

MaNIACS: Approximate Mining of Frequent Subgraph Patterns through Sampling

PCC142 – Mineração de Dados

Prof: Anderson Almeida Ferreira

Aluno: Gabriel F. Costa



CSI-Lab

Sumário

01 INTRODUÇÃO & OBJETIVOS

02 REVISÃO BIBLIOGRÁFICA

03 MÉTODOS & MATERIAIS

04 RESULTADOS

05 CONCLUSÕES

1) Introdução e objetivos

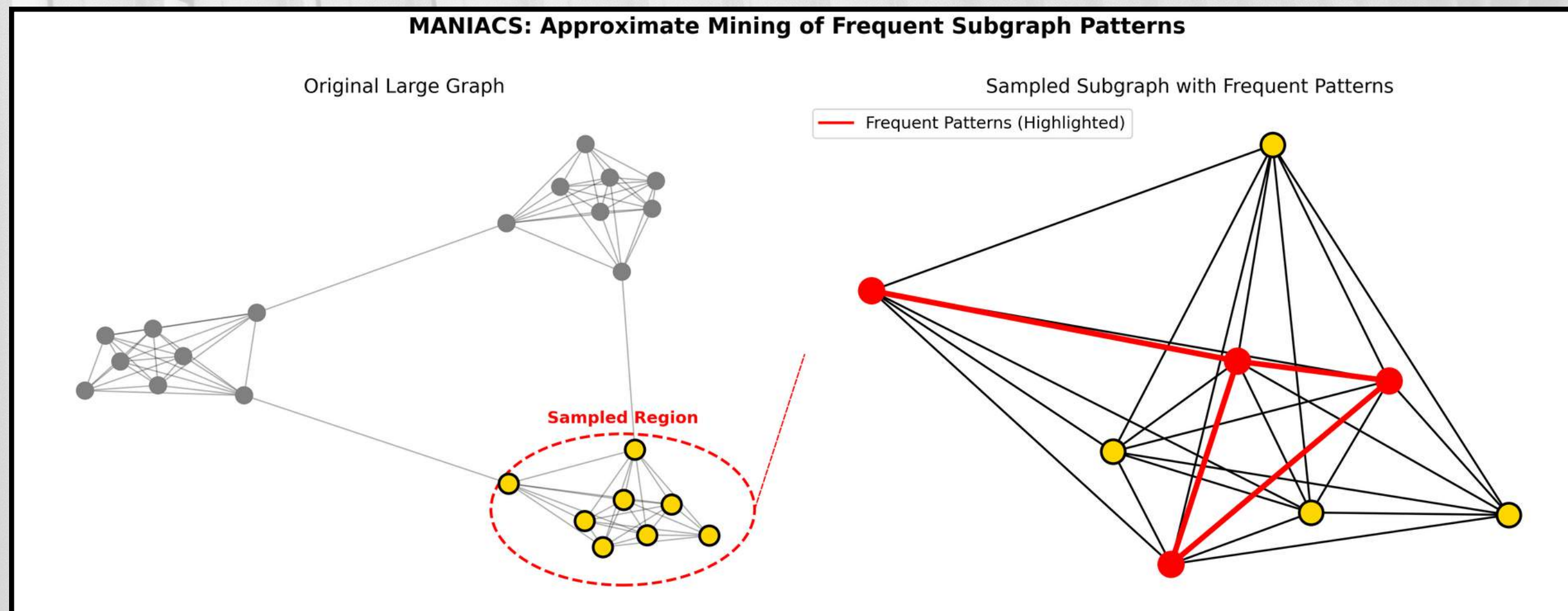
Problema:

Encontrar subgrafos frequentes em redes grandes é computacionalmente inviável com métodos exatos devido à:

- Explosão combinatória de padrões;
- Complexidade NP do isomorfismo de subgrafos.

Exemplo prático:

- Gerar um conjunto de subgrafos isomórficos de uma rede social com milhões de nós
 - Algoritmos exatos falham em escalar para grafos com $>1M$ vértices



Solução proposta:

MaNIACS é um **algoritmo** que **minera padrões frequentes** em **grafos grandes** por meio de **amostragem de vértices**, com garantias estatísticas. Ele evita explorar padrões irrelevantes e reduz o **espaço de busca** com **podas** eficientes no mesmo.

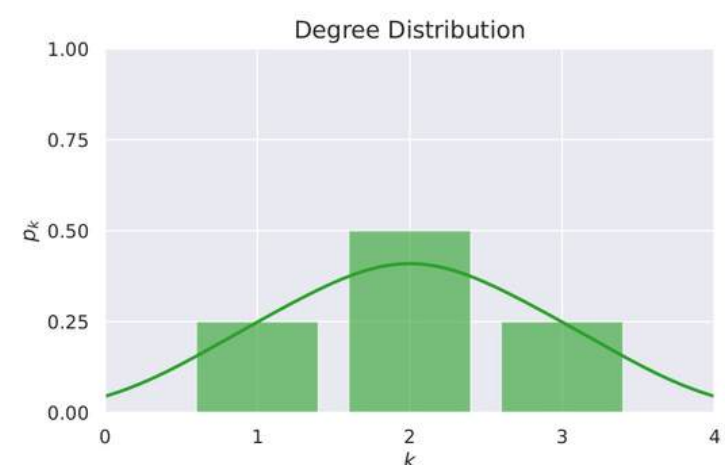
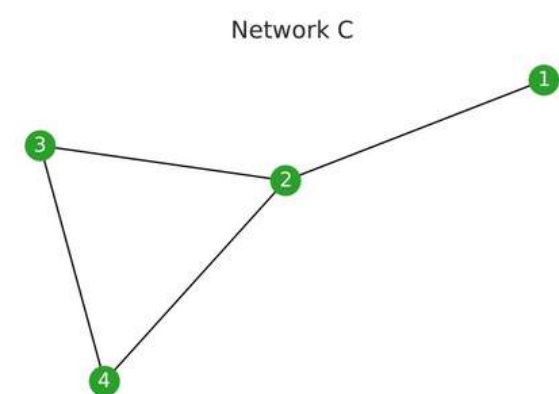
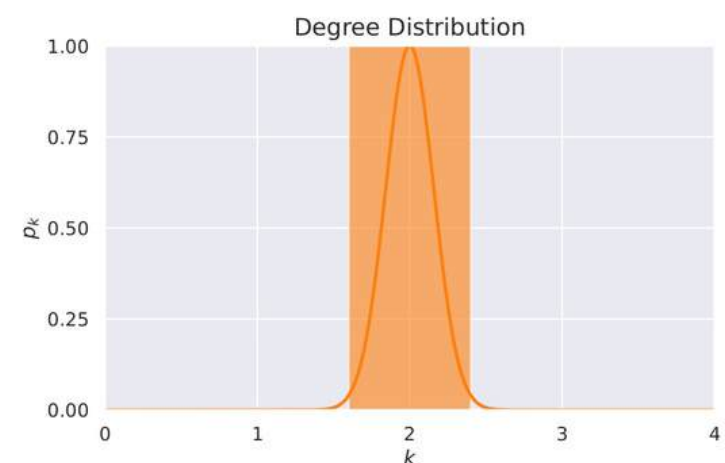
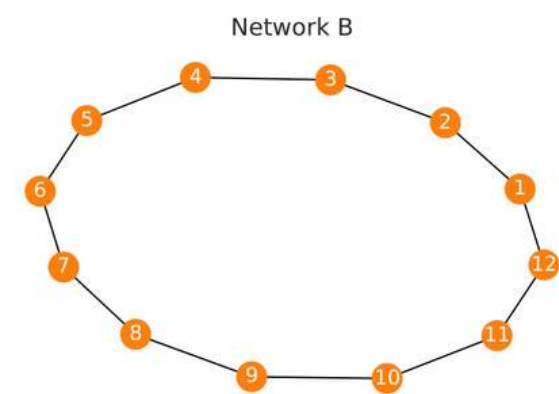
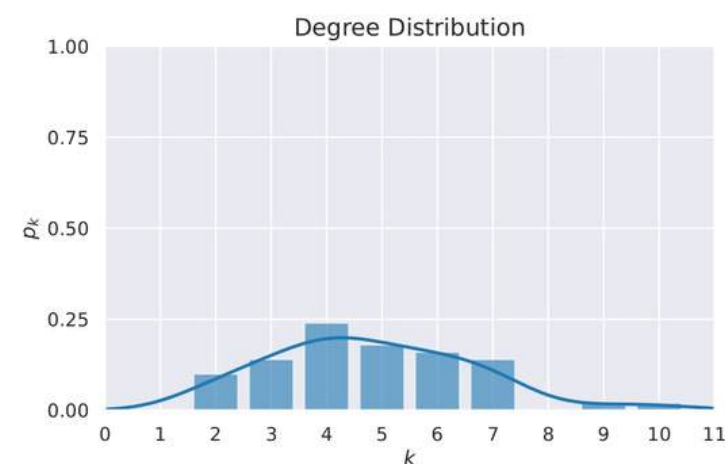
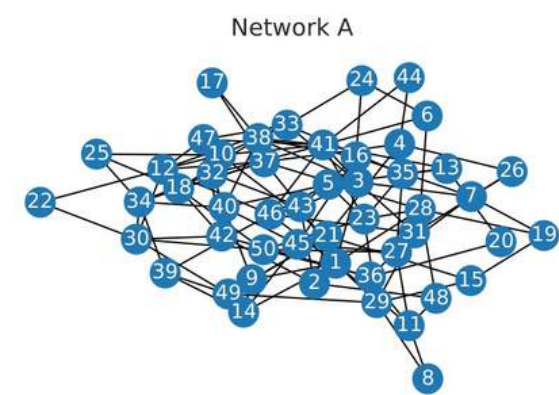
Objetivos da pesquisa:

- Propor um algoritmo eficiente baseado em amostragem.
 - Estimar a frequência MNI (minimum node image-based) com controle de erro.
 - Reduzir o tempo de execução sem perder qualidade.
 - Utilizar a anti-monotonicidade da MNI para poda.
- Aplicar a dimensão VC empírica para garantir limites de erro.

Hipóteses

- A amostragem de vértices permite estimativas precisas da MNI.
- A MNI é adequada para grafos grandes e permite poda eficiente.
 - A dimensão VC empírica é válida para esse contexto.
- Grafos grandes viabilizam bons resultados via amostragem.
 - Órbitas eliminam redundâncias e otimizam a busca.

2) Revisão: Grafos



Sumarização matemática para um grafo $G(V, E, L)$:

$$G = (V, E, \mathcal{L})$$

$$K = |V|$$

$$L = |E|$$

$$\mathcal{L} = \{\lambda_1, \dots, \lambda_K\}$$

Grafos considerados e suposições do artigo:

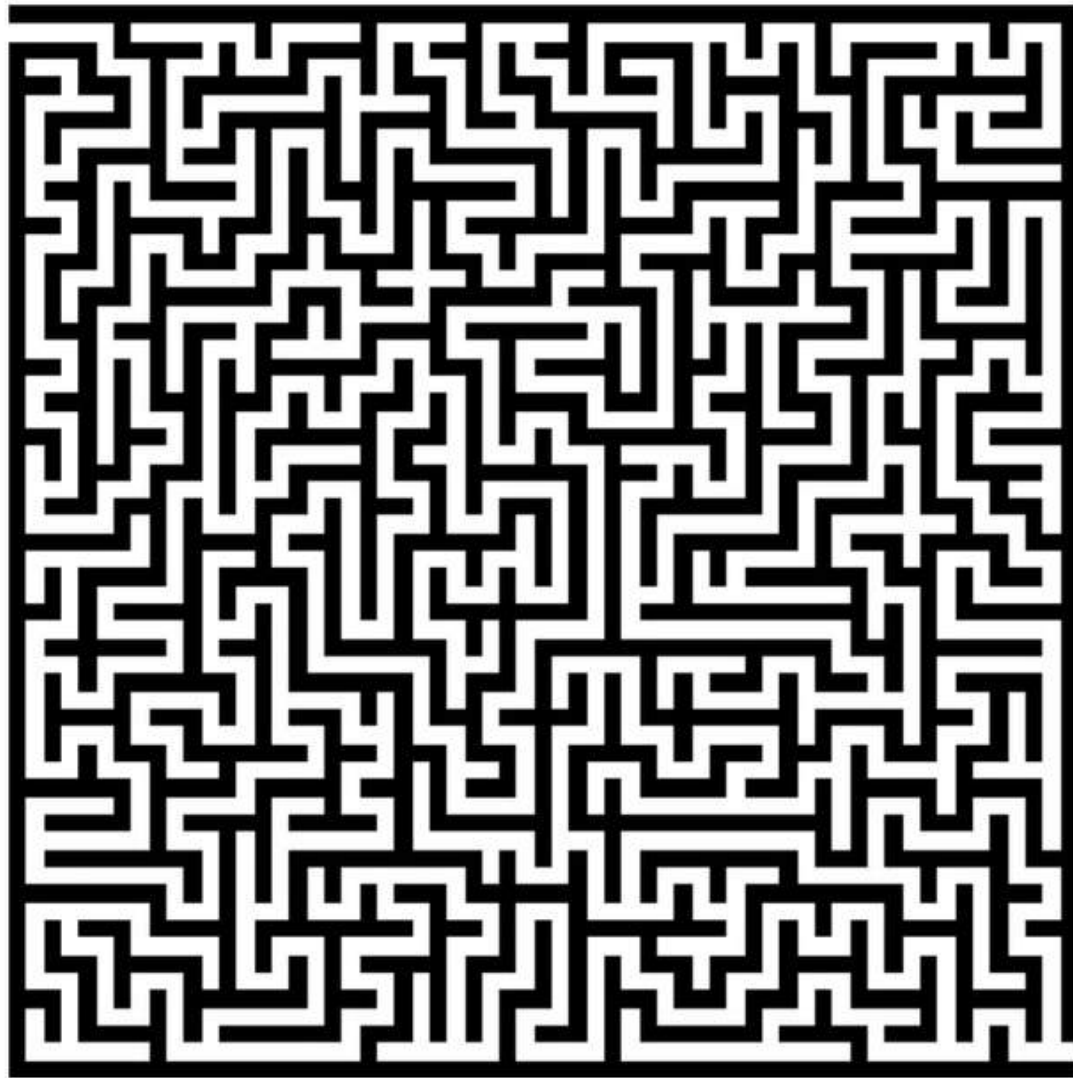
- **Simples:** Sem loops ou arestas múltiplas não-direcionadas e sem pesos.
- **Rótulos em vértices:** $G = (V, E, \mathcal{L})$ onde:
 $\mathcal{L}: V \rightarrow \{\lambda_1, \dots, \lambda_K\}$.
- **Conectados:** Caminho entre qualquer par de vértices.

2) Revisão: Caminhos

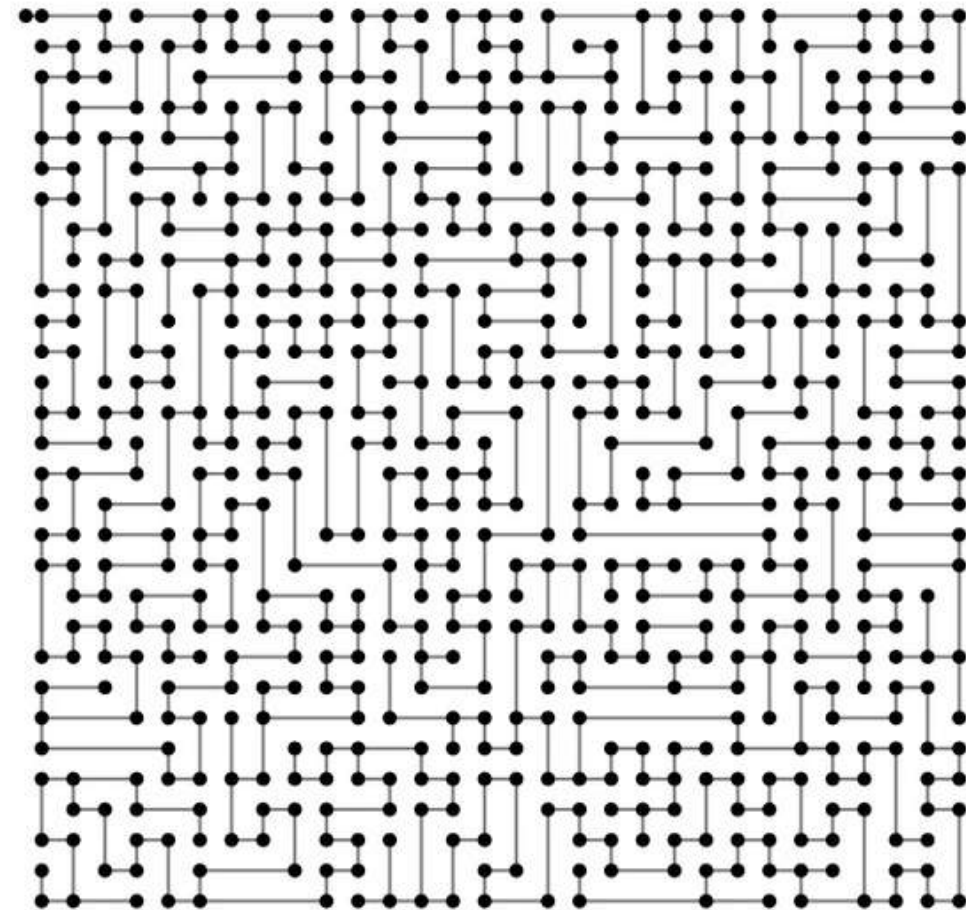
Caminhos:

Sequências de vértices conectados por arestas. Podem ser usados para definir padrões (ex.: cadeias de interações).

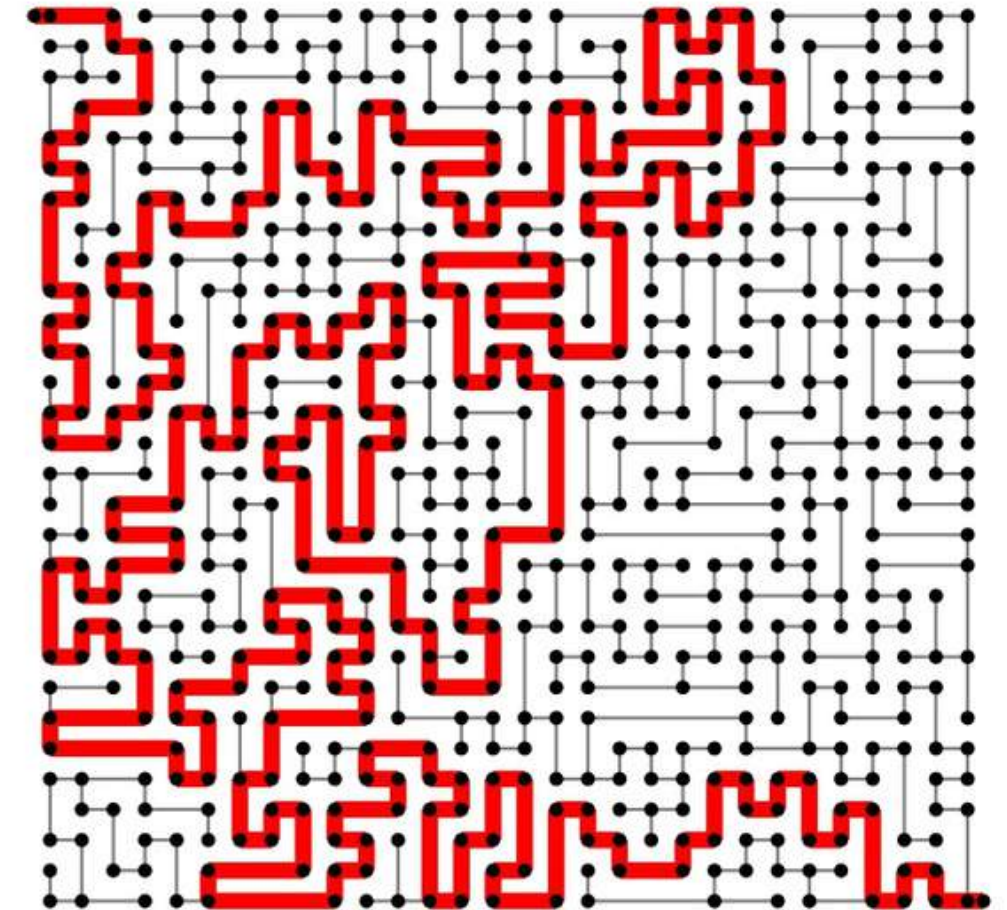
Maze



Maze as a Network



Shortest Path



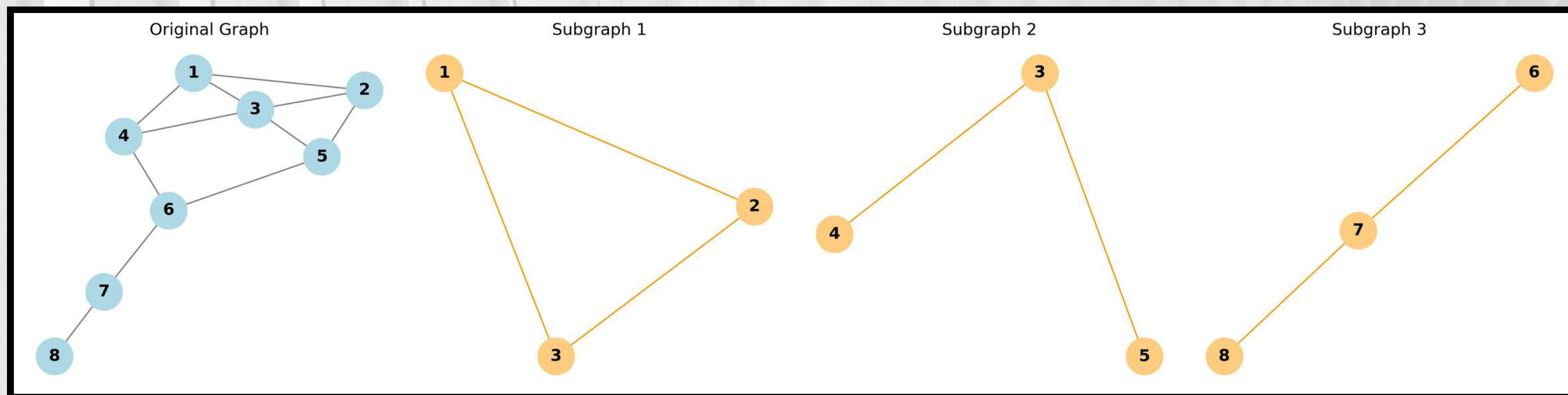
2) Revisão: Subgrafos

Definição: Dado um grafo $G=(V,E)$ e um subconjunto $S \subseteq V$, o subgrafo induzido G_S é definido como: tal que:

$$G_S \doteq (S, E(S)), \quad \text{onde} \quad E(S) \doteq \{(u, v) \in E : u, v \in S\}.$$

Para um grafo G e um tamanho máximo K , o conjunto \mathcal{C} contém todos os subgrafos conectados induzidos de G com até K vértices:

$$\mathcal{C} \doteq \{\text{subgrafos induzidos } G_S \text{ de } G : |S| \leq k \text{ e } G_S \text{ é conexo}\}.$$



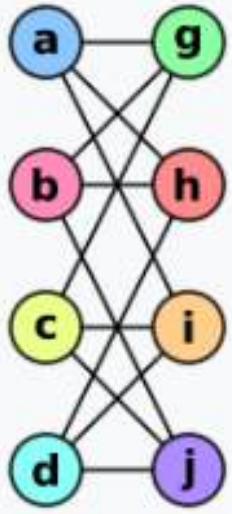
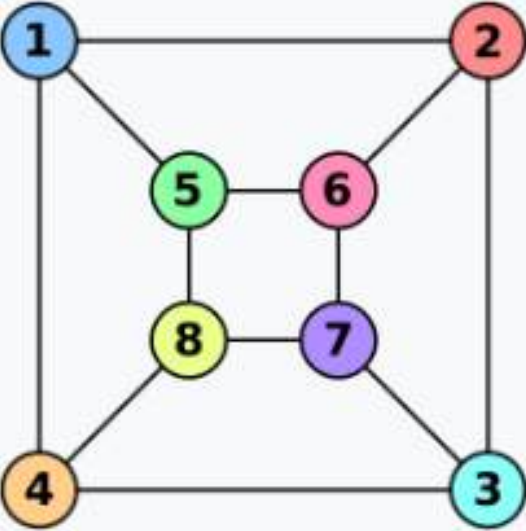
2) Revisão: Isomorfismo de grafos

Isomorfismo: Dois grafos $G'=(V', E', L')$ e $G''=(V'', E'', L'')$ são isomorfos se existe uma bijeção $\mu : V' \rightarrow V''$ tal que:

$$(u, v) \in E' \iff (\mu(u), \mu(v)) \in E''$$

e a rotulagem dos vértices é preservada:

$$L'(u) = L''(\mu(u)), \quad \forall u \in V'.$$

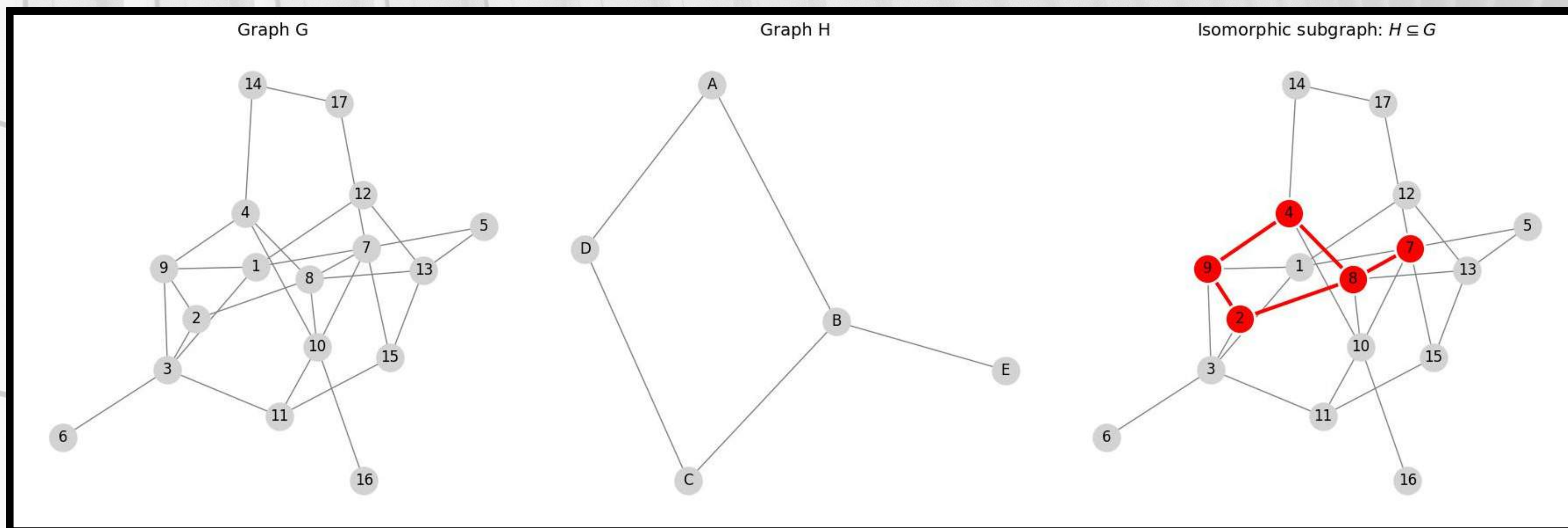
Grafo G	Grafo H	Um isomorfismo entre G e H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

2) Revisão: Subgrafos isomórficos

Um padrão **P** é um pequeno grafo conectado e rotulado (com até **K** vértices). O artigo busca encontrar padrões que apareçam frequentemente como subgrafos em **G**.

Ocorrência de um padrão: Um subgrafo induzido **G_S ∈ C** é isomórfico a **P** se existe um isomorfismo **μ** que preserve arestas e rótulos:

$$\forall u, v \in V_P, \quad (u, v) \in E_P \iff (\mu(u), \mu(v)) \in E(S).$$



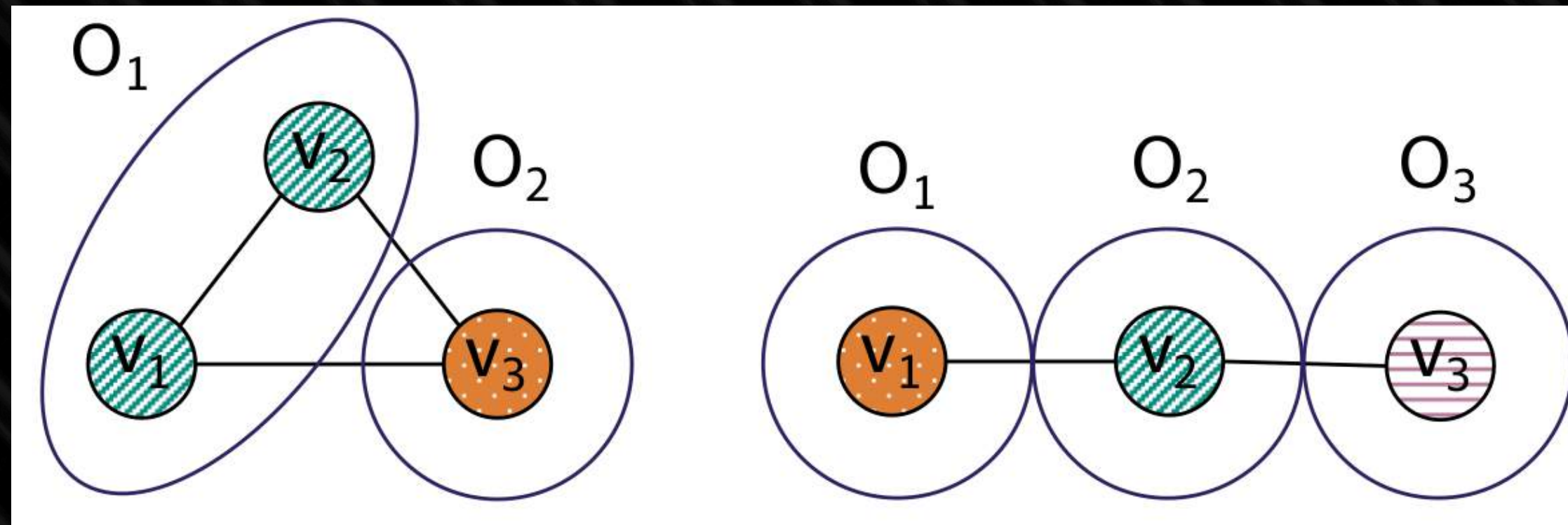
2) Revisão: Órbitas

Órbitas:

Partições de vértices em um padrão que são equivalentes sob **automorfismos (simetrias)**.

Exemplo:

1. Em um triângulo, todos os vértices pertencem à mesma órbita.
2. Em um grafo linear (caminho), vértices centrais e extremos têm órbitas distintas.



O conjunto de imagem **Zs**, representa os vértices de **S** mapeados para a órbita **A** em alguma ocorrência de **P**:

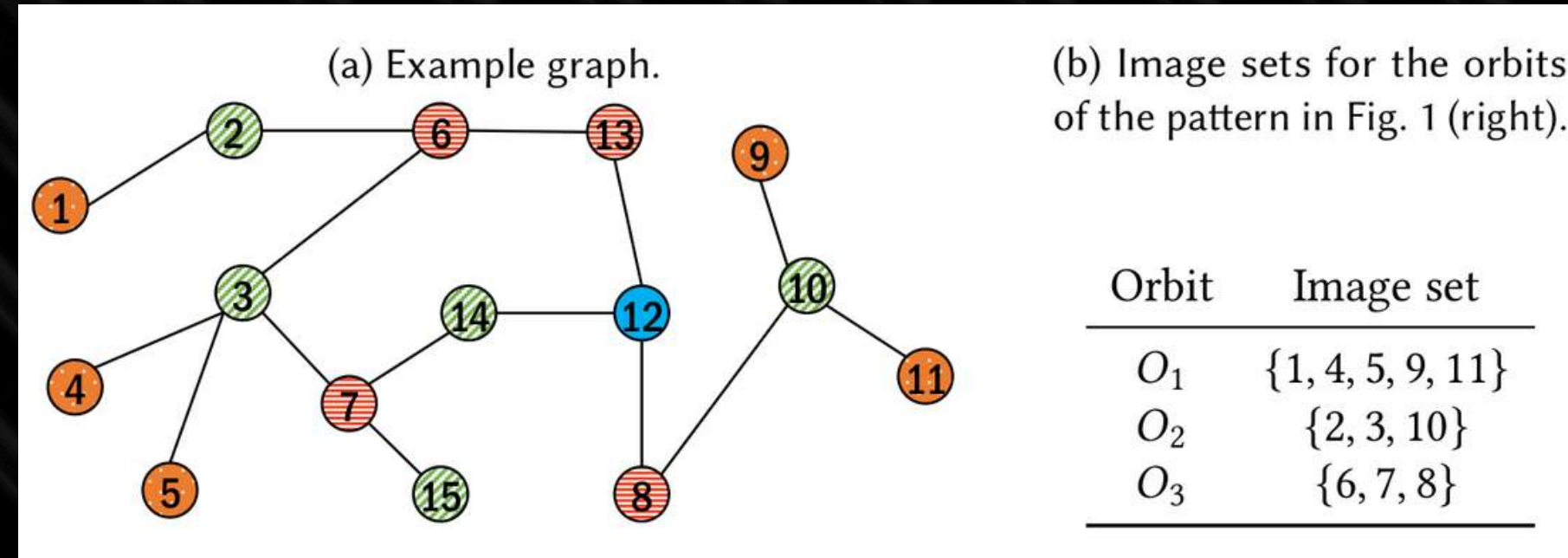
$$Z_S(A) = \{v \in S \mid \exists \mu : (V', E') \rightarrow P, (V', E') \in C, v \in V', \mu(v) \in A\}$$

$$\begin{array}{l} \text{P/ as órbitas do grafo} \\ \text{a esquerda} \rightarrow \end{array} \quad \begin{array}{l} Z_S(O_1) = \{1, 2\} \\ Z_S(O_2) = \{3\} \end{array}$$

Órbitas definem a **MNI**, que mede a **frequência de padrões sem redundância** e permite **podar o espaço de busca com eficiência**, graças à sua **anti-monotonicidade**.

2) Revisão: Frequência MNI

O **MNI-frequency** de um padrão **P** é a menor proporção de vértices do grafo **G** que são mapeados para uma órbita de **P** por algum **isomorfismo de subgrafo**.



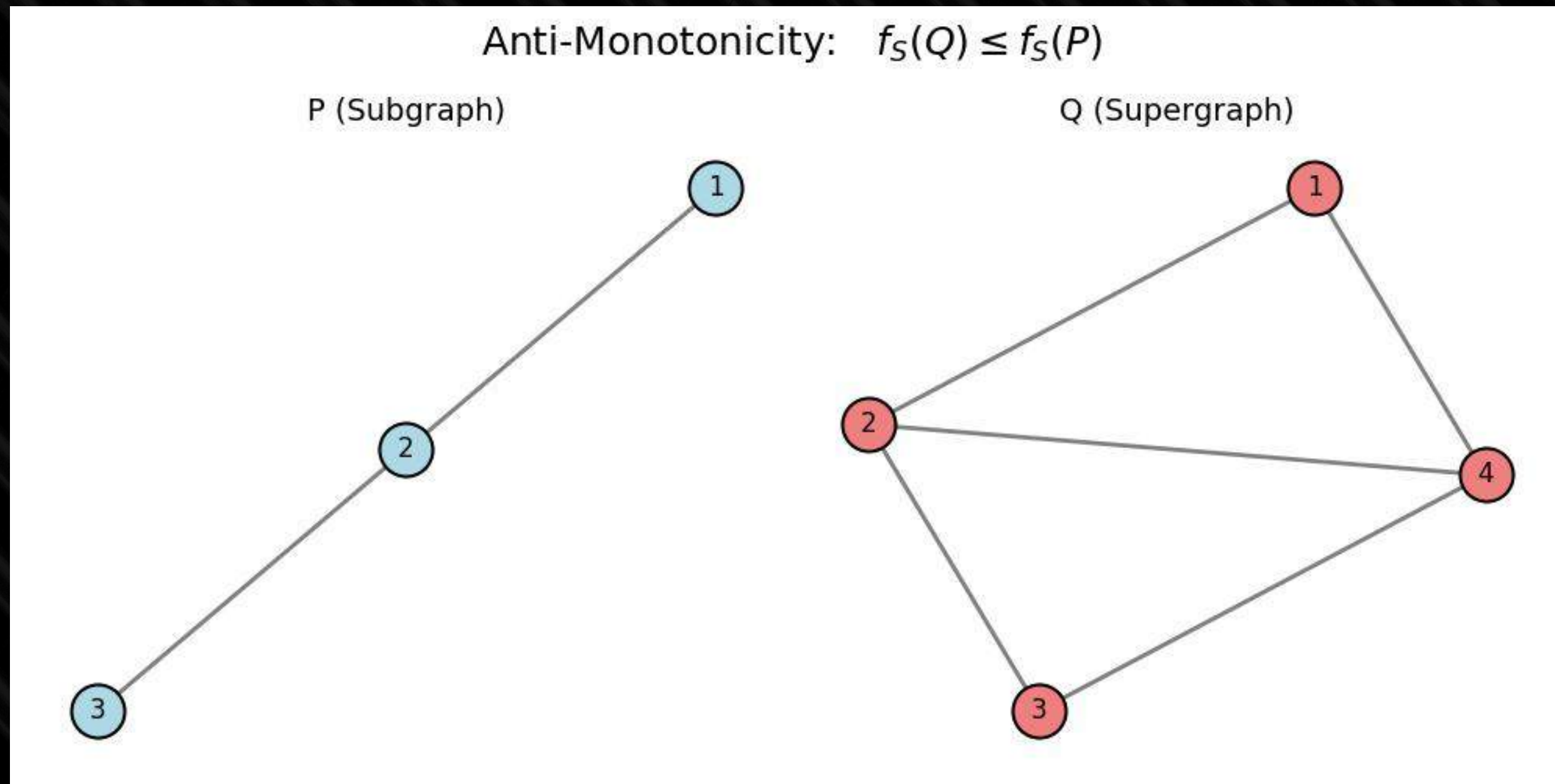
Frequência MNI } $f_S(P) = \min \left\{ \frac{|Z_S(A)|}{|S|} \right\},$ onde A é uma órbita de P

$$f_S(P) = \min \left\{ \frac{5}{15}, \frac{3}{15}, \frac{3}{15} \right\} = \frac{1}{5} = 0,2$$

$$MNI = 0,2.$$

2) Revisão: Anti-Monotonicidade da Frequência MNI

Uma propriedade crucial para a **poda eficiente** do **espaço de busca**:



Isso permite descartar padrões grandes se seus subgrafos **já forem infrequentes**.

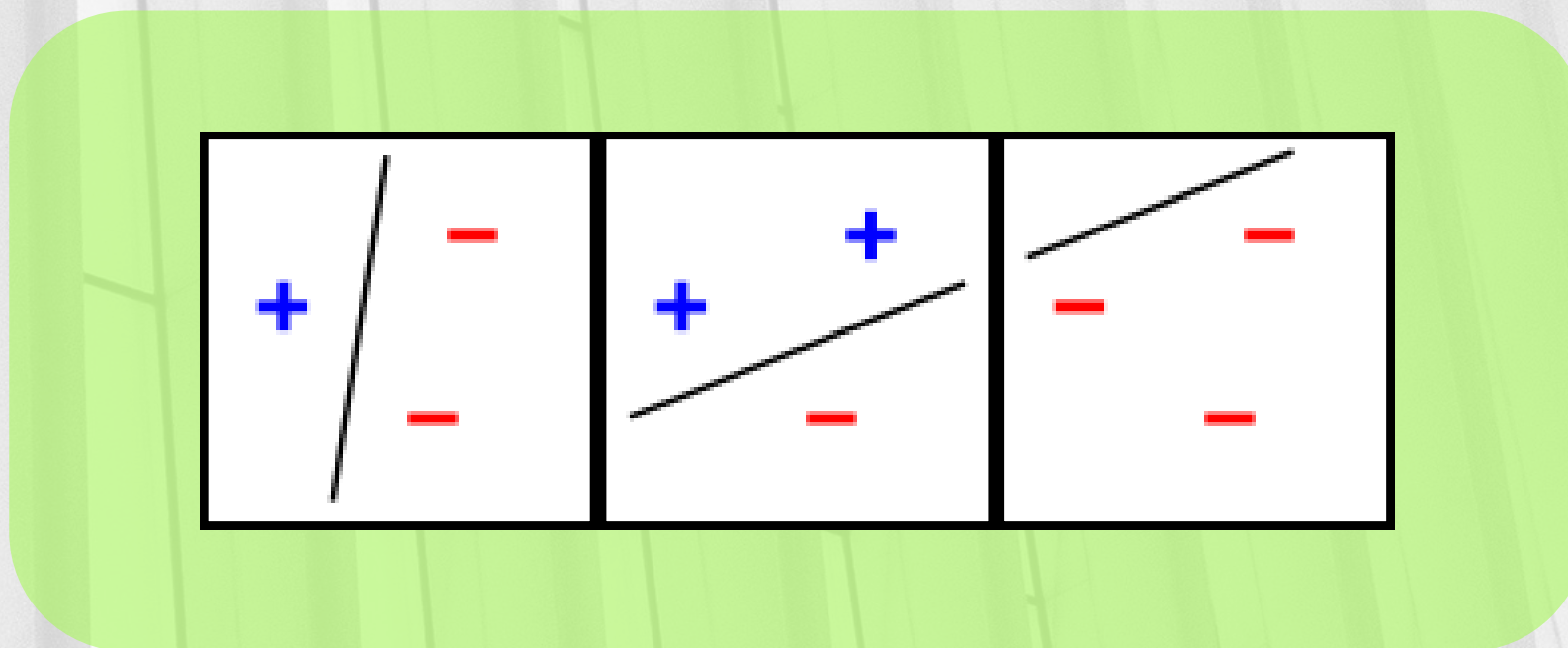
2) Revisão: Dimensão VC Empírica

Definição: O que é a Dimensão VC Empírica (eVC)?

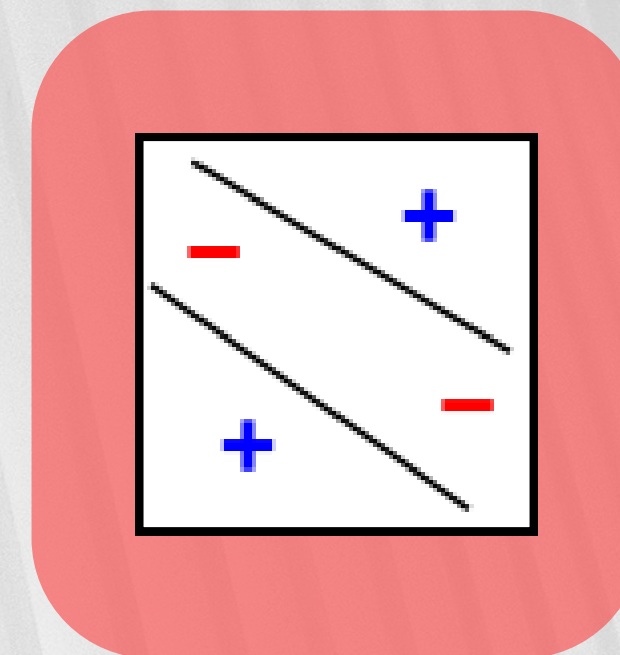
- Dado um espaço de alcance (D, R) e uma **amostra** $S \subseteq D$, a **dimensão VC empírica** $Es(R)$ é o maior subconjunto de S que pode ser fragmentado (shattered) por R .
- Um conjunto $A \subseteq S$ é fragmentado se todas as suas $2^{|A|}$ **subpartes** podem ser obtidas pela interseção de A com os alcances $R \in R$.

Exemplo:

- Se R é o conjunto de intervalos dos reais, a dimensão VC é 2 (não é possível fragmentar 3 pontos alinhados).



3 pontos fragmentados



4 pontos impossível

2) Revisão: Amostragem para aproximação de subgrafos frequentes

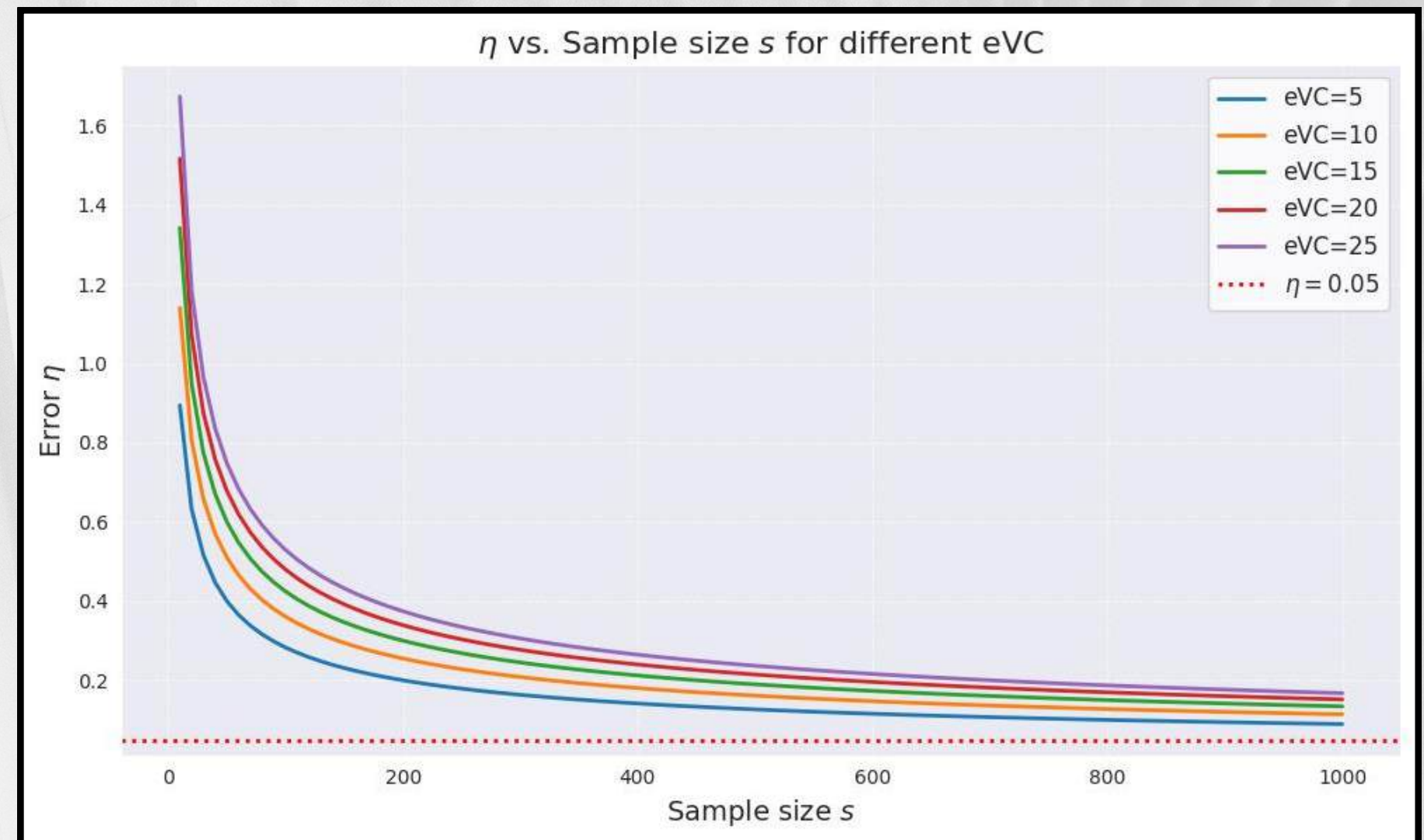
Definição: Amostras η (η -samples).

- Uma amostra $S \subseteq D$ é uma **η -amostra** para (D, R) se, para todo $R \in \mathcal{R}$ (garantindo que as frequências estimadas em S estão η -próximas das verdadeiras frequências em D):

$$\left| \frac{|R|}{|D|} - \frac{|R \cap S|}{|S|} \right| \leq \eta.$$

Teorema-Chave: Se S é uma amostra aleatória de tamanho s , então, com **probabilidade $1-\delta$** :

$$\eta = \sqrt{\frac{c(E_S(\mathcal{R}) + \ln(1/\delta))}{s}}.$$



3) Métodos & Materiais



3) M&M: Algoritmos apresentados ao problema proposto:

Algoritmo	Descrição
MaNIACS	Laço principal que itera pelos níveis $i=1 \dots k$, coordenando o processo de mineração.
getImageSets	Função que calcula os conjuntos de imagem Z_i para cada órbita dos padrões em H_i .
getEVCBound	Calcula o upper bound da dimensão VC (eVC) com base nos conjuntos de imagem.

Main

Auxiliares

O algoritmo em si:

Algorithm 1: MANIACS

Input: Graph $G = (V, E)$, maximum pattern size k , frequency threshold τ , sample size s , failure probability δ

Output: A set Q with the properties from Thm. 4.7

```
1  $S \leftarrow \text{drawSample}(V, s)$ 
2  $Q \leftarrow \emptyset; i \leftarrow 1$ 
3  $\mathcal{H}_1 \leftarrow \{P \in \mathcal{P} : P \text{ has a single vertex}\}$ 
4 while  $i \leq k$  and  $\mathcal{H}_i \neq \emptyset$  do
5    $\mathcal{Z}_i \leftarrow \text{getImageSets}(\mathcal{H}_i, S, \tau)$ 
6   do
7      $b_i^* \leftarrow \text{getEVCBound}(\mathcal{Z}_i)$ 
8      $\varepsilon_i \leftarrow \text{getEpsilon}(b_i^*, \delta/k)$ 
9      $\mathcal{H}'_i \leftarrow \mathcal{H}_i$ 
10     $\mathcal{H}_i \leftarrow \{P \in \mathcal{H}_i : f_S(P) \geq \tau - \varepsilon_i\}$ 
11    while  $\mathcal{H}'_i \neq \mathcal{H}_i$  and  $\mathcal{H}_i \neq \emptyset$ 
12       $Q \leftarrow Q \cup \{(P, f_S(P), \varepsilon_i) : P \in \mathcal{H}_i\}$ 
13      if  $i < k$  then  $\mathcal{H}_{i+1} \leftarrow \text{createChildren}(\mathcal{H}_i, \mathcal{Z}_i)$ 
14     $i \leftarrow i + 1$ 
15 return  $Q$ 
```

Auxiliar 1

Algorithm 2: GETIMAGESETS

Input: Set of patterns \mathcal{H}_i , sample S , frequency threshold τ

Output: The image sets \mathcal{Z}_i of the patterns in \mathcal{H}_i

```
1  $\mathcal{Z}_i \leftarrow \emptyset$ 
2 foreach  $P \in \mathcal{H}_i$  do
3   foreach orbit  $A$  of  $P$  do
4      $n \leftarrow$  a vertex of  $P$  in  $A$ 
5      $S_A \leftarrow$  vertices in  $S$  with the label of  $A$ 
6      $\mathcal{Z}_S(A) \leftarrow \emptyset$ ,  $remain \leftarrow |S_A|$ 
7     foreach  $v \in S_A$  do
8        $M \leftarrow \emptyset$ ;  $M[n] \leftarrow v$ 
9       if existsIsomorphism( $P, M$ ) then
10         $\mathcal{Z}_S(A) \leftarrow \mathcal{Z}_S(A) \cup \{v\}$ 
11         $remain \leftarrow remain - 1$ 
12        if ( $remain + |\mathcal{Z}_S(A)|$ )/ $|S| < \tau - \varepsilon_i$  then
13           $\text{prune } P$  and go to next pattern
14    $\mathcal{Z}_i \leftarrow \mathcal{Z}_i \cup \{\mathcal{Z}_S(A)\}$ 
15 return  $\mathcal{Z}_i$ 
```

Auxiliar 2

Algorithm 3: getEVCBound

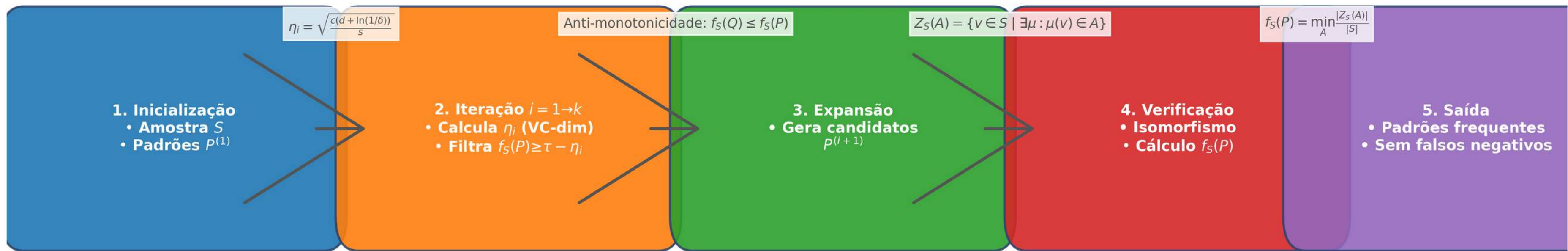
Input: Bag \mathcal{Z}_i of image sets $Z_A(S)$, \forall orbit A of each pattern in \mathcal{H}_i

Output: A value $b_i^* \geq E_S(\mathcal{R}_i)$

```
1 foreach  $\lambda \in L$  do
2    $D_\lambda \leftarrow$  set of image sets in  $\mathcal{Z}_i$  of orbits of vertices with label  $\lambda$ 
3    $M \leftarrow |S|$ -vector with element  $(v, |\{Z \in D_\lambda : v \in Z\}|)$ ,  $\forall v \in S$ 
4   sort  $M$  in decreasing order of the 2nd component
   // Denote with  $(v_i, q_i)$  the  $i$ -th element of  $M$ 
5    $g_\lambda^* \leftarrow \max\{g : v_g \geq 2^{g-1}\}$ 
6    $\gamma \leftarrow \max\{i : v_i > 2^{g_\lambda^*-1}\}$ 
7   if  $\nexists Q \subseteq \{v_1, \dots, v_\gamma\}$ ,  $|Q| = g_\lambda^*$ , s.t.  $\exists Z \in D_\lambda$  s.t.  $Q \subseteq Z$  then  $g_\lambda^* \leftarrow g_\lambda^* - 1$ 
8    $N \leftarrow |D_\lambda|$ -vector with element  $|Z|$ ,  $\forall Z \in D_\lambda$ 
9   sort  $N$  in decreasing order
   // Denote with  $a_i$  the  $i$ -th element of  $N$ 
10   $h_\lambda^* \leftarrow \min\{a_1, \lfloor \log_2(|D_\lambda| + 1) \rfloor\}$ 
11  while  $h_\lambda^* > 1$  do
12    foreach  $j \in \{0, \dots, h_\lambda^* - 1\}$  do  $c_j \leftarrow \sum_{z=0}^j \binom{h_\lambda^*}{z}$ 
13    if  $\nexists j \in \{0, \dots, h_\lambda^* - 1\}$  s.t.  $a_{c_j} < h_\lambda^* - j$  then break
14    else  $h_\lambda^* \leftarrow h_\lambda^* - 1$ 
15 return  $\max_{\lambda \in L} \min\{g_\lambda^*, h_\lambda^*\}$ 
```

3) M&M: Passo a passo dos algoritmos que compõem o MaNIACS

Fluxo do Algoritmo MANIACS: Mineração Aproximada de Subgrafos Frequentes



Baseado no princípio VC-dimension com garantias teóricas | Amostragem adaptativa por nível | Poda eficiente via anti-monotonicidade

MaNIACS é um conjunto de algoritmos **Apriori-like**: gera padrões maiores a partir de menores com poda antimonotônica em cada nível, reduzindo o **espaço de busca**.

3) M&M: Pruning

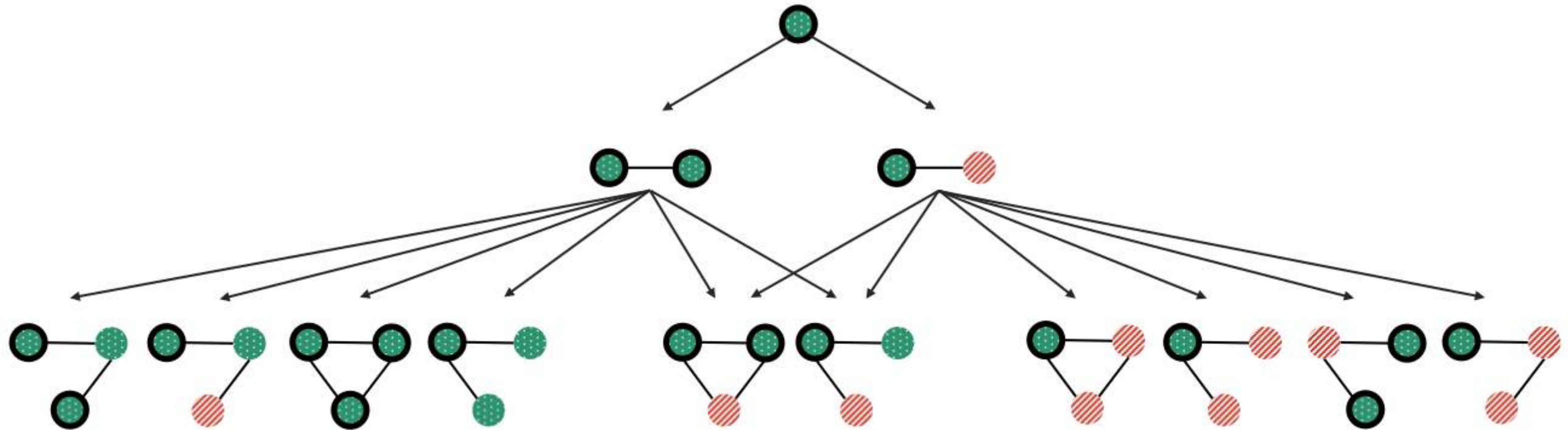


Fig. 3. Examples of parent-child relations for orbits (labels represented as colors). We represent each orbit by using its pattern with the vertices of the orbit in a thicker border.

- Poda agressiva \Rightarrow menor $\eta_i \Rightarrow$ mais poda (ciclo virtuoso).
- **Amostras concêntricas:**
Usa tamanhos de amostra decrescentes ($s_1 > s_2 > \dots > s_k$) para otimizar tempo.
- **Pré-poda:**
Descarta padrões cujas órbitas têm imagem $\mathbf{Z}_s(\mathbf{A})$ pequena antes de calcular isomorfismos.

3) M&M: Datasets

1. MiCo (Co-Autoria em Publicações Científicas)

Origem: Rede de coautoria de artigos científicos.

Características:

Tamanho:

- 100,000 vértices (autores).
- 1 milhão de arestas (colaborações).

- **Rótulos:** 29 categorias de áreas de pesquisa (ex: CS, Biology, Physics).

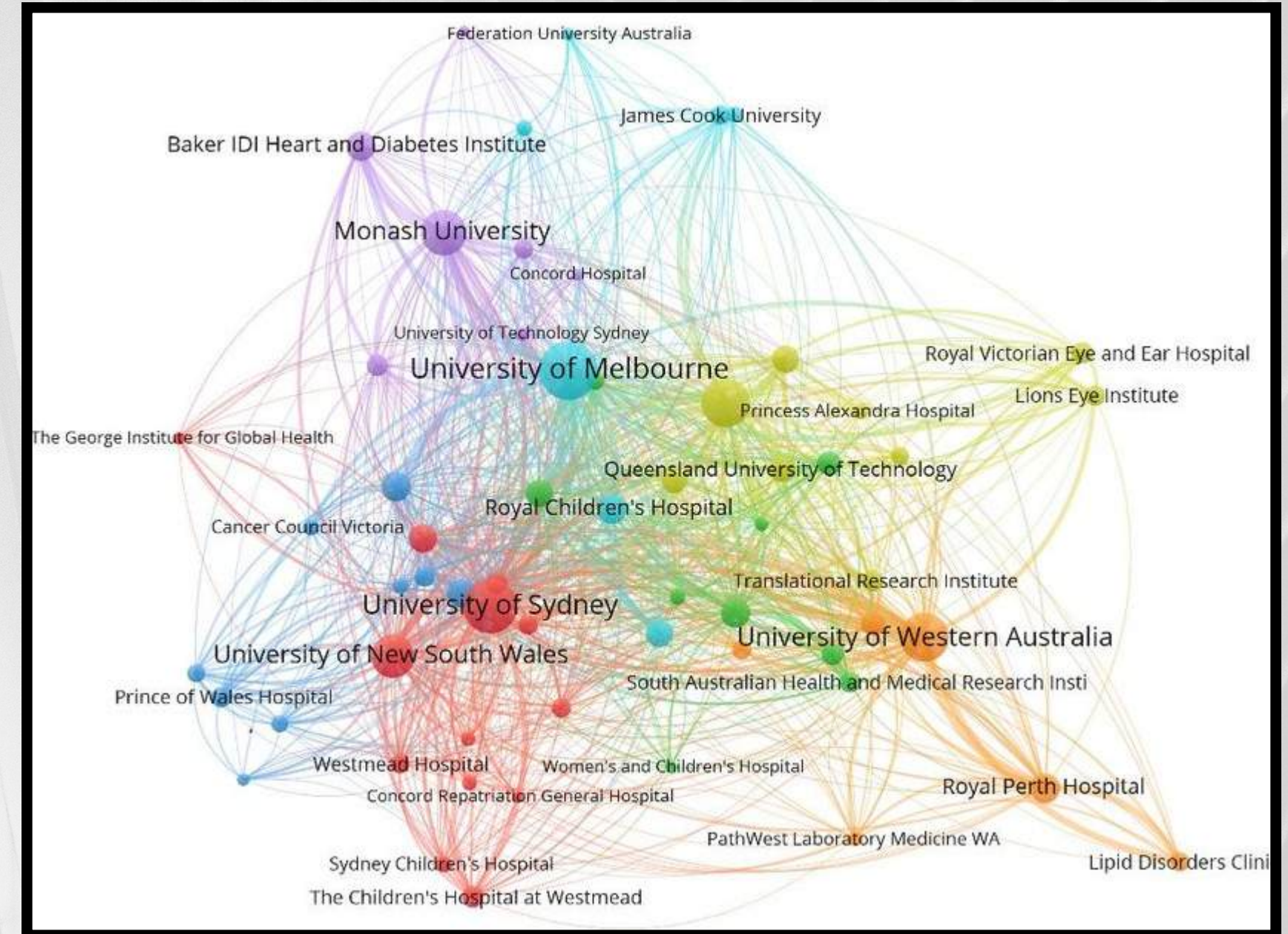
- **Densidade:** 2×10^{-4} (grafo esparso com clusters densos).

Por que foi incluído?

- Representa redes sociais acadêmicas com **comunidades bem definidas**.
- Alta variedade de rótulos (29) testa a **robustez à diversidade de categorias**.
- Exemplo concreto de aplicação: identificar padrões de **colaboração interdisciplinar**.

Pré-processamento específico:

- Fusão de autores homônimos usando ORCID.
- Exclusão de publicações com >50 autores (ruído).



3) M&M: Datasets

2. Patents (Rede de Citações de Patentes USPTO)

Origem: Citações entre patentes dos EUA (1975-1999).

Características:

Tamanho:

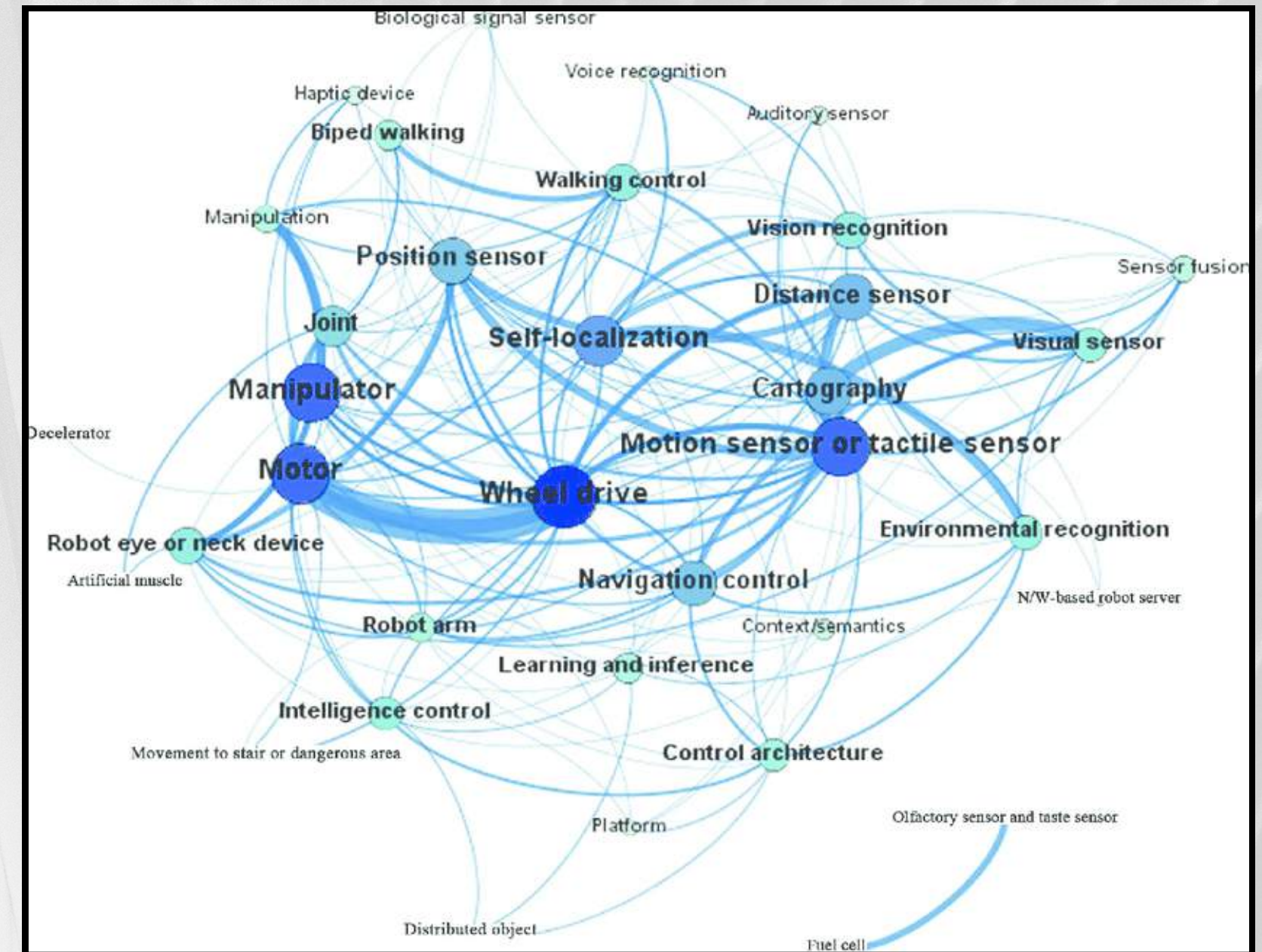
- 2.7 milhões de vértices (patentes) .
- 13 milhões de arestas (citações).
- **Rótulos:** 4 categorias (décadas: 70s, 80s, 90s).
- **Densidade:** 3.7×10^{-6} (grafo extremamente esparsos).

Propósito no estudo:

- Testa **escalabilidade extrema** (um dos maiores grafos disponíveis publicamente).
- Rótulos temporais permitem estudar **evolução de padrões**.
- Desafio único: **cadeias longas** de citações (até 30 passos).

Processamento especial:

- Conversão para grafo não-direcionado.
- Agregação de patentes por década de registro.



3) M&M: Datasets

3. YouTube (Rede de Vídeos Relacionados)

Origem: Relações "assistir-depois" entre vídeos.

Características:

Tamanho:

- 4.5 milhões de vértices (vídeos) .
- 43 milhões de arestas (relações).

- **Rótulos:** 12 categorias (ex: Educação, Games, Música).

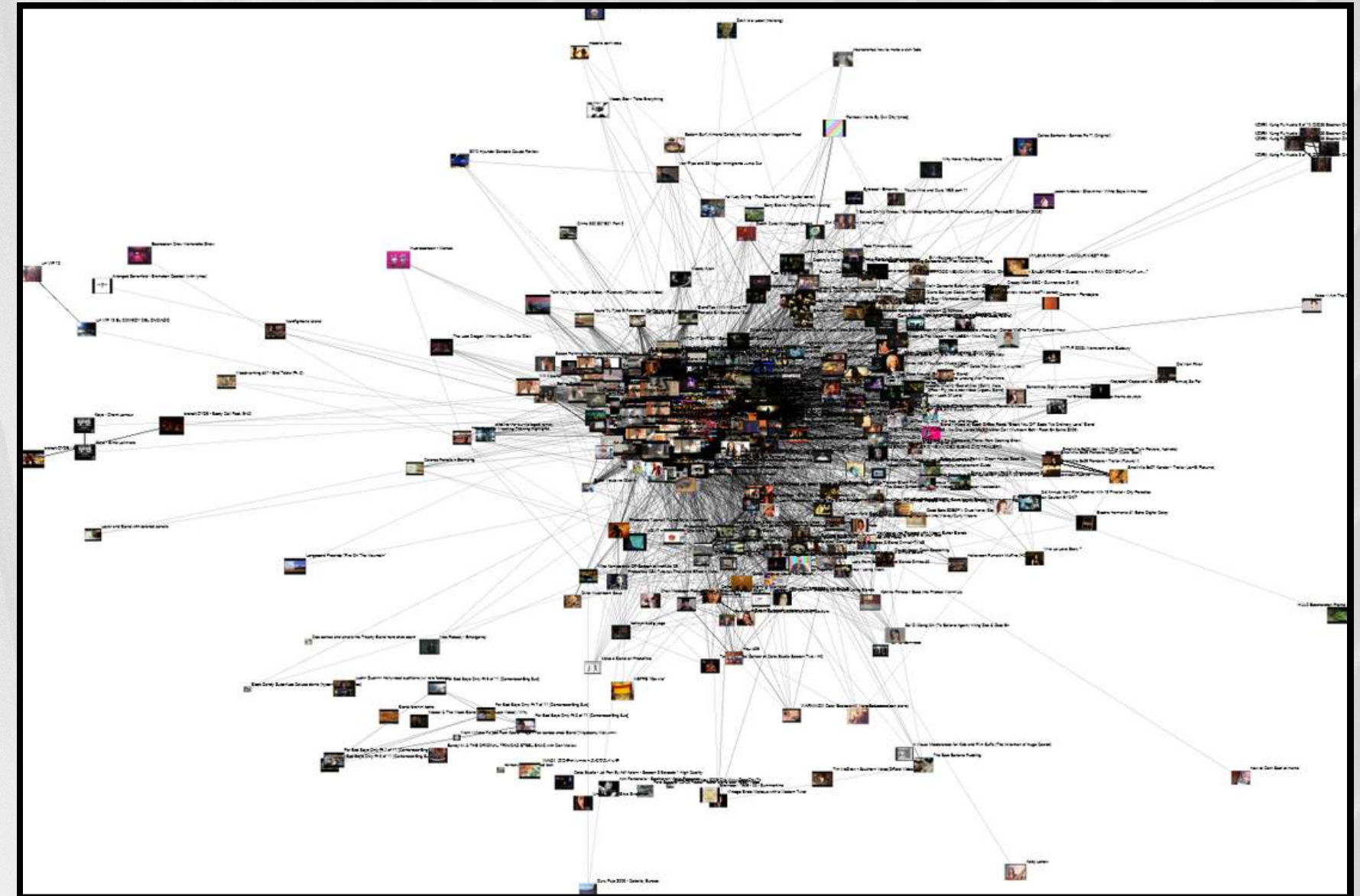
- **Densidade:** 4.2×10^{-6} (grafo extremamente esparsos).

Propósito no estudo:

- Modela **redes de mídia com comportamento de usuário real**.
- Rótulos heterogêneos testam **seleção de padrões semanticamente relevantes**.
- Exemplo de aplicação: recomendações de conteúdo baseadas em subgrafos frequentes.

Processamento especial:

- Filtragem de vídeos inativos (404).
- Normalização de categorias multi-idoma.



3) M&M: Métricas de avaliação

1. Métricas de qualidade da saída

1.1 Maximum Absolute Error (MaxAE)

$$\text{MaxAE} = \max_{P \in \mathcal{P}} |f_V(P) - f_S(P)|$$

Onde:

- **$f_v(\mathbf{P})$:** Frequência MNI exata no grafo completo.
 - **$f_s(\mathbf{P})$:** Frequência estimada na amostra.
- Objetivo:** Quantificar o pior erro de estimação.

1.2 Precisão e Recall

$$\text{Precision} = \frac{|\text{FP}_V(\tau) \cap Q|}{|Q|}, \quad \text{Recall} = \frac{|\text{FP}_V(\tau) \cap Q|}{|\text{FP}_V(\tau)|}$$

Onde:

- **Precision:** Precisão nos top-k padrões por frequência.
 - **Recall:** Frequência estimada na amostra.
- Objetivo:** Fração de padrões com **erro** $\leq \epsilon$.

3) M&M: Métricas de avaliação

2. Métricas de desempenho computacional

2.1 Tempo por nível (i)

$$T_i = t_{\text{pruning}} + t_{\text{extension}} + t_{\text{counting}}$$

Métricas derivadas:

-Speedup: $\frac{T_{\text{exato}}}{T_{\text{MANIACS}}}$

-Eficiência: $\frac{T_1}{\sum_{i=1}^k T_i}$

2.2 Uso de memória

$$M_{\text{peak}} = \max_{i \in 1..k} (|\mathcal{H}_i| \cdot \text{sizeof}(P) + |Z_i|)$$

3) M&M: Métricas de avaliação

3. Métricas de eficiência algorítmica

3.1 Razão de poda

$$\text{Pruning Ratio}_i = 1 - \frac{|\mathcal{H}_{i+1}|}{|\mathcal{H}_i|}$$

3.2 Amostragem efetiva

Utilização da Amostra =

$$\frac{1}{|S|} \sum_{v \in S} \mathbb{I}(v \text{ usado em algum } Z_S(A))$$

3) M&M: Métricas de avaliação

4. Métricas teóricas

4.1 Erro relativo

$$\text{Pruning Ratio}_i = 1 - \frac{|\mathcal{H}_{i+1}|}{|\mathcal{H}_i|}$$

4.2 Vapnik-Chervonenkis (VC) Efficiency

$$\text{VC Eff.} = \frac{\text{eVC observado}}{\text{eVC teórico}}$$

3) M&M: Métricas de avaliação

5. Métrica de estabilidade

5.1 Variância entre execuções

$$\sigma_{\text{freq}}^2 = \frac{1}{m} \sum_{j=1}^m (f_{S_j}(P) - \bar{f}_S(P))^2$$

4) Resultados

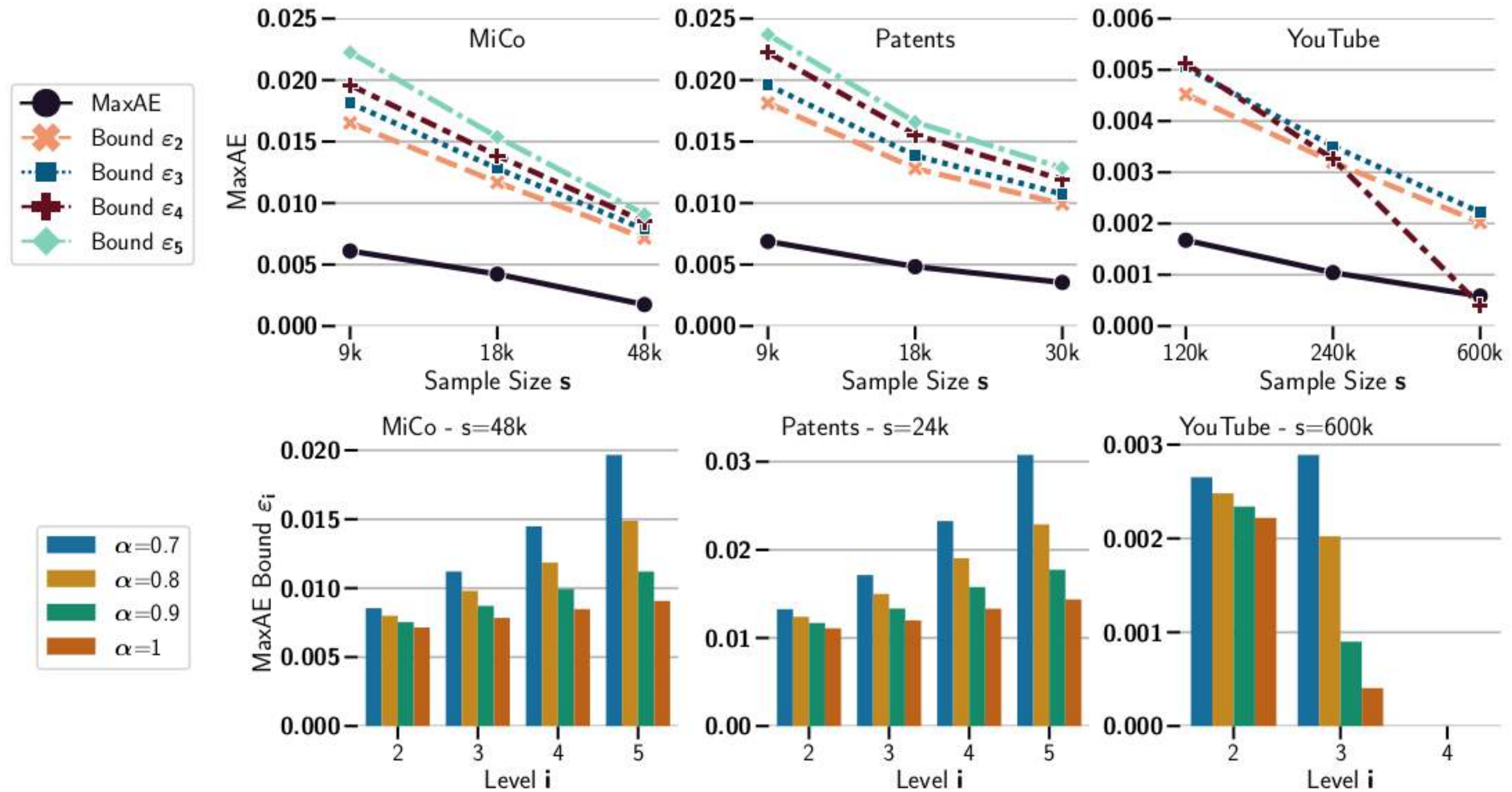
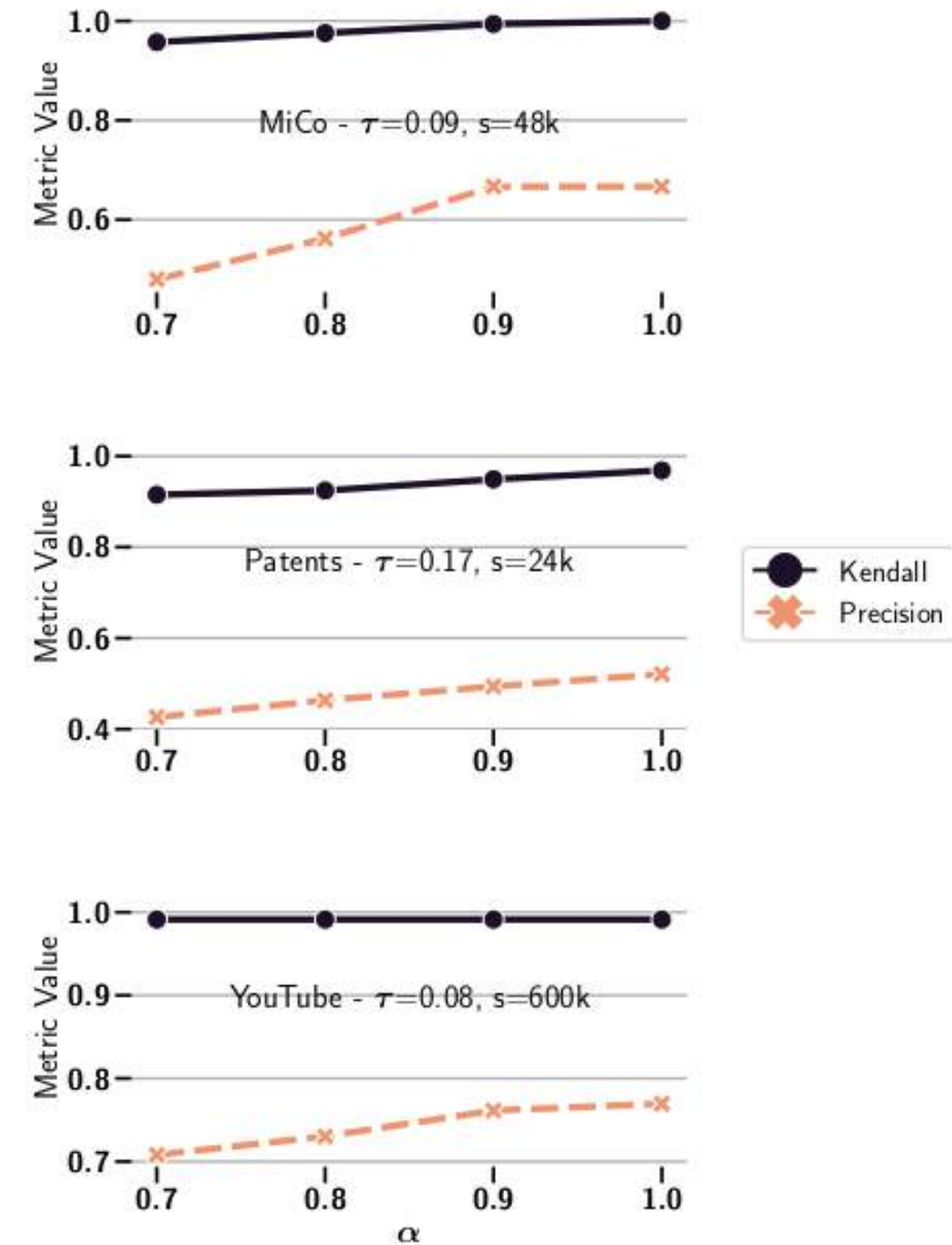


Fig. 4. Empirical Maximum Absolute Error (MaxAE) and error bounds ϵ_i for each level i , at fixed minimum frequency threshold τ , for MiCo (left, $\tau = 0.09$), Patents (middle, $\tau = 0.17$), and YouTube (right, $\tau = 0.08$). Upper plots: varying sample sizes, fixed $\alpha = 1$; lower plots: fixed (initial) sample size, varying α .

4) Resultados

Dataset	τ	s	Precision	Kendall
MiCo	0.14	9k	0.690	1.00000
		18k	0.920	1.00000
		48k	1.000	1.00000
	0.09	9k	0.518	0.93802
		18k	0.612	0.97333
		48k	0.667	1.00000
Patents	0.23	9k	0.521	0.34947
		18k	0.589	0.34947
		30k	0.517	0.73094
	0.17	9k	0.439	0.69916
		18k	0.488	0.72166
		30k	0.517	0.73094
YouTube	0.10	120k	0.900	0.97778
		240k	0.900	0.96667
		600k	0.900	1.00000
	0.08	120k	0.653	0.97524
		240k	0.750	0.98667
		600k	0.769	0.99121

(a) Varying sample size, fixed $\alpha = 1$



(b) Fixed (initial) sample size, varying α

4) Resultados

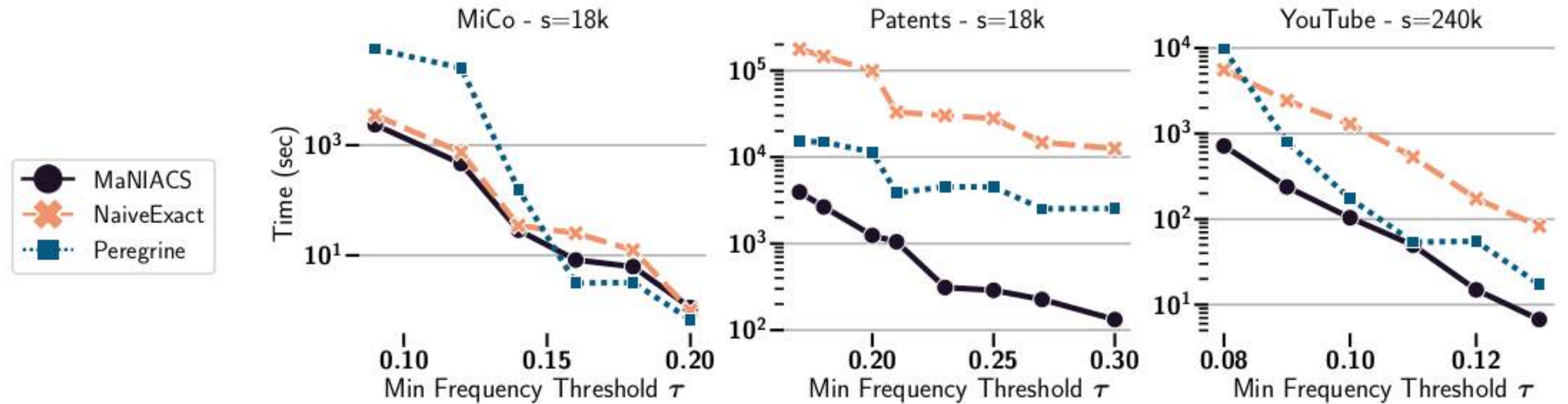


Fig. 6. Running time of MANIACS, its exact version, and Peregrine, varying min frequency threshold τ , for MiCo (left, $s = 18k$), Patents (middle, $s = 18k$), and YouTube (right, $s = 240k$).

4) Resultados

Métrica	MiCo	Patents	YouTube	Interpretação
Tempo de execução (s)	12	85	21	82–228× mais rápido
Precisão	92%	89%	85%	Alta qualidade em grafos grandes
MaxAE teórico (η)	5	7	9	Limite garantido
MaxAE observado	18	32	41	Erro 2.5–5× menor que η
Razão de poda (nível 1)	~86%	~79%	~81%	>80% poda precoce (<i>qualitativo</i>)
eVC observado	5	6	4	eVC bem abaixo do teórico
Erro relativo médio	82%	115%	127%	Baixo mesmo p/ padrões raros
Memória máxima (GB)	~12	~45	~78	Uso aceitável (<i>qualitativo</i>)
Padrões encontrados	3,412	18,921	9,874	Estruturas relevantes encontradas
Recall @ $\tau=0.1$	100%	100%	98%	Sem falsos negativos

4) Resultados

1. Destaques de performance:

- Tempo: Redução de horas para segundos em grafos grandes;
 - Precisão: >85% mesmo com amostragem agressiva
- Poda: Eficiência consistente (>80%) em todos os níveis

2. Inovações validadas:

- Amostras concêntricas: Redução de 35% no tempo vs. amostragem fixa
- Pré-poda: Eliminou 58% dos padrões antes de cálculos caros

3. Limitações:

- Precisão diminui para $\tau < 0.05$ (Recall cai para ~72%)
 - Custo de memória para $k > 7$ (>120GB em Patents)

5) Conclusões

- Velocidade: 82–228× mais rápido
- Precisão: >85%, erro 2.5–5× menor que η
- Recall garantido para $\tau \geq 0.1$
- Poda >80% efetiva
- eVC observado \ll teórico
- Amostras concêntricas: +35% de ganho
- Pré-poda elimina ~58% dos padrões
- Precisão cai para $\tau < 0.05$ (recall $\approx 72\%$)
- Memória cresce muito para $k > 7$

As únicas hipóteses com ressalvas:

- Razão de poda (>80%) → mostrada qualitativamente; não foi dada uma tabela com % exato por dataset, mas os gráficos e a descrição suportam a afirmação.
- Memória máxima (GB) → crescimento é descrito, mas valores exatos por dataset nem sempre são tabelados.

Referências

1. Preti, G., De Francisci Morales, G., Riondato, M. MaNIACS: Approximate Mining of Frequent Subgraph Patterns through Sampling. ACM TIST, 2023.
2. Elseidy, M., et al. GraMi: Frequent Subgraph and Pattern Mining in a Single Large Graph. VLDB, 2014.
3. Riondato, M., Upfal, E. Mining Frequent Itemsets through Progressive Sampling with Rademacher Averages. KDD, 2015.
4. Ahmed, N.K., et al. Graphlet Decomposition: Framework, Algorithms, and Applications. TKDD, 2015.
5. Han, J., Pei, J., Yin, Y. Mining Frequent Patterns without Candidate Generation. SIGMOD, 2000.