Correcting and Speeding-Up Bounds for Non-Uniform Graph Edit Distance

David B. Blumenthal and Johann Gamper | {david.blumenthal,gamper}@inf.unibz.it

unibz Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Motivation and Results

motivation

- approximation is important as exact computation is NP-hard
- **uniform edit costs**: many algorithms computing lower and upper bounds exist
- **non-uniform edit costs**: Bp [3, 4] is only algorithm that considers node and edge labels and allegedly computes lower and upper bounds

results

- **Bp is incorrect**: in general, it does not yield lower bound
- **Branch**, a corrected version of Bp that runs in $\mathcal{O}(n^5)$ time
- **BranchFast**, a speed-up of Bp that runs in $\mathcal{O}(n^4)$ time
- Branch and BranchFast are **Pareto optimal**: they outperform all competitors in terms of runtime or in terms of accuracy of lower bounds

Graph Edit Distance: Basic Definitions

graph edit distance

- minimum cost $c(P)$ of edit path P between graphs G and H

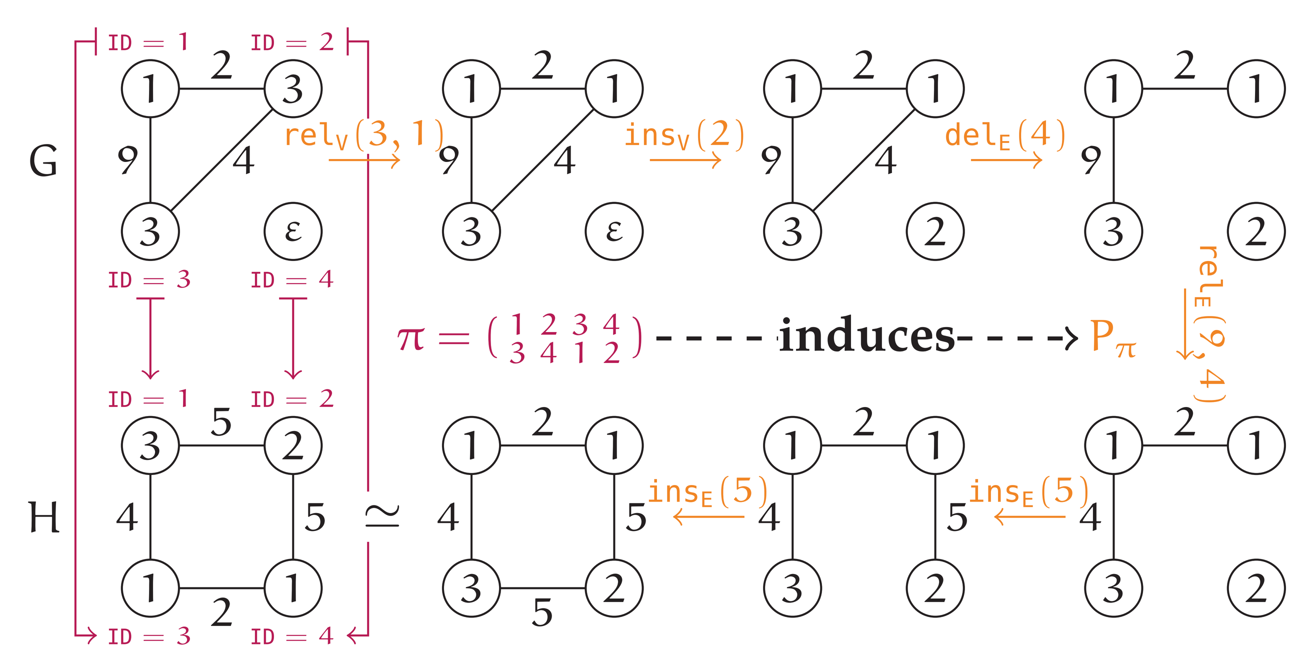
edit path

- sequence $\langle \mathtt{op}_1, \ldots, \mathtt{op}_r \rangle$ of edit operations transforming G into a graph isomorphic to H
- cost of edit path $P = \langle \mathtt{op}_1, \ldots, \mathtt{op}_r \rangle$: $c(P) := \sum_{s=1}^{r} c(\mathtt{op}_s)$

edit operations

- **inserting** isolated α-labelled node or α-labelled edge
- **deleting** isolated α-labelled node or α-labelled edge
- **relabelling**: changing node or edge label from α to $\beta \neq \alpha$
- costs $c(\mathtt{op})$ of edit operations are defined via metrics on label alphabets, e. g., discrete metric, Euclidean distance, string edit distance

Induced Edit Paths, Minimum Linear Assignment, and a Strategy for Computing Bounds



metric edit costs

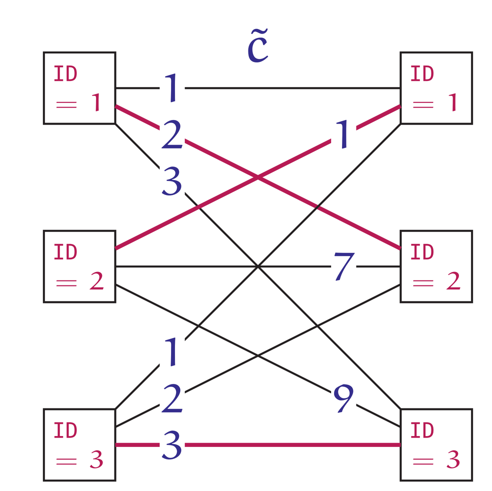
- can assume w. l. o. g. that $|V^G| = |V^H|$
- $\lambda(G,H)$ = minimum cost of edit path induced by permutation $\pi : V^G \to V^H$

strategy for computing bounds

- define edge costs $\tilde{c}$ for **auxiliary bipartite graph** $(V^G \times V^H, \tilde{c})$ s. t. a minimum linear assignment $\pi^\star$ for $(V^G \times V^H, \tilde{c})$ induces cheap edit path $P_{\pi^\star}$
- **upper bound:** is given as cost $c(P_{\pi^\star})$ of $P_{\pi^\star}$
- **lower bound:** can be obtained from $\tilde{c}(\pi^\star)$

minimum linear assignment

(solvable in $\mathcal{O}(n^3)$ time)



$\pi^\star = \left(\begin{smallmatrix}1&2&3\\2&1&3\end{smallmatrix}\right)$ is optimal assignment

The Algorithms Bp, Branch, and BranchFast

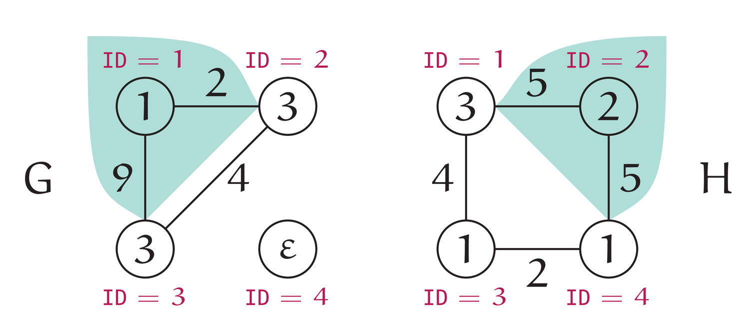
common approach

- decompose graphs into **branches rooted at the nodes**, i. e., nodes with incident edges
- define auxiliary edge cost $\tilde{c}(i,k) := \tilde{c}_V(i,k) + \tilde{c}_E(i,k)$ as **branch transformation costs**
- define cost $\tilde{c}_V(i,k)$ of **adjusting nodes of branches** as cost for changing i's label into k's label

differences

1. how to define cost $\tilde{c}_E(i,k)$ of **adjusting edges of branches**?
2. how to **obtain lower bound** from assignment cost $\tilde{c}(\pi^\star)$ of minimum linear assignment for $(V^G \times V^H, \tilde{c})$?

branches rooted at node 1 in G and at node 2 in H



Euclidean edit costs

- $\tilde{c}_V(1,2) = 1$
- **Bp**: $\tilde{c}_E(1,2) = 7$
- **Branch**: $\tilde{c}_E(1,2) = 7/2$
- **BranchFast**: $\tilde{c}_E(1,2) = 3$

Bp

1. $\tilde{c}_E(i,k)$: min. cost of linear assignment between edge labels of branches rooted at i and k
2. **lower bound**: $\tilde{c}_V(\pi^\star) + \tilde{c}_E(\pi^\star)/2 \rightsquigarrow$ **is incorrect**

Branch

1. $\tilde{c}_E(i,k)$: (min. cost of linear assignment between edge labels of branches rooted at i and k)/2
2. **lower bound**: $\tilde{c}_V(\pi^\star) + \tilde{c}_E(\pi^\star) \rightsquigarrow$ **runs in** $\mathcal{O}(\mathbf{n^5})$

BranchFast

1. $\tilde{c}_E(i,k)$: (min. cost of linear assignment between edge labels of branches rooted at i and k, where distance between different labels is approximated by minimal distance)/2
2. **lower bound**: $\tilde{c}_V(\pi^\star) + \tilde{c}_E(\pi^\star) \rightsquigarrow$ **runs in** $\mathcal{O}(\mathbf{n^4})$

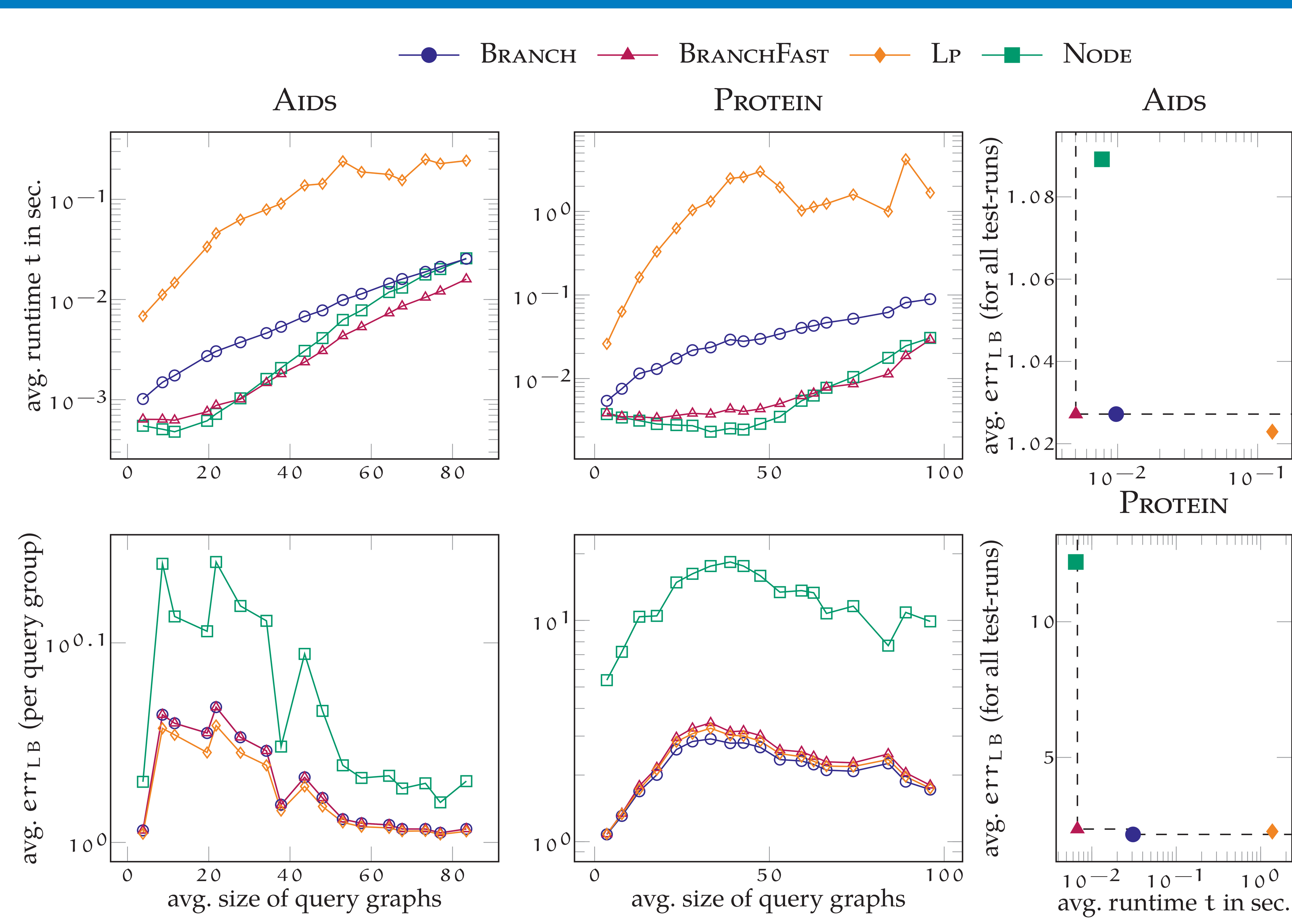
Experiments

competitors

- **Node**: runs in $\mathcal{O}(n^3)$, ignores edges [1]
- **Lp**: runs in $\mathcal{O}(n^7)$, ignores edge labels [1]

experimental setup

- $\mathbf{err_{LB}}(\textbf{Alg}) := $ (tightest LB)/LB(Alg); small values $\rightsquigarrow$ tight lower bounds
- **Aids, Protein**: frequently used, publicly available datasets with naturally induced relabelling costs [2]
- randomly selected 100 model graphs from datasets
- randomly constructed size-constrained query groups containing 5 query graphs H that satisfy $5(i-1) < |V_H| \leqslant 5i$
- ran each algorithm for all pairs of model and query graphs



References

[1] D. Justice and A. Hero, "A Binary Linear Programming Formulation of the Graph Edit Distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1200–1214, 2006.

[2] K. Riesen and H. Bunke, "IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning," in *SSPR'08*, 2008, pp. 287–297.

[3] ——, "Approximate Graph Edit Distance Computation by Means of Bipartite Graph Matching," *Image Vis. Comput.*, vol. 27, no. 7, pp. 950–959, 2009.

[4] K. Riesen, A. Fischer, and H. Bunke, "Computing Upper and Lower Bounds of Graph Edit Distance in Cubic Time," in *ANNPR'14*, 2014, pp. 129–140.

presented at

33rd *IEEE International Conference on Data Engineering*, San Diego, USA, April 19–22, 2017.