Similaridade Estrutural e Graph Matching

Pedro Henrique Pamplona Savarese

Graph similarity scoring and matching

Graph similarity scoring and matching

- Críticas ao algoritmo HITS de Kleinberg
- Forma generalizada de Blondel:

$$\tilde{x}_{ij}(k) = \sum_{r:(r,i)\in E_B, s:(s,j)\in E_A} x_{rs}(k-1) + \sum_{r:(i,r)\in E_B, s:(j,s)\in E_A} x_{rs}(k-1)$$

  - Resultados de números pares e ímpares de iterações convergem para resultados diferentes
  - Resultado final depende do estado inicial

Graph similarity scoring and matching

- Proposta:
  - Algoritmo que não depende da inicialização

- Extensão: similaridade de arestas

$$y_{pq}(k) \leftarrow x_{s(p)s(q)}(k-1) + x_{t(p)t(q)}(k-1)$$

- Onde $s(p)$ é a origem da aresta p e $t(p)$ é o alvo da aresta p.
- x_{ij} é a similaridade entre os vértices i e j

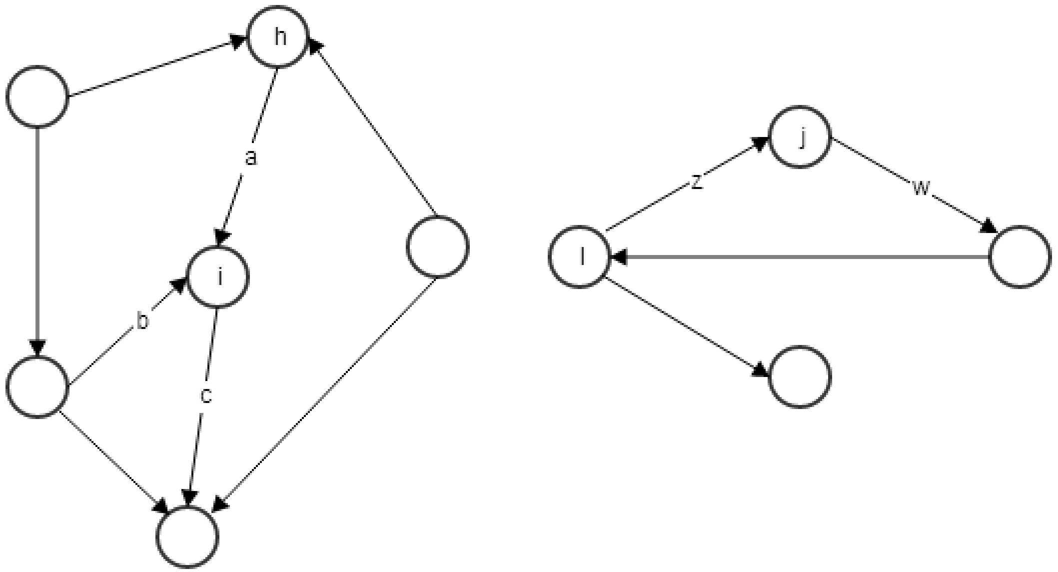
Graph similarity scoring and matching

- Similaridade entre vértices:
  - Passa a depender das similaridades entre arestas

$$x_{ij}(k) \leftarrow \sum_{t(k)=i,t(l)=j} y_{kl}(k-1) + \sum_{s(k)=i,s(l)=j} y_{kl}(k-1)$$

Graph similarity scoring and matching

- Exemplo:



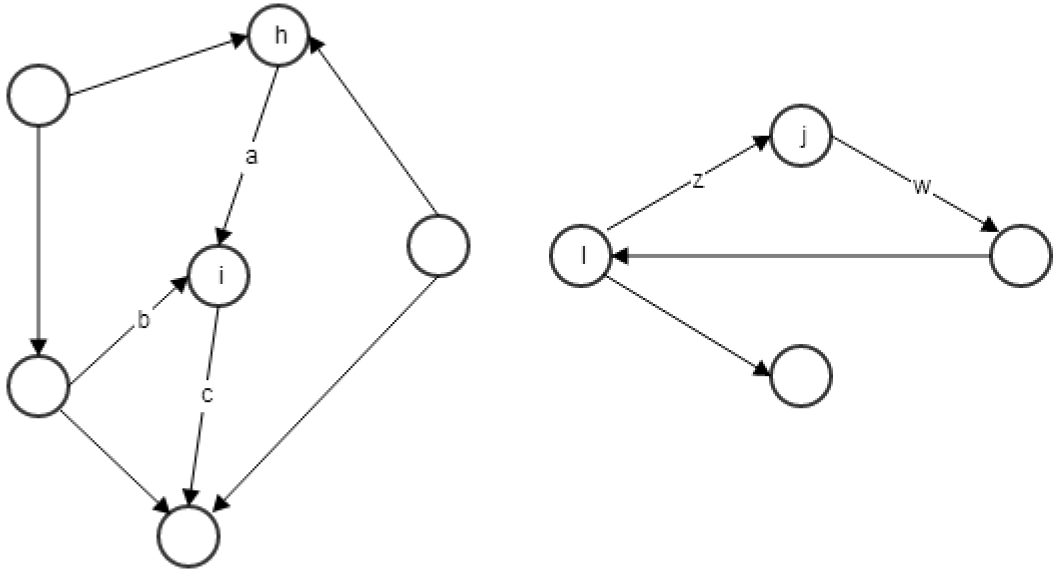
$$x_{ij}(k) = y_{az}(k-1) + y_{bz}(k-1) + y_{cw}(k-1)$$

$$y_{az}(k) = x_{hl}(k-1) + x_{ij}(k-1)$$

Graph similarity scoring and matching

$$X^k = \begin{pmatrix} x_{li} & \cdots & x_{lh} \\ \vdots & \ddots & \vdots \\ x_{ji} & \cdots & x_{jh} \end{pmatrix}$$

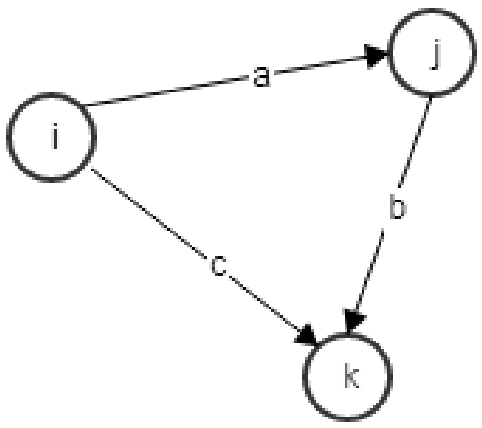
$$Y^k = \begin{pmatrix} y_{za} & \cdots & x_{zc} \\ \vdots & \ddots & \vdots \\ x_{wa} & \cdots & x_{wc} \end{pmatrix}$$



Graph similarity scoring and matching

$$A_s = \begin{array}{c|ccc} & a & b & c \\ \hline i & 1 & 0 & 1 \\ j & 0 & 1 & 0 \\ k & 0 & 0 & 0 \end{array}$$

$$A_T = \begin{array}{c|ccc} & a & b & c \\ \hline i & 0 & 0 & 0 \\ j & 1 & 0 & 0 \\ k & 0 & 1 & 1 \end{array}$$



Graph similarity scoring and matching

- Versão matricial:

$$Y_k \leftarrow B_S^\top X_{k-1} A_S + B_T^\top X_{k-1} A_T, \quad X_k \leftarrow B_S Y_{k-1} A_S^\top + B_T Y_{k-1} A_T^\top$$

Graph similarity scoring and matching

- Operador vec:

$$vec \begin{pmatrix} a & b \\ c & d \end{pmatrix} = (a\ b\ c\ d)^T$$

- Produto de Kronecker:

$$\begin{matrix} a & b \\ c & d \\ e & f \end{matrix} \otimes \begin{matrix} z \\ w \end{matrix} = \begin{matrix} az & aw & cz & cw & ez & ew \\ bz & bw & dz & dw & fz & fw \end{matrix}^{T}$$

Graph similarity scoring and matching

- Versão matricial:

$$Y_k \leftarrow B_S^\top X_{k-1} A_S + B_T^\top X_{k-1} A_T, \quad X_k \leftarrow B_S Y_{k-1} A_S^\top + B_T Y_{k-1} A_T^\top$$

- Versão vetorial:

$$y_k \leftarrow \left(A_S^\top \otimes B_S^\top + A_T^\top \otimes B_T^\top\right) x_{k-1} \equiv G x_{k-1}$$

$$x_k \leftarrow (A_S \otimes B_S + A_T \otimes B_T)\, y_{k-1} \equiv G^\top y_{k-1}$$

Graph similarity scoring and matching

- Assumindo:

$$x_k = G^\top y_{k-1}, \quad y_k = G x_k$$

- Podemos explicitar:

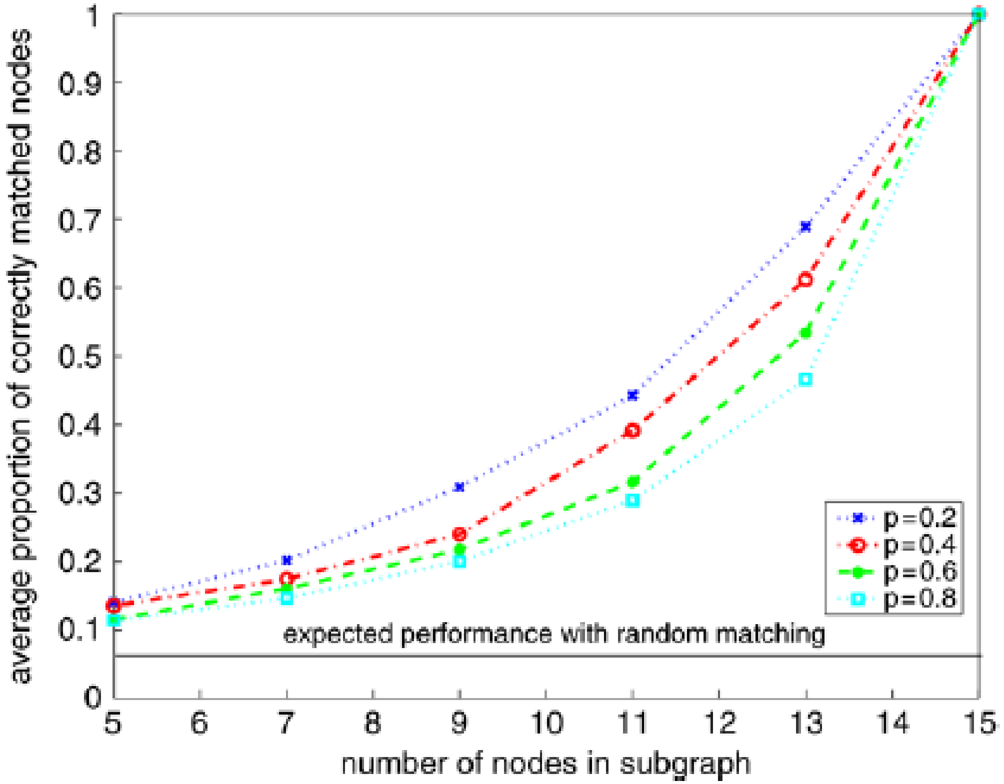
$$\begin{aligned} x_k &\leftarrow (G^\top G) x_{k-2} \\ &= (\mathcal{A} \otimes \mathcal{B} + \mathcal{A}^\top \otimes \mathcal{B}^\top + D_{A_S} \otimes D_{B_S} + D_{A_T} \otimes D_{B_T}) x_{k-2} \end{aligned}$$

- Algoritmo de Blondel:

$$x_k \leftarrow (\mathcal{A} \otimes \mathcal{B} + \mathcal{A}^\top \otimes \mathcal{B}^\top) x_{k-1}$$

Graph similarity scoring and matching

- Graph-Matching (com subgrafos):



Exact and Approximate Graph Matching Using Random Walks

Exact and Approximate Graph Matching Using Random Walks

- Modelo de Random Walk:

$$x_p(t+1) = \sum_{q \in \boldsymbol{G}} x(p\,|q,j)\; x(j|q) x_q(t) + \sum_{q \in pa(p)} x(p\,|q,l)\; x(l|q) x_q(t)$$

Exact and Approximate Graph Matching Using Random Walks

- Matricialmente:

$$\boldsymbol{x}(t+1) = \left(\boldsymbol{\Sigma}\, \boldsymbol{D_j}\right)^T \boldsymbol{x}(t) + (\Delta\, \boldsymbol{D_l})^T \boldsymbol{x}(t)$$

$\boldsymbol{\Sigma}$ é a matriz com probabilidades de destinos de pulos $x(p\,|q,j)$

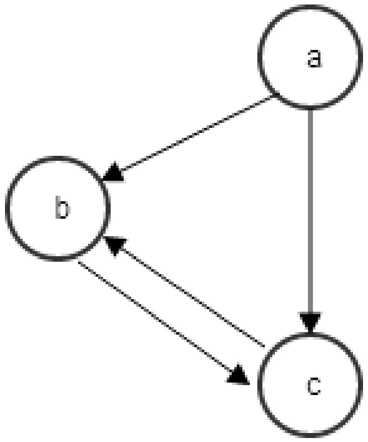
Δ é a matriz com probabilidades de destinos de passos $x(p\,|q,l)$

$\boldsymbol{D_j}$ é a matriz diagonal com probabilidades de realizar pulos $x(j\,|q)$

$\boldsymbol{D_l}$ é a matriz diagonal com probabilidades de realizar passos $x(l\,|q)$

Exact and Approximate Graph Matching Using Random Walks

- Exemplo:



$$\boldsymbol{\Sigma} = \begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{matrix}$$

$$\Delta = \begin{matrix} 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$\boldsymbol{D_j} = \begin{matrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{matrix}$$

$$\boldsymbol{D_l} = \begin{matrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{matrix}$$

Exact and Approximate Graph Matching Using Random Walks

- Simplificações:

  - A probabilidade de realizar um salto independe do nó de origem
  - A probabilidade do destino do salto é uniforme
  - A probabilidade do destino do passo é uniforme

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{J}$$

$$\Delta = \boldsymbol{\theta}\mathbf{A}$$

$$\boldsymbol{D_j} = (1-d)\ \boldsymbol{I}$$

$$\boldsymbol{D_l} = d\ \boldsymbol{I}$$

Exact and Approximate Graph Matching Using Random Walks

- Simplificações:

$$\boldsymbol{x}(t+1) = \left(\frac{1-d}{n}\boldsymbol{J}\right)^T \boldsymbol{x}(t) + (\boldsymbol{\theta}\mathbf{A}\ d\ \boldsymbol{I}\ )^T \boldsymbol{x}(t)$$

$$= \frac{1-d}{n}\bar{I} + d\boldsymbol{W}\ \boldsymbol{x}(t)$$

$$\boldsymbol{x}^* = \left(\frac{1-d}{n}\right)(\boldsymbol{I} - d\boldsymbol{W})^{-\mathbf{1}}\bar{I}$$

Exact and Approximate Graph Matching Using Random Walks

- Provas:
  - Se G_1 e G_2 são isomórficos, então $\boldsymbol{x}_1^* = \boldsymbol{P}\boldsymbol{x}_2^*$, onde $\boldsymbol{A_1} = \boldsymbol{P}\boldsymbol{A_2}\boldsymbol{P^T}$
  - Dado $D = \{d_1, d_2, \dots, d_N\}$ e $\boldsymbol{Z}^*(G, D) = |\boldsymbol{x}^*(d_1)|, \dots, |\boldsymbol{x}^*(d_2)|$, se $Rank(\boldsymbol{Z}^*(G, D)) = N$, o grafo pode ser reconstruído a partir de $\boldsymbol{Z}^*(G, D)$
  - Se $\boldsymbol{Z}^*(G_1, D) = \boldsymbol{P}\,\boldsymbol{Z}^*(G_2, D)$, e $Rank(\boldsymbol{Z}^*(G_1, D)) = N$, então G_1 e G_2 são isomórficos

Exact and Approximate Graph Matching Using Random Walks

- Exemplo:

Para $d = 0.8$:

$$\boldsymbol{x}_1^* = (0.5\ 0.2, 0.2\ 0.1)$$

$$\boldsymbol{x}_2^* = (0.1\ 0.5, 0.2\ 0.2)$$

G_1 e G_2 não são necessariamente isomórficos (é preciso testar todas as permutações)

Exact and Approximate Graph Matching Using Random Walks

- Exemplo:

$$D = \{0.8, 0.6\}$$

$$\boldsymbol{Z}^*(G_1, D) = \begin{matrix} \mathbf{0.5} & \mathbf{0.25} \\ \mathbf{0.2} & \mathbf{0.4} \\ \mathbf{0.2} & \mathbf{0.25} \\ \mathbf{0.1} & \mathbf{0.1} \end{matrix} \qquad \boldsymbol{Z}^*(G_2, D) = \begin{matrix} \mathbf{0.2} & \mathbf{0.25} \\ \mathbf{0.1} & \mathbf{0.1} \\ \mathbf{0.2} & \mathbf{0.25} \\ \mathbf{0.5} & \mathbf{0.25} \end{matrix}$$

Como $\boldsymbol{Z}^*(G_1, D) = \boldsymbol{P}\, \boldsymbol{Z}^*(G_2, D)$, porém $Rank(\boldsymbol{Z}^*(G_1, D)) = 2 < 4$ não podemos concluir que G_1 e G_2 são isomórficos

Exact and Approximate Graph Matching Using Random Walks

- Grafos com Labels:
  - Passos:

$$x(p|q,l) = \frac{f_l(L_p, L_q)}{\sum_{i \in ch(q)} f_l(L_i, L_q)}$$

Onde f_l é uma métrica de similaridade entre dois labels

Exact and Approximate Graph Matching Using Random Walks

- Grafos com Labels:
  - Pulos:

$$x(p|q,j) = \frac{f_j(L_p, L_q)}{\sum_{i \in G} f_j(L_i, L_q)}$$

Onde f_j é outra métrica de similaridade entre dois labels

Exact and Approximate Graph Matching Using Random Walks

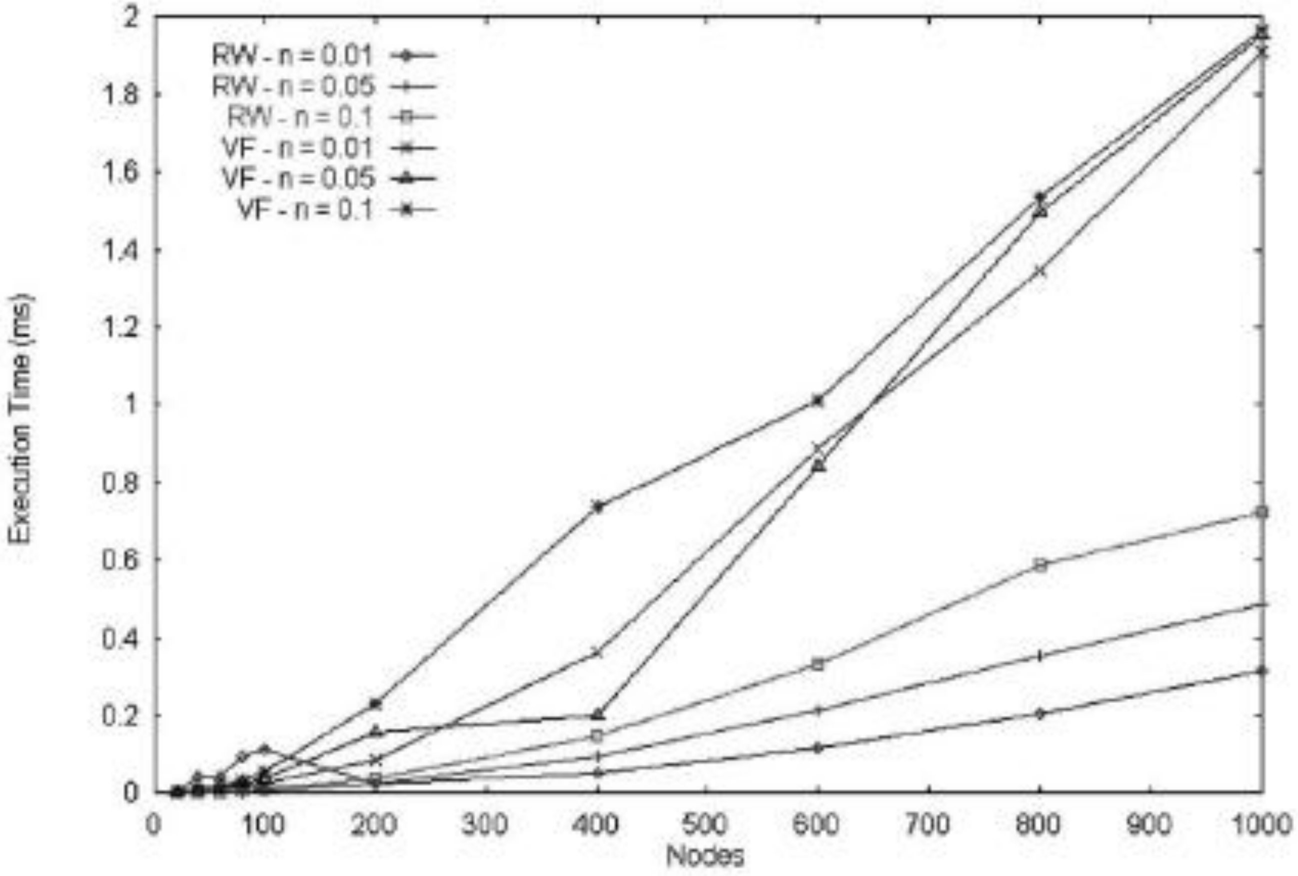
- Matching exato:
  - Calcular $\boldsymbol{Z}^*(G_1, D)$ e $\boldsymbol{Z}^*(G_2, D)$ para conjuntos crescentes de D
  - Preencher partes de $\boldsymbol{P}$ tal que $\boldsymbol{Z}^*(G_1, D) = \mathbf{P}\, \boldsymbol{Z}^*(G_2, D)$
  - Quando $\boldsymbol{P}$ estiver completa (se estiver), e $\boldsymbol{P}^T \boldsymbol{A_1} \boldsymbol{P} = \boldsymbol{A_2}$, G_1 e G_2 são isomórficos
  - Se não estiver, todas as possibilidades de $\boldsymbol{P}$ devem ser testadas

Exact and Approximate Graph Matching Using Random Walks

- Matching aproximado:
  - Cada nó recebe um vetor de informações como label
  - Calcula $\boldsymbol{Z}^*(G_1, D)$ e $\boldsymbol{Z}^*(G_2, D)$
  - Adiciona ao vetor de label de cada nó i seus espectros (i-ésima linha de $\boldsymbol{Z}^*$)
  - Cria um grafo bipartido que representa o problema
  - Minimiza a soma dos pesos com o algoritmo Húngaro

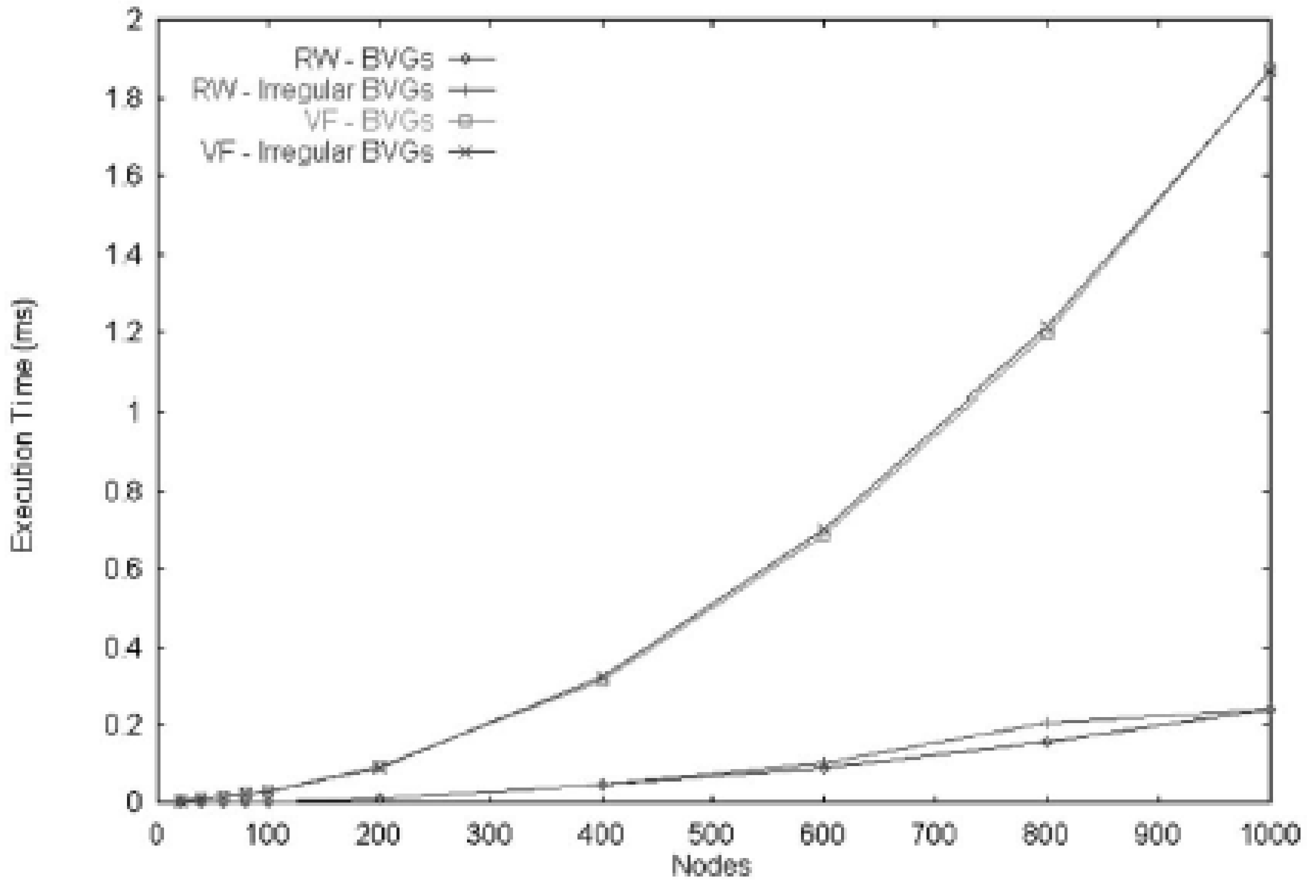
Exact and Approximate Graph Matching Using Random Walks

- Matching exato:
  - G(n,p):



Exact and Approximate Graph Matching Using Random Walks

- Matching exato:
  - BVG/iBVG:



Exact and Approximate Graph Matching Using Random Walks

- Matching aproximado:
  - COIL:

Exact and Approximate Graph Matching Using Random Walks

- Matching aproximado:
  - Segmentação: cada nó representa uma região (≈30 por imagem)
  - Labels: perímetro, área, cor, etc.
  - f_l e f_j dependem da cor e da área, respectivamente
  - Peso das arestas do grafo bipartido são dadas pela distância Euclideana entre os vetores label

Exact and Approximate Graph Matching Using Random Walks

- Matching aproximado:

Number of clusters	Accuracy Rate
10	0.9412
20	0.9735
40	0.952
60	0.9504
80	0.9299

Accuracy Rate Obtained Using Both the Visual and Topological Features or Only the Visual Features

Retrieved Images	Visual and Topological Features	Visual Features
4	0.9735	0.8121
9	0.9544	0.8004
14	0.9337	0.7312