsilon}\right) \cdot \alpha \cdot (r-1)$, se $r \geq 1 + \varepsilon$
- **Conclusão:** apesar da redução contínua ser mais "relaxada" (ótimos em P_2 não precisam levar a ótimos em P_1), em geral podem ser transformadas em Redução AP com ajuda das funções g e g^{-1} que levam soluções de um problema para outro, relacionando seus valores.

NPO-completude e APX-completude - Redução AP

- Problema $\mathbf{P_2}$ é **NPO-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ NPO
- Problema $\mathbf{P_2}$ é **poly-APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ poly-APX
- Problema $\mathbf{P_2}$ é **log-APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ log-APX
- Problema $\mathbf{P_2}$ é **APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ APX

Conclusões

- $\mathbf{P_1} \leq_{\mathbf{ap}} \mathbf{P_2}$ e $\mathbf{P_1} \in$ NPO-difícil $\Rightarrow$ $\mathbf{P_2} \in$ NPO-difícil
- $\mathbf{P_1} \leq_{\mathbf{ap}} \mathbf{P_2}$ e $\mathbf{P_1} \in$ poly-APX-difícil $\Rightarrow$ $\mathbf{P_2} \in$ poly-APX-difícil
- $\mathbf{P_1} \leq_{\mathbf{ap}} \mathbf{P_2}$ e $\mathbf{P_1} \in$ log-APX-difícil $\Rightarrow$ $\mathbf{P_2} \in$ log-APX-difícil
- $\mathbf{P_1} \leq_{\mathbf{ap}} \mathbf{P_2}$ e $\mathbf{P_1} \in$ APX-difícil $\Rightarrow$ $\mathbf{P_2} \in$ APX-difícil
- **MaxSAT**, BinPacking, MaxCut, MinCV, TSPM são APX-completos [Papadimitriou, Yannakakis'91]
- MaxWeighted-SAT, **TSP** são NPO-completos [Orponen, Mannila'87]
- **Clique** é poly-APX-completo [Bazgan et al.'05]
- **Set Cover** é log-APX-completo [Escoffier, Paschos'06]

NPO-completude versus exp-APX-completude

- Classe **exp-APX**: problemas NPO que tem algoritmo de tempo polinomial com fator de aproximação exponencial $O(2^{n^k})$ para alguma constante k, onde n é o tamanho da instância do problema.
- É possível provar que todo problema **NPO** tal que toda instância é viável (ou seja, tem alguma solução) pertence a **exp-APX**.
- Por outro lado, problemas **NPO** nos quais decidir se uma instância é viável é NP-difícil não estão em **exp-APX**, se P$\neq$ NP.
- **PO** $\subsetneq$ **FPTAS** $\subsetneq$ **PTAS** $\subsetneq$ **APX** $\subsetneq$ **log-APX** $\subsetneq$ **poly-APX** $\subsetneq$ **exp-APX** $\subsetneq$ **NPO**
- Problema $\mathbf{P_2}$ é **exp-APX-difícil** se $P_1 \leq_{ap} \mathbf{P_2}$, $\forall P_1 \in$ exp-APX
- Problema $\mathbf{P_2}$ é **exp-APX-completo** se é exp-APX e exp-APX-difícil
- **TSP*** é exp-APX-completo: versão com grafo completo, pois sempre tem uma solução.
- **TSP** é NPO-completo [Orponen, Mannila'87]: versão com grafo não necessariamente completo, pois decidir se possui um ciclo hamiltoniano é NP-difícil.

Redução AP: MinWeighted-SAT $\leq_{ap}$ MinProgLinear01

Instância de *Minimum Weighted Satisfiability*

- Satisfazer a fórmula obtendo peso mínimo nas variáveis verdadeiras.
- $(u_1 \vee \overline{u_2} \vee u_3) \quad \wedge \quad (u_4 \vee u_5) \quad \wedge \quad (u_2 \vee \overline{u_3} \vee \overline{u_4})$
- Variáveis u_1, u_2, u_3, u_4 e u_5 com pesos 6, 5, 4, 3 e 2

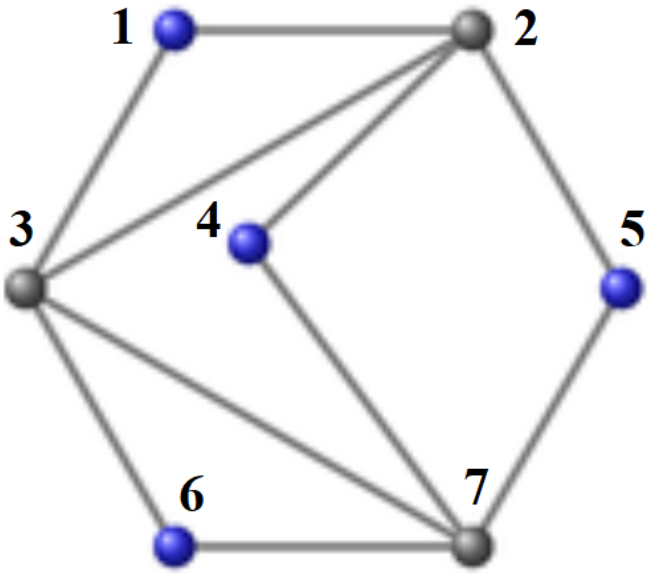
Instância de *Minimum* $\{0,1\}$*-Linear Programming*

- Minimizar $6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5$ restrito a
- $x_1 + (1 - x_2) + x_3 \geq 1$
- $x_4 + x_5 \geq 1$
- $x_2 + (1 - x_3) + (1 - x_4) \geq 1$

Melhorando:

- Minimizar $6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5$ restrito a
- $x_1 - x_2 + x_3 \geq 0$
- $x_4 + x_5 \geq 1$
- $x_2 - x_3 - x_4 \geq -1$
- **MinProgLinear01 e MaxProgLinear01 são NPO-Completos**

Redução AP: Independent Set $\leq_{ap}$ Max P_3-convex Set



Podemos assumir que o grafo G é conexo

f

strict

Conjunto Independente: 1 2 3 4 5 6 7

12 13 23 24 25 36 37 47 57 67

12 13 23 24 25 36 37 47 57 67

Clique

Objetivo: conjunto máximo de vértices que não infectam outros vértices, onde um vértice é infectado se tem dois ou mais vizinhos infectados.

Se tem mais de 1 vértice, todos são de cima, representando vértices do grafo original

I = {1, 4, 5, 6}
conj. indep.

g

g^{-1}

Convex Set = {1, 4, 5, 6}

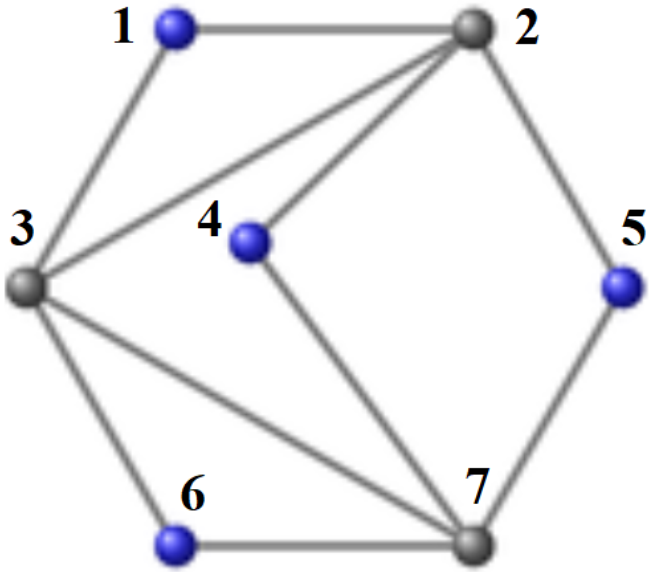
poly-APX-Completo

mesmo em grafos split:

poly-APX-Completo

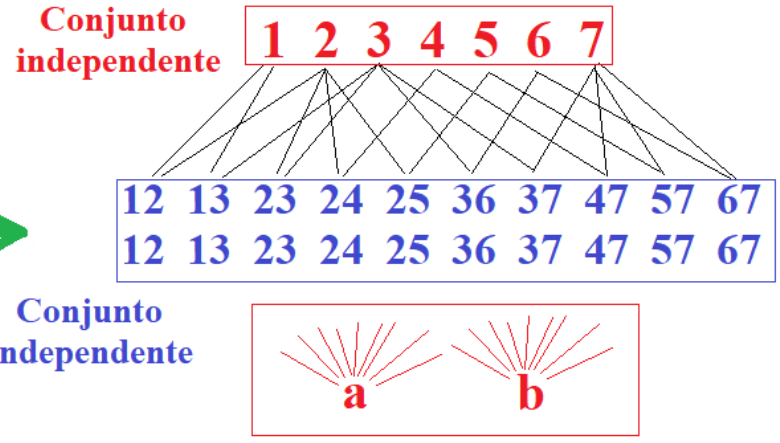
pois é poly-APX (algoritmo trivial)

Redução AP: Independent Set $\leq_{ap}$ Max P_3-convex Set



Podemos assumir que o grafo G é conexo

f



Objetivo: conjunto máximo de vértices que não infectam outros vértices, onde um vértice é infectado se tem dois ou mais vizinhos infectados. Se tem mais de 2 vértices, no máximo 1 do meio e nenhum de baixo. O resto são todos de cima (representando vértices do grafo original)

I = {1, 4, 5, 6}
conj. indep.

g

g^{-1}

Convex Set = {1, 4, 5, 6, 12}

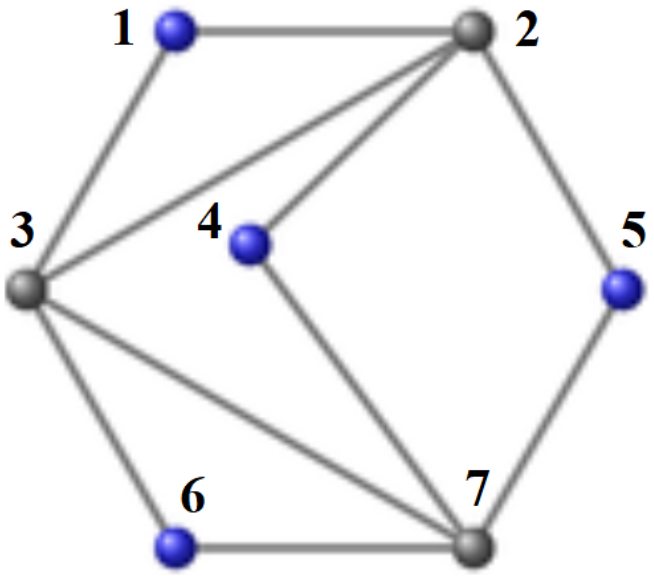
$$R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) = \frac{opt_1}{val_1} = \frac{opt_2 - 1}{val_2 - 1} = \frac{opt_2}{val_2} \cdot \left(\frac{opt_2 - 1}{opt_2} \cdot \frac{val_2}{val_2 - 1} \right)$$

$$\Rightarrow R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq 2 \cdot R_{P_2}(f_r(\mathrm{I}), S) \Rightarrow \textbf{Redução co}$$

$$R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) - 1 = \frac{opt_1 - val_1}{val_1} = \frac{opt_2 - val_2}{val_2 - 1} = \frac{opt_2 - val_2}{val_2} \cdot \left(\frac{val_2}{val_2 - 1} \right)$$

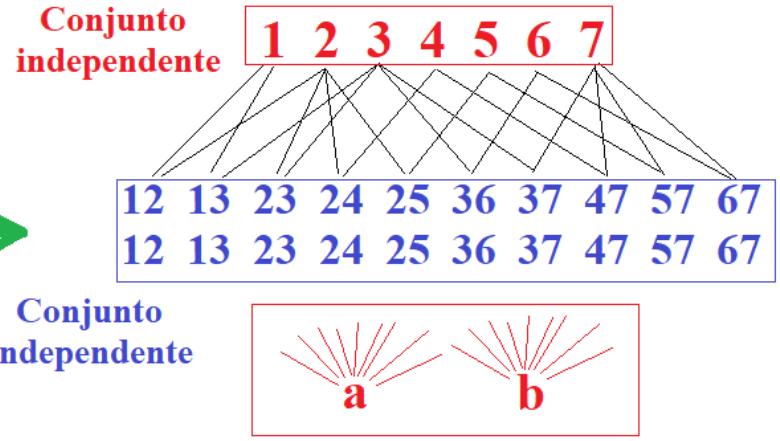
$$\Rightarrow R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq 1 + 2 \cdot \left(R_{P_2}(f_r(\mathrm{I}), S) - 1 \right) \Rightarrow \textbf{Redução AP}$$

Redução AP: Independent Set $\leq_{ap}$ Max P_3-convex Set



Podemos assumir que o grafo G é conexo

f



Objetivo: conjunto máximo de vértices que não infectam outros vértices, onde um vértice é infectado se tem dois ou mais vizinhos infectados. Se tem mais de 2 vértices, no máximo 1 do meio e nenhum de baixo. O resto são todos de cima (representando vértices do grafo original)

I = {1, 4, 5, 6}
conj. indep.

g

g^{-1}

Convex Set = {1, 4, 5, 6, 12}

poly-APX-Completo

mesmo em grafos bipartidos:

poly-APX-Completo

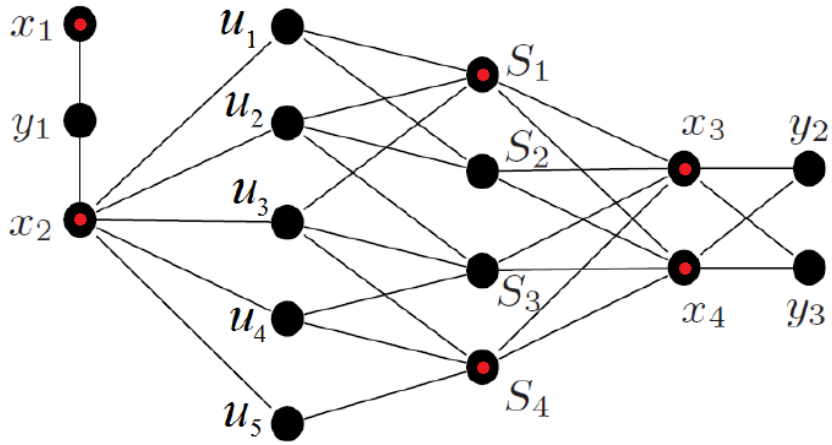
pois é poly-APX (algoritmo trivial)

Redução AP: Set Cover $\leq_{ap}$ Min P_3-interval Set

$$U = \{u_1, u_2, u_3, u_4, u_5\}$$

$$S_1 = \{u_1, u_2, u_3\}$$
$$S_2 = \{u_1, u_2\}$$
$$S_3 = \{u_2, u_3, u_4\}$$
$$S_4 = \{u_3, u_4, u_5\}$$

f



$$DD = \{S_1, S_4, x_1, x_2, x_3, x_4, y_1\}$$

$$DD = \{S_1, S_4, x_1, x_2, x_3, x_4\}$$

g

g^{-1}

$$C = \{S_1, S_4\}$$

$$R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) - 1 = \frac{val_1 - opt_1}{opt_1} = \frac{val_2 - opt_2}{opt_2 - 4} = \frac{val_2 - opt_2}{opt_2} \cdot \left(\frac{opt_2}{opt_2 - 4}\right)$$

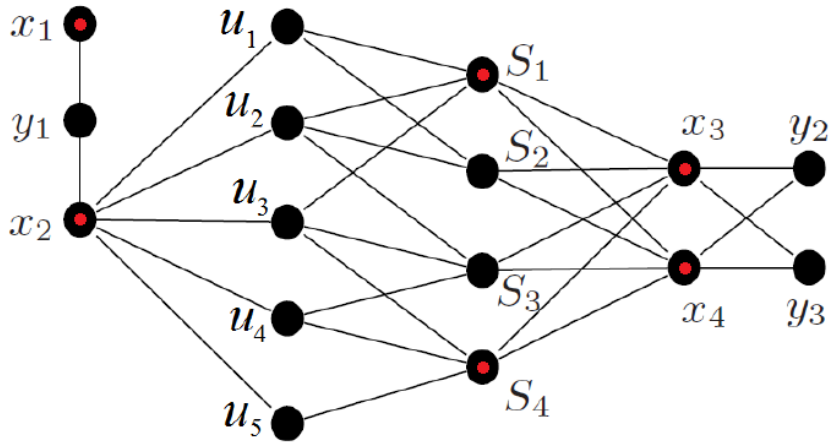
$$\Rightarrow R_{P_1}(\mathrm{I}, g_r(\mathrm{I}, S)) \leq 1 + 5 \cdot \left(R_{P_2}(f_r(\mathrm{I}), S) - 1 \right) \Rightarrow \textbf{Redução AP}, \text{ pois } opt_2 \geq 5$$

Redução AP: Set Cover $\leq_{ap}$ Min P_3-interval Set

$$U = \{u_1, u_2, u_3, u_4, u_5\}$$

$$S_1 = \{u_1, u_2, u_3\}$$
$$S_2 = \{u_1, u_2\}$$
$$S_3 = \{u_2, u_3, u_4\}$$
$$S_4 = \{u_3, u_4, u_5\}$$

f



$C = \{S_1, S_4\}$

g

$$DD = \{S_1, S_4, x_1, x_2, x_3, x_4, y_1\}$$

$$DD = \{S_1, S_4, x_1, x_2, x_3, x_4\}$$

g^{-1}

log-APX-difícil

mesmo em grafos bipartidos:

log-APX-difícil

Redução L (linear) entre Problemas NPO P_1 e P_2

- f e g: funções computáveis tempo polin, no tam de suas instâncias
- Instância I de P_1 $\Rightarrow$ $f(\mathrm{I})$ é uma instância de P_2
- $g(\mathrm{I}, S)$ é solução de I em P_1 $\Leftarrow$ Solução S de $f(\mathrm{I})$ em P_2

$P_1 \leq_L P_2$: Redução L (f, g, α, β) de P_1 para P_2

- **se** $opt_{P_2}(f(\mathrm{I})) \leq \alpha \cdot opt_{P_1}(\mathrm{I})$, e
- **se** $\left|opt_{P_1}(\mathrm{I}) - val_{P_1,\mathrm{I}}(g(I,S))\right| \leq \beta \cdot \left|opt_{P_2}(f(\mathrm{I})) - val_{P_2,f(\mathrm{I})}(S)\right|$

Fatos importantes:

- $\mathbf{P_1 \leq_{strict} P_2} \Rightarrow \mathbf{P_1 \leq_E P_2} \Rightarrow \mathbf{P_1 \leq_C P_2}$

  $\Downarrow$ $\Downarrow$

  $\mathbf{P_1 \leq_{ap} P_2} \Rightarrow \mathbf{P_1 \leq_{co} P_2}$

  $\Downarrow$

  $\mathbf{P_1 \leq_L P_2} \Rightarrow \mathbf{P_1 \leq_{ptas} P_2}$
- $\boxed{\mathbf{P_1 \leq_L P_2} \text{ e } \mathbf{P_1 \in APX} \Rightarrow \mathbf{P_1 \leq_{ap} P_2}}$

PROVA: $\mathbf{P_1 \leq_L P_2}$ e $\mathbf{P_1 \in APX \Rightarrow P_1 \leq_{ap} P_2}$

- $P_1 \leq_L P_2 \Rightarrow$ existem funções f e g polinomiais.
- $P_1 \leq_L P_2 \Rightarrow opt_2 \leq \alpha \cdot opt_1$ e $|opt_1 - val_1| \leq \beta \cdot |opt_2 - val_2|$.
- $P_1 \in APX \Rightarrow$ existe algoritmo $\mathcal{A}_1$ poli $(\gamma_{\geq 1})$-aprox.
- Se P_1 é de minimização:

$$\mathcal{R}_1 - 1 = \frac{val_1 - opt_1}{opt_1} \leq \beta \cdot \frac{|opt_2 - val_2|}{opt_2/\alpha} \leq \alpha\beta \cdot \left|1 - \frac{val_2}{opt_2}\right| \leq \alpha\beta \cdot (\mathcal{R}_2 - 1)$$

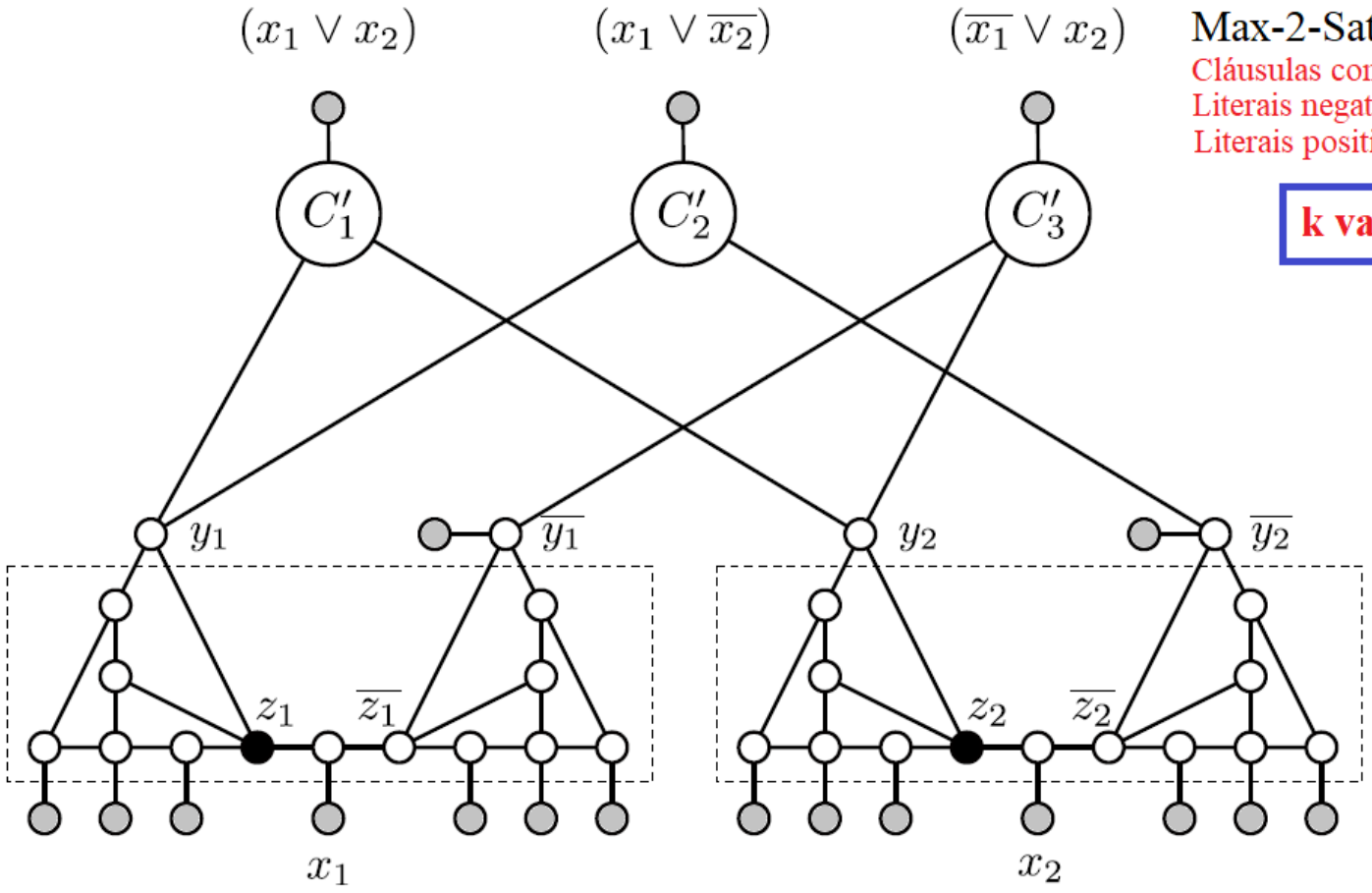
- Se P_1 é de maximização: tome $g'(S)$ o melhor entre $g(S)$ e $\mathcal{A}_1(\mathrm{I}_1)$

$$\mathcal{R}_1 - 1 = \frac{opt_1 - val_1}{val_1} \leq \frac{|opt_1 - val_1|}{opt_1/\gamma} \leq \alpha\beta\gamma \cdot (\mathcal{R}_2 - 1)$$

Redução AP: MaxCut $\leq_{ap}$ Max-3SAT

- Seja G instância de MaxCut. **Função f:** p/ cada vértice u e cada aresta e de G, crie variáveis x_u e x_e em Max-3SAT.
- Para cada aresta $e = uv$ de G, crie 5 cláusulas em Max-3SAT: $x_u \vee x_v \vee \overline{x_e}$, $\overline{x_u} \vee \overline{x_v} \vee \overline{x_e}$, $x_u \vee \overline{x_v} \vee x_e$, $\overline{x_u} \vee x_v \vee x_e$, x_e.
- **Função g:** Dada atribuição V ou F às variáveis em Max-3SAT, podemos obter uma "igual ou melhor" tomando $x_e = x_u \oplus x_v$ para cada aresta $e = uv$ (satisfazendo pelo menos 4 entre 5 cláusulas). Assuma então que tem esta forma "canônica". Seja (A, B) o seguinte corte em G: $u \in A$ se x_u é V; cc. $u \in B$.
- Note que x_e é V se e só se a aresta e passa o corte. Além disso, cada corte (A, B) em G define atribuição em Max-3SAT (função g^{-1}).
- Logo: $val_1 = val_2 - 4|E_G|$ e $opt_1 = opt_2 - 4|E_G|$.
- Pelo algoritmo MaxCut-Erdős: $opt_1 \geq |E_G|/2$.
- Portanto: $\mathbf{opt_2} = opt_1 + 4|E_G| \leq opt_1 + 8opt_1 = \mathbf{9 \cdot opt_1}$. $(\alpha = 9)$
- Além disso: $opt_1 - val_1 = opt_2 - val_2$. $(\beta = 1)$
- Então é **Redução L**. Como $MaxCut \in$ APX, então é **Redução AP**.

Redução AP: Max-2-SAT(3) $\leq_{ap}$ Min P_3-hull Set



Max-2-Sat(3) é APX-difícil

Cláusulas com no máximo 2 literais

Literais negativos em exatamente 1 cláusula

Literais positivos em 1 ou 2 cláusulas

k variáveis e m cláusulas

Grafo G construído é bipartido e tem **23k + 2m vértices**

Fórmula é satisfatível **se e só se** **h(G) = 9k + m**

Bom hull set S: vértices de grau 1, vértices de cláusulas e vértices de literais. $|S| = 9k + m + X(S)$

g(I, S) retorna atribuição dos vértices de literais. **X(S):** número de vértices de cláusulas.

g(I, S) satisfaz **m-X(S)** claúsulas.

$|S| - |S^*| = X(S)-X(S^*) = (m-X(S^*)) - (m-X(S))$

$|S^*| = 9k+m+X(S^*) \leqslant 9k+2m \leqslant 20m \leqslant 40\ (m-X(S^*))$, pois $k \leqslant 2m$ e $m-X(S^*) \geqslant m/2$

$\alpha=40, \beta=1$

Redução L