Relationships between Graph Edit Distance and Maximal Common Unlabeled Subgraph.

Luc Brun, Benoit Gaüzère, Sébastien Fourey

Relationships between Graph Edit Distance and Maximal Common Unlabeled Subgraph

Luc Brun Benoît Gaüzère Sébastien Fourey

GREYC, UMR6072 CNRS, ENSICAEN, University of Caen
ENSICAEN, 6 bd maréchal Juin 14050 Caen CEDEX, France

Abstract

Graph edit distance measures the distance between two graphs as the number of elementary operations (vertex/edge insertion, deletion, substitution) required to transform the first graph into the second one. Such a distance allows to define a metric between graphs and has many applications in the structural pattern recognition framework. However, the complexity of the computation of this distance is exponential in the size of both graphs to be compared. In this technical report, we focus our attention on applications where families of graphs to be considered have a finite set of structures. We then investigate under which relationships between the costs of the different elementary operations, such a priori knowledge may be used to pre compute most of the optimal edit path between any two graphs.

1 Introduction

An edit path between two graphs is defined as a set of vertex and edge removals, insertions and substitutions which transforms the first graph into the second. If each elementary operation is associated to a cost, the global cost of an edit path is defined as the sum of the costs of its elementary operations.

The edit distance between two graphs is defined as the minimal costs of all edit paths transforming the first graph into the second. Such an edit distance reflects the minimal amount of transformations that we have to apply in order to transform one graph into another one. Such a distance encodes thus a dissimilarity measure between two graphs. Neuhaus and Bunke [2] have shown that if each elementary operation satisfies the criteria of a distance (separability, symmetry, triangular inequality) then the edit distance defines a metric between graphs.

Using the notations of Bunke [1], let us denote by c_{vd}, c_{vi} the costs of vertex deletion and insertion and by c_{vs}, c_{es} the costs of vertex and edge substitutions. Given an edit path P and two input graphs G_1 and G_2, let us denote by $\hat{G}_1$ the sub graph obtained from G_1 by performing all vertex removals contained

in P. Let us further denote by $\hat{G}_2$ the sub graph of G_2 obtained from G_1 by the successive application of all vertex removals and vertex/edge substitutions contained in P. Note that using Bunke [1] formalism, all graphs are considered as complete graphs, and vertex/edge removals are performed by substituting edge's label by null. Conversely, an edge insertion is performed by substituting its null label by a non null one. Bunke [1] has shown that if $c_{vd} + c_{vi} \leq c_{vs}$ and $c_{vd} + c_{vi} \leq c_{es}$, the sub graphs $\hat{G}_1$ and $\hat{G}_2$ both correspond to a maximum common sub graph of G_1 and G_2.

Intuitively, the constraints $c_{vd} + c_{vi} \leq c_{vs}$ and $c_{vd} + c_{vi} \leq c_{es}$ state that a vertex substitution may always be replaced (at lower cost) by the removal of this vertex followed by its insertion with a new label. Based on these constraints the determination of the edit path with minimal costs which encodes the edit distance must avoid as much as possible vertex and edge substitutions hence leading to the two isomorphic sub graphs $\hat{G}_1$ and $\hat{G}_2$ which can be transformed into one another without any substitution.

One may first note, that in many applications, the constraint imposed by Bunke is counter intuitive, since the removal and the insertion of a vertex modifies twice the structure of a graph while its substitution is a basic operation which modifies the label attached to a vertex without altering the structure of the graph. Moreover, beside the fact that the result shown by Bunke [1] nicely connects the graph edit distance problem with the one of maximum common sub graph, this connection as only minor practical consequences since the efficient computation of a maximum common sub graph remains a challenging problem for which only algorithms with a high complexity are available.

In this study we are mainly interested by a family of graph databases such that for each database $D = \{G_1, \ldots, G_n\}$, each graph G_i is defined by a structure graph (V_i, E_i) belonging to a finite set of known structures B and two label functions μ_i and ν_i mapping respectively the vertices and edges of G_i into two sets of vertex and edge labels. In other words, we are working on databases based on a finite set of structure and arbitrary label functions. Within such a framework the connection between graph edit distance and maximum common sub graph is of little help to compute the graph edit distance. Indeed, the maximum common sub graph of two graphs highly depends on labels which may vary freely in our case. In order to exploit the finite number of graph structure encountered in our databases, we have investigated under which conditions on the different costs of elementary operations, the two sub graphs defined by an edit path may correspond to a maximum *unlabeled* sub graph, i.e. a maximum sub graph computed without taking into account vertex and edge labels. When satisfied, such conditions allow us to decompose the computation of the edit distance between two graphs G_1 and G_2 into three steps, two of these steps being pre-computed.

After the introduction of some basic concepts (Section 2), we study connections between graph edit distance and maximum common unlabeled sub graphs in Section 3. As shown by this last section such connections occur only if some ratios between the costs of elementary operations is above some thresholds. In order to decrease the value of such thresholds we restrict our study to edit paths

preserving connectedness in Section 4. We finally present an efficient method to compute the graph edit distance between two graphs whose structure is known in Section 5.

2 Main Definitions

Definition 1 Unlabeled simple graph

An unlabeled undirected simple graph G is defined by the couple $G = (V, E)$ where V is the set of vertices and $E \subset \mathcal{P}_2(V)$ is the set of edges, where $\mathcal{P}_2(V)$ is the set of 2-element subsets of V.

Definition 2 Labeled simple graph

Let $\mathcal{L}$ be a finite alphabet of vertex and edge labels. A labeled simple graph is a tuple $G = (V, E, \mu, \nu)$ where

- *the couple (V, E) defines an unlabeled simple graph,*
- *$\mu : V \to \mathcal{L}$ is a vertex labeling function,*
- *$\nu : E \to \mathcal{L}$ is an edge labeling function.*

The unlabeled graph associated to a given labeled graph $G = (V, E, \mu, \nu)$ is defined by the couple (V, E).

In the following we will only consider simple graphs that we will simply denote by unlabeled (resp. labeled) graphs. The term graph will denote indifferently a labeled or an unlabeled graph.

Definition 3 Sub graph

- *An unlabeled graph $G_1 = (V_1, E_1)$ is said to be an unlabeled sub graph of $G_2 = (V_2, E_2)$ if $V_1 \subset V_2$ and $E_1 \subset E_2 \cap \mathcal{P}_2(V_1)$.*
- *If $G_1 = (V_1, E_1, \mu_1, \nu_1)$ and $G_2 = (V_2, E_2, \mu_2, \nu_2)$ are both labeled graphs then G_1 is a sub graph of G_2 if (V_1, E_1) is an unlabeled sub graph of (V_2, E_2) and if the following additional constraint is fulfilled:*

  $$\mu_{2|V_1} = \mu_1 \text{ and } \nu_{2|E_1} = \nu_1$$

  where $f_|$ denotes the restriction of function f to a particular domain.
- *An unlabeled sub graph of a labeled graph G is an unlabeled sub graph of the unlabeled graph associated to G.*

Definition 4 Graph isomorphism

- *Given two unlabeled graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we say that there is a structural isomorphism between G_1 and G_2 and we denote it by $G_1 \simeq_s G_2$ if there exists a bijective function f from V_1 to V_2 such that:*

$$\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2.$$

With a slight abuse of notation we consider that f also applies from E_1 to E_2 and maps any edge $\{u, v\}$ of E_1 onto $\{f(u), f(v)\}$ in E_2.

- *Given two labeled graphs G_1 and G_2 we say that G_1 is isomorphic to G_2 (denoted as $G_1 \simeq G_2$) if the unlabeled graph associated to G_1 is isomorphic to the one associated to G_2 and the following additional constraint is fulfilled:*

$$\mu_2 \circ f = \mu_1 \text{ and } \nu_2 \circ f = \nu_1$$

Definition 5 Common sub graph

- *A graph G is said to be a common unlabeled sub graph of two graphs G_1 and G_2 if there are two unlabeled sub graphs $\hat{G}_1$ and $\hat{G}_2$ of G_1 and G_2 such that:*

$$G \simeq_s \hat{G}_1 \simeq_s \hat{G}_2$$

- *A graph G is said to be a common sub graph of two graphs G_1 and G_2 if there are two sub graphs $\hat{G}_1$ and $\hat{G}_2$ of G_1 and G_2 such that:*

$$G \simeq \hat{G}_1 \simeq \hat{G}_2$$

Definition 6 Maximal/Maximum common sub graphs

- *A common sub graph of two graphs G_1 and G_2 is maximal if it is a common sub graph of G_1 and G_2 and if it is not a sub graph of any common sub graphs of G_1 and G_2.*

- *A common sub graph G is maximum if it is a maximal sub graph of G_1 and G_2 and if there exists no other common sub graph of G_1 and G_2 with more nodes than G.*

Note that a common graph can not be maximal if it is a sub graph of a common graph with a same number of vertices but with less edges. On the other hand, one common sub graph may be maximum despite the fact that there exist other maximal common sub graphs with more edges but less vertices.

Definition 7 Elementary edit operations *An elementary edit operation is one of the following operation applied on a graph:*

- *Vertex/Edge removal. Such removals are defined as the removal of the considered element from sets V or E.*

- *Vertex/Edge insertion. On labeled graphs, a vertex/edge insertion also associates a label to the inserted element.*

- *Vertex/Edge substitution if the graph is a labeled one. Such an operation modifies the label of a vertex or an edge and thus transforms the vertex or edge labeling functions.*

Definition 8 Cost of an elementary edit operation *Each elementary operation is associated to a cost encoded by a specific function for each type of operation. More precisely, let x denote an elementary operation, we distinguish the following cost functions:*

- *Vertex ($c_{vd}(x)$) and edge removal ($c_{ed}(x)$)*
- *Vertex ($c_{vi}(x)$) and edge ($c_{ei}(x)$) insertion,*
- *Vertex ($c_{vs}(x)$) and edge ($c_{es}(x)$) substitution on labeled graphs.*

By extension, we will consider that functions c_{vd} and c_{vi} (resp. c_{ed} and c_{ei}) apply on the set of vertices (resp. set of edges) of a graph. Hence, the cost $c_{vd}(v)$ denotes the cost of the elementary operation "removing vertex v".

We assume that a substitution transforming one label into the same label has zero cost:

$$\forall l \in \mathcal{L},\ c_{vs}(l \to l) = c_{es}(l \to l) = 0$$

where $l \to l'$ denotes the substitution of label l into l' on some edge or vertex.

Definition 9 Edit path

- *An edit path of a graph G is a sequence of elementary operations applied on G, where vertex removal and edge insertion have to satisfy the following constraints:*
  1. *A vertex removal implies a first removal of all its incident edges,*
  2. *An edge insertion can be applied only between two existing or already inserted vertices.*
- *An edit path between two graphs G_1 and G_2 is an edit path of G_1 whose last graph is G_2.*

If G_1 and G_2 are unlabeled we assume that no vertex nor edge substitutions are performed.

Definition 10 Cost of an edit path

The cost of an edit path P, denoted $\gamma(P)$ is the sum of the costs of its elementary edit operations.

Definition 11 Edit distance

The edit distance between two graphs G_1 and G_2 is defined as the minimal cost of all edit paths between G_1 and G_2.

$$d(G_1, G_2) = \min_{P \in \mathcal{P}(G_1, G_2)} \gamma(P)$$

where $\mathcal{P}(G_1, G_2)$ is the set of all edit paths transforming G_1 into G_2. An edit path from G_1 to G_2 with minimal cost is called an optimal path.

Proposition 1 *Given any graph G, and any edit path P of G, the transformation of G by P is still a simple graph.*

Proof:

Let $G = (V, E, \mu, \nu)$ and $G' = (V', E', \mu', \nu')$ denote the initial graph and its transformation by P. Since the insertion of vertices and edges induces the definition of their labels, function μ' (resp. ν') defines a valid labeling function over V' (resp. E'). Let us consider $\{u, v\} \in E'$.

- If $\{u, v\} \in E \cap E'$ then u and v belong to $V \cap V'$, otherwise, the removal of either u or v would have implied the removal of $\{u, v\}$ (Definition 9, condition 1).
- If $\{u, v\} \in E' \setminus E$ then $\{u, v\}$ has been inserted. In this case, both u and v should either be present in V or have been inserted before the insertion of edge $\{u, v\}$ (Definition 9, condition 2). Moreover, none of these vertices can be removed after the last insertion of edge $\{u, v\}$ since such a removal would imply the removal of $\{u, v\}$ (Definition 9, condition 1). Both u and v thus belong to V'.

In both cases, vertices u and v belong to V', which shows that $E' \subset \mathcal{P}_2(V')$. It follows that $G' = (V', E', \mu', \nu')$ is a labeled simple graph according to Definition 2. □

Definition 12 Independent edit path *An independent edit path between two labeled graphs G_1 and G_2 is an edit path such that:*

1. *No vertex nor edge is both substituted and removed,*
2. *No vertex nor edge is simultaneously substituted and inserted,*
3. *Any inserted element is never removed,*
4. *Any vertex or edge is substituted at most once,*

Note that an independent edit path is not minimal in the number of operations. Indeed, definition 12 still allows to replace one substitution by one removal followed by one insertion (but such an operation can be performed only once for each vertex or edge thanks to condition 3). We however forbid useless operations such as the substitution of one vertex followed by its removal (condition 1) or the insertion of a vertex with a wrong label followed by its substitution (condition 2). In the following we will only consider independent edit paths that we simply call edit paths.

Proposition 2 *The elementary operations of any edit path between two graphs G_1 and G_2 may be ordered into a sequence of removals, followed by a sequence of substitutions and terminated by a sequence of insertions.*

Proof:

Let R, S and I denote the sub-sequence of Removals, Substitutions and Insertions of an edit path P, respectively. Since no removal may be performed on a substituted element (condition 1) and no removal may be performed on an inserted element (condition 3), removals only apply on elements which are neither substituted nor inserted. Such removals operations may thus be grouped at the beginning of the edit path. Now, since an element can not be substituted after its insertion, substitutions apply only on the remaining elements after the removal step and can be grouped after the removal operations. The remaining operations only contain insertions.

Let us consider the sequence of elementary operations (R, S, I) the order within sequences R, S and I being deduced from the one of P. Such a sequence may be defined since operations in R apply on elements not in S and I while operations in S do not apply on the same elements than operations in I. Such sets are independent, hence the definition of the independent edit path. We, however, still have to show that the sequence (R, S, I) defines a valid edit path.

1. Since R contains all the removal operations contained in P, if P satisfies condition 1 of Definition 9, so does the sequence R.

2. Let us suppose that an edge insertion is valid in sequence P while it violates Definition 9, condition 2 in sequence (R, S, I). Let us denote by $\{u, v\}$ such an edge. Edge $\{u, v\}$ violates condition 2 in sequence (R, S, I) only if either the removal of u or v belongs to R. In such a case the insertion of $\{u, v\}$ in P should be made before the removal of u or v. But such a removal would imply the removal of all the incident edges of u or v (Definition 9, condition 1) including the newly inserted edge $\{u, v\}$. Such an operation would violate the independence of P (Definition 12, condition 3).

The sequence (R, S, I) is thus a valid edit path which transforms a graph G_1 into G_2 if P do so. □

Proposition 3 *Let P be an edit path between two graphs G_1 and G_2. Let us further denote by R, S and I the sequence of vertex and edge Removals, Substitutions and Insertions performed by P, the order in each sequence being deduced from the one of P. Then:*

- *the graph $\hat{G}_1$ obtained from G_1 by applying removal operations R is a sub graph of G_1,*
- *the graph $\hat{G}_2$ obtained from G_1 by applying the sequence of operations (R, S) is a sub graph of G_2,*
- *Both $\hat{G}_1$ and $\hat{G}_2$ correspond to a same common unlabeled sub graphs of G_1 and G_2.*

Proof:

1. Since the sequence R is an edit path, $\hat{G}_1$ is a graph by Proposition 1. Moreover, since R is only composed of removal operations, we trivially have $\hat{V}_1 \subset V_1$ and $\hat{E}_1 \subset E_1$. The fact that $\hat{E}_1 \subset E_1 \cap \mathcal{P}_2(\hat{V}_1)$ is induced by the fact that $\hat{G}_1$ is a graph. Moreover, if G_1 is a labeled graph, since removal operations do not modify labels, labels on $\hat{G}_1$ are only the restriction of the ones on G_1 to $\hat{V}_1$ and $\hat{E}_1$.

2. The graph $\hat{G}_2$ is deduced from G_1 by the edit path (R, S), it is thus a graph. Moreover, G_2 is deduced from $\hat{G}_2$ by the sequence of insertions I. We thus trivially have: $\hat{V}_2 \subset V_2$ and $\hat{E}_2 \subset E_2 \cap \mathcal{P}_2(\hat{V}_2)$. Moreover, since insertion operations do not modify the label of existing elements, the restriction of the label functions of G_2 to $\hat{V}_2$ and $\hat{E}_2$ corresponds to the label functions of $\hat{G}_2$.

3. Sub graph $\hat{G}_2$ is deduced from $\hat{G}_1$ by the sequence of substitution operations S. Since substitution operations only modify label functions, the structure of both graphs is the same and there exists a structural isomorphism between both graphs.

□

One should note that it may exist several structural isomorphisms between $\hat{G}_1$ and $\hat{G}_2$. The set of substitutions S fixes one of them, say f such that the image of any element of $\hat{G}_1$ by f have the same label than the one defined by the substitution. More precisely, let us suppose that we enlarge the set of substitution S by 0 cost substitutions so that all the vertices and edges of $\hat{G}_1 = (\hat{V}_1, \hat{E}_1, \mu_1, \nu_1)$ are substituted. In this case, we have:

$$\begin{cases} \forall v \in \hat{V}_1, & \mu_2(f(v)) = l_v \\ \forall e \in \hat{E}_1, & \nu_2(f(e)) = l_e \end{cases}$$

Where l_v and l_e correspond to the labels defined by the substitutions of v and e and μ_2 and ν_2 define respectively the vertex and edge labeling functions of G_2.

Corollary 1 *Using the same notations than in Proposition 3, the cost $\gamma(P)$ of an edit path is defined by:*

$$\begin{aligned} \gamma(P) = & \sum_{v \in V_1 \setminus \hat{V}_1} c_{vd}(v) + \sum_{e \in E_1 \setminus \hat{E}_1} c_{ed}(e) + \sum_{v \in \hat{V}_1} c_{vs}(v) + \sum_{e \in \hat{E}_1} c_{es}(e) \\ & + \sum_{v \in V_2 \setminus \hat{V}_2} c_{vi}(v) + \sum_{e \in E_2 \setminus \hat{E}_2} c_{ei}(e) \end{aligned}$$

Proof:

The path P and its rewriting in (R, S, P) have the same set of operations and thus a same cost.

From G_1 to $\hat{G}_1$: Operations in R remove vertices in $V_1 \setminus \hat{V}_1$ and edges in $E_1 \setminus \hat{E}_1$.

From $\hat{G}_1$ to $\hat{G}_2$: Substitutions of S apply between the two graphs $\hat{G}_1$ and $\hat{G}_2$. Let us consider the set of substitutions S' which corresponds to the completion of S by 0 cost substitutions so that all vertices and edges of $\hat{G}_1$ are substituted. Both S and S' have a same cost. The cost of S' is defined as the sum of costs of the substituted vertices and edges of $\hat{G}_1$.

From $\hat{G}_2$ to G_2: Operations in I insert vertices of $V_2 \setminus \hat{V}_2$ and edges of $E_2 \setminus \hat{E}_2$ in order to obtain G_2 from $\hat{G}_2$.

□

Corollary 2 *If all costs do not depend on the vertex/edge involved the cost of an edit path P is equal to:*

$$\begin{aligned}\gamma(P) &= (|V_1| - |\hat{V}_1|)c_{vd} + (|E_1| - |\hat{E}_1|)c_{ed} + V_f c_{vs} + E_f c_{es} \\ &\quad + (|V_2| - |\hat{V}_2|)c_{vi} + (|E_2| - |\hat{E}_2|)c_{ei}\end{aligned}$$

where V_f (resp. E_f) denotes the number of vertices (resp. edges) substituted with a non zero cost and c_{vd}, c_{ed}, c_{vs}, c_{es}, c_{vi}, and c_{ei} denote the constant costs of the associated functions.

Moreover, in this case minimizing the cost of the edit path is equivalent to maximizing the following formula:

$$M(P) \stackrel{not.}{=} |\hat{V}_1|(c_{vd} + c_{vi}) + |\hat{E}_1|(c_{ed} + c_{ei}) - V_f c_{vs} - E_f c_{es}$$

Proof:

We deduce immediately from Corollary 2 the following formula:

$$\begin{aligned}\gamma(P) &= (|V_1| - |\hat{V}_1|)c_{vd} + (|E_1| - |\hat{E}_1|)c_{ed} + V_f c_{vs} + E_f c_{es} \\ &\quad + (|V_2| - |\hat{V}_2|)c_{vi} + (|E_2| - |\hat{E}_2|)c_{ei}\end{aligned}$$

We obtain by grouping constant terms:

$$\begin{aligned}\gamma(P) &= |V_1|c_{vd} + |E_1|c_{ed} + |V_2|c_{vi} + |E_2|c_{ei} \\ &\quad - \Big[|\hat{V}_1|c_{vd} + |\hat{E}_1|c_{ed} + |\hat{V}_2|c_{vi} + |\hat{E}_2|c_{ei} - V_f c_{vs} - E_f c_{es}\Big]\end{aligned}$$

Since there is a structural isomorphism between $\hat{G}_1$ and $\hat{G}_2$, we have $\hat{V}_1 = \hat{V}_2$ and $\hat{E}_1 = \hat{E}_2$. So:

$$\begin{aligned}\gamma(P) &= |V_1|c_{vd} + |E_1|c_{ed} + |V_2|c_{vi} + |E_2|c_{ei} \\ &\quad - \Big[|\hat{V}_1|(c_{vd} + c_{vi}) + |\hat{E}_1|(c_{ed} + c_{ei}) - V_f c_{vs} - E_f c_{es}\Big]\end{aligned}$$

The first part of the above equation being constant, the minimization of $\gamma(P)$ is equivalent to the maximization of the last part of the equation. □

3 Edit Distance and Maximal/Maximum Common Unlabeled Sub Graphs

Proposition 4 *Given two graphs G_1 and G_2, and a common unlabeled sub graph G of G_1 and G_2, there exists at least one edit path transforming G_1 into G_2 whose:*

- *sequence of removals produces a sub graph of G_1 structurally isomorphic to G,*
- *sequence of removals and substitutions produces a sub graph of G_2 structurally isomorphic to G,*

Such an edit path is called an edit path between G_1 and G_2 associated to G.

Proof:

Let us consider two sub graphs G'_1 and G'_2 of G_1 and G_2 structurally isomorphic to G. Such sub graphs exist by definition of a common unlabeled sub graph (Definition 5). We build an edit path $P = (R, S, I)$ from G_1 to G_2 passing through G'_1 and G'_2 as follows:

- The set of operations R removes all vertices in $V_1 \setminus V'_1$ and all edges in $E_1 \setminus E'_1$. Such a sequence of operations produces the sub graph G'_1.
- Let f be an isomorphism between G'_1 and G'_2. The set of substitutions is defined as the set of substitutions of each vertex's label $\mu_1(v)$ into $\mu_2(f(v))$ and each edge's label $\nu_1(e)$ into $\nu_2(f(e))$. These substitutions transform G'_1 into G'_2.
- The set of insertions I is defined as the insertion of vertices in $V_2 \setminus V'_2$ and edges in $E_2 \setminus E'_2$. It produces the graph G_2 from G'_2.

□

Proposition 5 *Given two graphs G_1 and G_2, let us denote by δ_v the number of vertices of their maximum common sub graphs and by δ_e the maximal number of edges of their common sub graphs. Then:*

- *If $c_{vs} = 0$ or*
$$\frac{c_{vd} + c_{vi}}{c_{vs}} \geq \delta_v$$
and
- *If $c_{es} = 0$ or*
$$\frac{c_{ed} + c_{ei}}{c_{es}} \geq \delta_e$$
then for any optimal edit path $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, the sub graphs of G_1 and G_2, $\hat{G}_1$ and $\hat{G}_2$ defined respectively by the edit operations R_{opt} and (R_{opt}, S_{opt}) are maximal common unlabeled sub graphs of G_1 and G_2.

Proof:

Let us suppose that there exists one common sub graph G_1' of G_1 and G_2 such that $\hat{G}_1$ is a sub graph of G_1'.

Let us consider the edit path $P = (R, S, I)$ associated to G_1' (Proposition 4). Following Corollary 2, we may compare the costs of the edit paths P and P_{opt} by comparing the two values:

$$\begin{cases} M(P) & = & |V_1'|(c_{vd} + c_{vi}) + |E_1'|(c_{ed} + c_{ei}) - (V_f' c_{vs} + E_f' c_{es}) \\ M(P_{opt}) & = & |\hat{V}_1|(c_{vd} + c_{vi}) + |\hat{E}_1|(c_{ed} + c_{ei}) - (V_f c_{vs} + E_f c_{es}) \end{cases}$$

where V_f and V_f' (resp. E_f and E_f') denote the number of vertices (resp. edges) substitutions performed respectively by P_{opt} and P.

Since $\hat{G}_1$ is a sub graph of G_1' we have $|\hat{V}_1| \leq |V_1'|$ and $|\hat{E}_1| \leq |E_1'|$. Hence if $c_{vs} = c_{es} = 0$ we trivially have $M(P) \geq M(P_{opt})$ which contradicts the optimality of P_{opt}.

Let us suppose that both c_{vs} and c_{es} are strictly positive. We have then:

$$\begin{aligned} M(P) - M(P_{opt}) = \quad & c_{vs}\left[(|V_1'| - |\hat{V}_1|)\tfrac{c_{vd}+c_{vi}}{c_{vs}} - (V_f' - V_f)\right] + \\ & c_{es}\left[(|E_1'| - |\hat{E}_1|)\tfrac{c_{ed}+c_{ei}}{c_{es}} - (E_f' - E_f)\right] \end{aligned}$$

Since $\hat{G}_1$ is a sub graph of G_1, both $|V_1'| - |\hat{V}_1|$ and $|E_1'| - |\hat{E}_1|$ correspond to integer quantities greater or equal to 1. Thus:

$$\begin{aligned} M(P) - M(P_{opt}) & \geq c_{vs}\left[\tfrac{c_{vd}+c_{vi}}{c_{vs}} - (V_f' - V_f)\right] + c_{es}\left[\tfrac{c_{ed}+c_{ei}}{c_{es}} - (E_f' - E_f)\right] \\ & \geq c_{vs}\left[\tfrac{c_{vd}+c_{vi}}{c_{vs}} - V_f'\right] + c_{es}\left[\tfrac{c_{ed}+c_{ei}}{c_{es}} - E_f'\right] \end{aligned} \quad (1)$$

V_f' and E_f' are respectively bounded by $|V_1'| \leq \delta_v$ and $|E_1'| \leq \delta_e$. Using our hypothesis on the ratios $\frac{c_{vd}+c_{vi}}{c_{vs}}$ and $\frac{c_{ed}+c_{ei}}{c_{es}}$ we obtain $M(P_{opt}) - M(P) \geq 0$ which still contradicts our hypothesis on the optimality of P.

If only one of the variables c_{vs} and c_{es} is equal to 0, its associated term on the right side of equation 2 vanishes and the same argument as above holds for the remaining term. □

Proposition 9 provides very poor results since the bounds on both ratios may be very large. We can however not expect to do much more, since the enlargement of a common sub graph may drastically change the implied substitutions by an edit path.

This point is illustrated in Fig. 1. Let us suppose that the first row of this figure represents common sub graphs $\hat{G}_1$ and $\hat{G}_2$ and let us suppose that both right branches of these graphs have a same length. Then $\hat{G}_1$ may be transformed into $\hat{G}_2$ with 0 substitution cost, just by switching the two long branches of both graphs. However, if we suppose that the bottom graphs in Fig. 1 corresponds

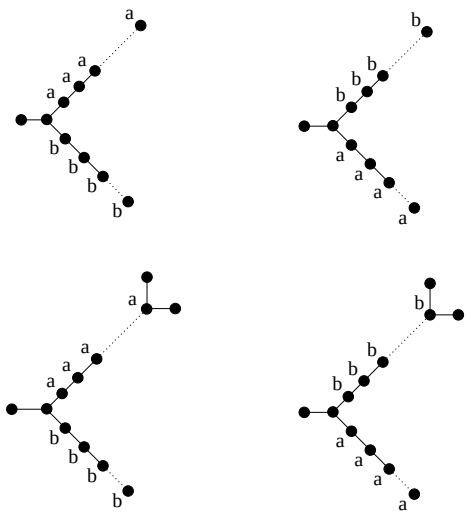

Figure 1: Drastic change of the number of substitutions implied by a slight enlargement of a common sub graph.

to G'_1 and G'_2, then the fact of enlarging the common sub graph by two vertices, forces to match the branch made of vertices labeled a with the one made of vertices labeled b. The number of substitutions is in this case equal to twice the length of the branches.

Proposition 6 *Let us suppose that $c_{ed} = c_{ei} = 0$ and $c_{es} \leq c_{vs}$. Given two graphs G_1 and G_2, let us further denote by δ_v the number of vertices of their maximum common unlabeled sub graphs and by δ_e the maximal number of edges of all maximum common sub graphs. Then if:*

$$\frac{c_{vd} + c_{vi}}{c_{vs}} \geq \delta_v + \delta_e$$

any edit path $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, associated to a maximum common sub graph of G_1 and G_2, induces a lower edit distance than any edit path associated to a non maximal sub graph.

Proof:

Let us denote by $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$ an edit path passing through a maximum common sub graph of G_1 and G_2.

If $c_{ed} = c_{ei} = 0$ the function to maximize by the edit distance (Corollary 2) associated to the edit path P_{opt} reduces to:

$$M(P_{opt}) = |\hat{V}_1|(c_{vd} + c_{vi}) - V_f c_{vs} - E_f c_{es}$$

where V_f and E_f denote the number of vertex and edge substitutions encoded by S_{opt}.

Let us consider an alternative edit path $P = (R, S, I)$ associated to the non maximum sub graphs G'_1 and G'_2 of G_1 and G_2. We have:

$$M(P) = |\hat{V}'_1|(c_{vd} + c_{vi}) - V'_f c_{vs} - E'_f c_{es}$$

where V'_f and E'_f denote the number of vertex and edge substitutions encoded by S.

Since $\hat{G}_1$ is a maximum common sub graph of G_1 and G_2 and not G'_1 we have $\delta = |\hat{V}_1| \geq |V'_1| + 1$. Therefore:

$$\frac{c_{vd} + c_{ni}}{c_{vs}} \geq \delta_v + \delta_e \geq |\hat{V}_1| + |\hat{E}_1| \geq \frac{|\hat{V}_1| + |\hat{E}_1|}{|\hat{V}_1| - |V'_1|}$$

We have thus:

$$\begin{aligned}(|\hat{V}_1| - |V'_1|)(c_{nd} + c_{vi}) &\geq c_{ns}\left[|\hat{V}_1| + |\hat{E}_1|\right] \\ &\geq c_{ns}\left[V_f + E_f\right] \\ &\geq c_{ns}V_f + c_{es}E_f\end{aligned}$$

Enventually,

$$M(P_{opt}) = |\hat{V}_1|(c_{vd} + c_{vi}) - c_{vs}V_f - c_{es}Ef \geq |V'_1|(c_{vd} + c_{vi}) \geq M(P)$$

□

Corollary 3 *Using the same hypothesis than in Proposition 6, an optimal edit path between two graphs G_1 and G_2 is an edit path whose associated common sub graph is an unlabeled maximum common sub graph and which minimizes the number of substitutions: $V_f c_{vs} + E_f c_{es}$.*

The main advantage of Corollary 3 is that if the structure of two graphs is known, their maximum common sub graphs may be pre computed and then the computation of the edit distance reduces to a computation of a minimal number of substitutions between two known structures.

An equivalent result may be obtain without hypothesis on the number of edges of maximum common sub graphs at the price of a higher bound:

Corollary 4 *Using the same notations and hypothesis concerning c_{ed}, c_{ei}, c_{es} and c_{vs} than in Proposition 6, if*

$$\frac{c_{vd} + c_{vi}}{c_{vs}} \geq \frac{\delta_v(\delta_v + 1)}{2}$$

then any edit path $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, associated to a maximum common sub graph of G_1 and G_2, induces a lower edit distance than any edit path associated to a non maximal sub graph.

Proof:

$$\frac{c_{vd}+c_{vi}}{c_{vs}} \geq \frac{\delta_v(\delta_v+1)}{2} = \delta_v + \frac{\delta_v(\delta_v-1)}{2} \geq \delta_v + \delta_e$$

□

A smaller bound may also be obtained by restricting the class of graphs, as stated by the following corollaries.

Corollary 5 *Using the same notations and hypothesis concerning c_{ed}, c_{ei}, c_{es} and c_{vs} than in proposition 6, if G_1 or G_2 is planar and if*

$$\frac{c_{vd}+c_{vi}}{c_{vs}} \geq 4\delta_v - 6$$

then any edit path $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, associated to a maximum common sub graph of G_1 and G_2, induces a lower edit distance than any edit path associated to a non maximal sub graph.

Proof:

If G_1 or G_2 is planar, the common sub graphs of G_1 and G_2 are also planar. Let us denote by G one such common sub graph and by G' the graph obtained from G by removing all its bridges. The planar graph G' is 2-connected. It is well known that the number of edges and vertices within 2-connected planar graphs are related by the following equation: $|E'| \geq \frac{3}{2}|F'|$, where F' denotes the set of faces of G' including the outer face.

Since removals of bridges do not modify the set of faces, the set F of faces of G is equal to F', the one of G'. Moreover, since G' is obtained from G by edge removals we have:

$$|E| \geq |E'| \geq \frac{3}{2}|F'| = \frac{3}{2}|F|.$$

Finally, the number of vertices, edges and faces of a planar graph G are connected by the Euler characteristic:

$$\#cc + 1 = |V| - |E| + |F|$$

where $\#cc$ corresponds to the number of connected components of the graph. Combining Euler characteristic with previous inequalities we obtain:

$$2 \leq \#cc + 1 = |V| - |E| + |F| \leq |V| - |E| + \frac{2}{3}|E| \leq |V| - \frac{1}{3}|E|$$

We have thus:

$$|E| \leq 3|V| - 6$$

This inequality being true for any common sub graph we have:

$$\delta_e \leq 3\delta_v - 6$$

Therefore:

$$\frac{c_{vd}+c_{vi}}{c_{vs}} \geq 4\delta_v - 6 \geq \delta_v + 3\delta_v - 6 \geq \delta_v + \delta_e$$

and the result follows using Proposition 6. □

Corollary 6 *Using the same notations and hypothesis concerning c_{ed}, c_{ei}, c_{es} and c_{vs} than in proposition 6, if G_1 or G_2 is a tree and if*

$$\frac{c_{vd} + c_{vi}}{c_{vs}} \geq 2\delta_v - 1$$

then any edit path $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, associated to a maximum common sub graph of G_1 and G_2, induces a lower edit distance than any edit path associated to a non maximal sub graph.

Proof:

If G_1 or G_2 is a tree, the common sub graph of G_1 and G_2 is a forest. Let $G = (V, E, \mu, \nu)$ denote such a common sub graph of G_1 and G_2. The Euler characteristic of the forest G satisfies:

$$\#cc + 1 = |V| - |E| + |F| = |V| - |E| + 1$$

where $\#cc$ denotes the number of connected components of G and F its set of faces, reduced in this case to the outer one. We have thus :

$$\#cc = |V| - |E| \geq 1 \Rightarrow |E| \leq |V| - 1$$

If G denote now a maximum common sub graph, we obtain $|E| \leq \delta_v - 1$ and by taking the maximum over all maximum common sub graphs we obtain: $\delta_e \leq \delta_v - 1$. We have thus:

$$\frac{c_{vd} + c_{vi}}{c_{vs}} \geq 2\delta_v - 1 = \delta_v + \delta_v - 1 \geq \delta_v + \delta_e$$

and the result follows using Proposition 6. □

4 Edit Path Preserving Connectedness

Definition 13 Elementary edit operations preserving connectedness *An elementary edit operation preserving connectedness is one of the following operation applied on a graph:*

- *Edge removal: Such removals are defined as the removal of the considered element from sets V or E. An edge removal can not increase the number of connected components of a graph.*
- *Edge insertion: An edge insertion in a graph $G = (V, E)$ is restricted to the insertion of an element of $\mathcal{P}_2(V)$. Such an insertion can not decrease the number of connected components of the graph. On labeled graphs, an edge insertion also associates a label to the inserted element.*
- *Vertex-Edge insertion: A vertex-edge insertion on a graph $G = (V, E)$ corresponds to the addition of a vertex $v \notin V$ to v and an edge $\{v, u\}$ with $u \in V$. On labeled graphs, a vertex-edge insertion also associates a label to the inserted elements.*

- *Vertex Removal: The removal of a vertex v in a graph $G = (V, E)$ is defined as the removal of v from V and the removal of any edge $\{v, u\}$ in E. Such a removal can not increase the number of connected components of a graph.*

- *Vertex/Edge substitution if the graph is a labeled one. Such an operation modifies the label of a vertex or an edge and thus transforms the vertex or edge labeling functions.*

The costs of elementary edit operations preserving connectedness which correspond to already defined elementary edit operations are defined as in Definition 8. Using notations of Definition 13, the cost of a vertex-edge insertion is defined as $c_{vi}(v) + c_{ei}(\{v, u\})$. The cost of the removal of a vertex v is equal to:

$$c_{vd}(v) + \sum_{\{v,u\} \in E} c_{ed}(\{v, u\})$$

Definition 14 Edit path preserving connectedness

An edit path of a graph G preserving connectedness is an edit path (Definition 14) composed of elementary edit operations preserving connectedness.

If G_1 and G_2 are unlabeled we assume that no vertex nor edge substitutions are performed.

Proposition 7 *Given two graphs G_1 and G_2 and an edit path P between G_1 and G_2 preserving connectedness. If $C_1^1, \ldots, C_n^1$ and $C_1^2, \ldots, C_n^2$ correspond to the connected components of G_1 and G_2, the sub graphs $\hat{G}_1$ and $\hat{G}_2$ of G_1 and G_2 associated to P (Proposition 3) have n connected components, $\hat{C}_1^1, \ldots, \hat{C}_n^1$ and $\hat{C}_1^2, \ldots, \hat{C}_n^2$, each connected component $(\hat{C}_i^j)_{i \in \{1,\ldots,n\}, j \in \{1,2\}}$ being a connected sub graph of C_i^j.*

Proof:

From the very definition of an edit path preserving connectedness, the sub graph $\hat{G}_1$ has exactly n connected components. Since $\hat{G}_2$ is isomorphic to $\hat{G}_1$, it follows that $\hat{G}_2$ has n connected components.

By Proposition 1, since $\hat{G}_1$ is a graph, each of its connected components is also a simple connected graph. Let us consider one of such connected components $\hat{C}_i^1 = (\hat{V}_i^1, \hat{E}_i^1)$.

Since $\hat{C}_i^1$ is a sub graph of G_1 there exists at least one k in $\{1, \ldots, n\}$ such that $\hat{V}_i^1 \cap V_k^1 \neq \emptyset$.

Let us suppose that there exists $j \in \{1, \ldots, n\}$, $j \neq k$ such that $\hat{V}_i^1 \cap V_j^1 \neq \emptyset$ and let us consider $x \in \hat{V}_i^1 \cap V_k^1$ and $y \in \hat{V}_i^1 \cap V_j^1$. Since both x and y belong to $\hat{V}_i^1$ and $\hat{C}_i^1$ is connected, there exists a path in $\hat{C}_i^1$ which connects x to y. Since $\hat{G}_1$ is obtained from G_1 by edge and vertex removals, such a path should also exist in G_1. This contradicts the fact that x and y belong to two different connected components of G_1.

For each $i \in \{1, \ldots, n\}$ there exists thus only one k such that $\hat{V}_i^1 \subset V_k^1$. Let us denote by $f(i)$ such a k.

Now since $\hat{C}_i^1$ is a graph, we have:

$$\hat{E}_i^1 \subset \mathcal{P}_2(\hat{V}_i^1) \subset \mathcal{P}_2(V_{f(i)}^1)$$

Since $\hat{E}_i^1$ is deduced from E_1 by edge removals we have $\hat{E}_i^1 \subset \hat{E}_1 \subset E_1$. Thus :

$$\hat{E}_i^1 = \left(E_1 \cap \hat{E}_i^1\right) \subset E_1 \cap \mathcal{P}_2(V_{f(i)}^1) = E_i^1$$

We have thus:

$$\left\{ \begin{array}{lcl} \hat{V}_i^1 & \subset & V_{f(i)}^1 \\ \hat{E}_i^1 & \subset & E_{f(i)}^1 \end{array} \right.$$

and $\hat{C}_i^1$ is a connected sub graph of $C_{f(i)}^1$.

The relation f is thus a map from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$ which associates to each i the index $f(i)$ such that $\hat{C}_i^1$ is a connected sub graph of $C_{f(i)}^1$. Let us now suppose that f is not surjective. It means that there exists one $i \in \{1, \ldots, n\}$ such that for any j in $\{1, \ldots, n\}$, the graph $\hat{C}_j^1$ is not a sub graph of C_i^1. In other words:

$$\forall j \in \{1, \ldots, n\},\ \hat{V}_j^1 \cap V_i^1 = \emptyset$$

and thus $\hat{V}_1 \cap V_i^1 = \emptyset$. All vertices of C_i^1 being removed in $\hat{G}_1$, this connected component is removed by the edit path from G_1 to $\hat{G}_1$ which contradicts the fact that P preserves connectedness.

Since f is thus a surjective map from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$ it is therefore bijective. Eventually, we have proved that, up to a renumbering of connected components $(\hat{C}_i^1)_{i \in \{1,\ldots,n\}}$, each $\hat{C}_i^1$ is a sub graph of C_i^1.

The same demonstration holds between connected components of $\hat{G}_2$ and the ones of G_2. $\square$

Note that each connected component of G_1 is related by a bijective relationship to a component of $\hat{G}_1$ which itself is related by an isomorphism to a connected component of $\hat{G}_2$ finally related by an other bijective relationship with a connected component of G_2. An edit path preserving connectedness induces thus a bijective relationship between the connected components of G_1 and G_2.

Proposition 8 *Given two graphs G_1 and G_2 with a same number of connected components, the cost of an edit path preserving connectedness between G_1 and G_2 is equal to the sum of costs of edit paths transforming each connected component of G_1 into a connected component of G_2.*

Proof:

Let us consider such an edit path preserving connectedness P. By Corol-

lary 2:

$$\begin{aligned}\gamma(P) \quad = \quad & \sum_{v\in V_1\setminus\hat{V}_1} c_{vd}(v) + \sum_{e\in E_1\setminus\hat{E}_1} c_{ed}(e) + \\ & \sum_{v\in\hat{V}_1} c_{vs}(v\to g(v)) + \sum_{e\in\hat{E}_1} c_{es}(e\to g(e)) + \\ & \sum_{v\in V_2\setminus\hat{V}_2} c_{vi}(v) + \sum_{e\in E_2\setminus\hat{E}_2} c_{ei}(e)\end{aligned}$$

where g is the isomorphism between $\hat{G}_1$ and $\hat{G}_2$ associated to P (Proposition 3) and $c_{vs}(v\to g(v))$ (resp. $c_{es}(e\to g(e))$) denotes the cost of the rewriting of the labels of v and e in $\hat{G}_1$ into the ones of $g(v)$ and $g(e)$ in $\hat{G}_2$.

Let us denote by n the number of connected components of G_1 and G_2 and let us decompose $G_1 = (V_1, E_1)$, $\hat{G}_1 = (\hat{V}_1, \hat{E}_1)$, $\hat{G}_2 = (\hat{V}_2, \hat{E}_2)$, and $G_2 = (V_2, E_2)$ into their n connected components (label functions are omitted here to prevent overloaded notations):

- $\left(C_i^1 = (V_i^1, E_i^1)\right)_{i\in\{1,\ldots,n\}}$ for G_1,
- $\left(\hat{C}_i^1 = (\hat{V}_i^1, \hat{E}_i^1)\right)_{i\in\{1,\ldots,n\}}$ for $\hat{G}_1$,
- $\left(\hat{C}_i^2 = (\hat{V}_i^2, \hat{E}_i^2)\right)_{i\in\{1,\ldots,n\}}$ for $\hat{G}_2$, and
- $\left(C_i^2 = (V_i^2, E_i^2)\right)_{i\in\{1,\ldots,n\}}$ for G_2.

Using Proposition 7 we have:

$$\begin{aligned} V_1 \setminus \hat{V}_1 &= \cup_{i=1}^n V_i^1 \setminus \hat{V}_i^1 \\ V_2 \setminus \hat{V}_2 &= \cup_{i=1}^n V_i^2 \setminus \hat{V}_i^2 \\ E_1 \setminus \hat{E}_1 &= \cup_{i=1}^n E_i^1 \setminus \hat{E}_i^1 \\ E_2 \setminus \hat{E}_2 &= \cup_{i=1}^n E_i^2 \setminus \hat{E}_i^2 \end{aligned}$$

Moreover, if we denote by g_i the restriction of g to $\hat{C}_i^1$, f_i is an isomorphism from $\hat{C}_i^1$ to $\hat{C}_i^2$ and we have:

$$\begin{aligned}\gamma(P) = \quad \sum_{i=1}^{n} \quad & \sum_{v\in V_i^1\setminus\hat{V}_i^1} c_{vd}(v) + \sum_{e\in E_i^1\setminus\hat{E}_i^1} c_{ed}(e) + \\ & \sum_{v\in\hat{V}_i^1} c_{vs}(v\to g_i(v)) + \sum_{e\in\hat{E}_i^1} c_{es}(e\to g_i(e)) + \\ & \sum_{v\in V_i^2\setminus\hat{V}_i^2} c_{vi}(v) + \sum_{e\in E_i^2\setminus\hat{E}_i^2} c_{ei}(e)\end{aligned}$$

Let us consider R_i the set of vertex and edge removals $V_i^1 \setminus \hat{V}_i^1$ and $E_i^1 \setminus \hat{E}_i^1$, S_i the set of vertex and edge substitutions between $\hat{G}_i^1$ and $\hat{G}_i^2$ encoded by f_i and I_i the set of vertex and edge insertions $V_i^2 \setminus \hat{V}_i^2$ and $E_i^2 \setminus \hat{E}_i^2$. The sequence $P_i = (R_i, S_i, I_i)$ defines a valid edit path which preserves connectedness as P do. Moreover, we have:

$$\gamma(P) = \sum_{i=1}^{n} \gamma(P_i)$$

with

$$\begin{aligned}\gamma(P_i) \quad = \quad & \sum_{v \in V_i^1 \setminus \hat{V}_i^1} c_{vd}(v) + \sum_{e \in E_i^1 \setminus \hat{E}_i^1} c_{ed}(e) + \\ & \sum_{v \in \hat{V}_i^1} c_{vs}(v \to g_i(v)) + \sum_{e \in \hat{E}_i^1} c_{es}(e \to g_i(e)) + \\ & \sum_{v \in V_i^2 \setminus \hat{V}_i^2} c_{vi}(v) + \sum_{e \in E_i^2 \setminus \hat{E}_i^2} c_{ei}(e)\end{aligned}$$

□

Let us now suppose that costs of elementary operations do not depend on their parameter. As previously, we denote such costs using the same symbol than the associated function.

Proposition 9 *Let us consider a forest G_1 and a graph G_2 with a same number of connected components, let us denote by δ_v the maximal number of vertices of their maximum common connected sub graphs in each connected component. Then:*

- *If $c_{vs} = 0$ or*

$$\frac{c_{vd} + c_{vi}}{c_{vs}} > \delta_v$$

and

- *If $c_{es} = 0$ or*

$$\frac{c_{ed} + c_{ei}}{c_{es}} > \delta_v - 1$$

then for any optimal edit path preserving connectedness $P_{opt} = (R_{opt}, S_{opt}, I_{opt})$, the sub graphs of G_1 and G_2, $\hat{G}_1$ and $\hat{G}_2$ defined respectively by the edit operations R_{opt} and (R_{opt}, S_{opt}) are maximum common unlabeled sub graphs of G_1 and G_2.

Proof:

Since the cost of any edit path between G_1 and G_2 is defined as a sum of costs between the connected components of G_1 and G_2, we may restrict our demonstration to a single connected component without loss of generality.

We thus suppose that G_1 is a tree and G_2 a connected graph. Any common sub graph of G_1 and G_2 is a forest. Moreover, for any edit path preserving

connectedness, the associated sub graphs $\hat{G}_1$ and $\hat{G}_2$ are connected. They thus correspond to trees.

Let us consider a non maximal common sub tree G'_1 of G_1 and G_2. Let us additionally consider the edit path $P = (R, S, I)$ associated to G'_1 (Proposition 4). Following Corollary 2, we may compare the costs of the edit paths P and P_{opt} by comparing the two values:

$$\left\{ \begin{array}{lcl} M(P) & = & |V'_1|(c_{vd} + c_{vi}) + |E'_1|(c_{ed} + c_{ei}) - (V'_f c_{vs} + E'_f c_{es}) \\ M(P_{opt}) & = & |\hat{V}_1|(c_{vd} + c_{vi}) + |\hat{E}_1|(c_{ed} + c_{ei}) - (V_f c_{vs} + E_f c_{es}) \end{array} \right.$$

where V_f and V'_f (resp. E_f and E'_f) denote the number of vertices (resp. edges) substitutions performed respectively by P_{opt} and P.

Since G'_1 is a non maximal sub tree we have $|V'_1| < |\hat{V}_1|$ and $|E'_1| = |V'_1| - 1 < |\hat{V}_1| - 1$. Hence if $c_{vs} = c_{es} = 0$ we trivially have $M(P) < M(P_{opt})$.

Let us suppose that both c_{vs} and c_{es} are strictly positive. We have then:

$$\begin{array}{rcl} M(P_{opt}) - M(P) = & c_{vs} \left[(|\hat{V}_1| - |V'_1|) \frac{c_{vd}+c_{vi}}{c_{vs}} - (V_f - V'_f) \right] + \\ & c_{es} \left[(|\hat{E}_1| - |E'_1|) \frac{c_{ed}+c_{ei}}{c_{es}} - (E_f - E'_f) \right] \end{array}$$

Since $|V'_1| < |\hat{V}_1|$ and $|E'_1| < |E_1|$ both $|\hat{V}_1| - |V'_1|$ and $|\hat{E}_1| - |E'_1|$ correspond to positive integer quantities. Thus:

$$\begin{array}{rcl} M(P_{opt}) - M(P) & \geq & c_{vs} \left[\frac{c_{vd}+c_{vi}}{c_{vs}} - (V_f - V'_f) \right] + c_{es} \left[\frac{c_{ed}+c_{ei}}{c_{es}} - (E_f - E'_f) \right] \\ & \geq & c_{vs} \left[\frac{c_{vd}+c_{vi}}{c_{vs}} - V_f \right] + c_{es} \left[\frac{c_{ed}+c_{ei}}{c_{es}} - E_f \right] \end{array} \quad (2)$$

V_f and E_f are respectively bounded by $|\hat{V}_1| \leq \delta_v$ and $|\hat{E}_1| \leq \delta_v - 1$. Using our hypothesis on the ratios $\frac{c_{vd}+c_{vi}}{c_{vs}}$ and $\frac{c_{ed}+c_{ei}}{c_{es}}$ we obtain $M(P_{opt}) - M(P) > 0$ which demonstrates that P can not be an optimal edit path.

If only one of the variable c_{vs} and c_{es} is equal to 0, its associated term on the right side of equation 2 vanishes and the same argument as above holds for the remaining term. □

5 Efficient Computation of the Graph Edit Distance

Given two graphs $G_1 = (V_1, E_1, \mu_1, \nu_1)$ and $G_2 = (V_2, E_2, \mu_2, \nu_2)$, let us suppose without loss of generality that G_1 and G_2 have only one maximum common unlabeled sub graph G. Boths sub graphs of G_1 and G_2, $\hat{G}_1 = (\hat{V}_1, \hat{E}_1, \mu_1, \nu_1)$ and $\hat{G}_2 = (\hat{V}_2, \hat{E}_2, \mu_2, \nu_2)$ are isomorphic to G. We have thus:

$$\begin{cases} |\hat{V}_1| = |\hat{V}_2| = |V| \text{ and} \\ |\hat{E}_1| = |\hat{E}_2| = |E| \end{cases} \tag{3}$$

Moreover, assuming that the costs of elementary operations do not depend on their parameters, the cost of any edit path P may be decomposed into two parts:

$$\gamma(P) = \gamma_{struc}(P) + \gamma_{label}(P)$$

with

$$\begin{cases} \gamma_{struc}(P) = \left|V_1 \setminus \hat{V}_1\right| c_{vd} + \left|V_2 \setminus \hat{V}_2\right| c_{vi} + \left|E_1 \setminus \hat{E}_1\right| c_{ed} + \left|E_2 \setminus \hat{E}_2\right| c_{ei} \\ \gamma_{label}(P) = V_f c_{vs} + E_f c_{es} \end{cases}$$

where V_f (resp. E_f) denotes the number of vertex (resp. edge) substitutions performed by P.

5.1 Computation of the Structural Cost

Since $\hat{V}_i \subset V_i$ and $\hat{E}_i \subset E_i$ for i in $\{1, 2\}$ we have by equation 3:

$$\forall i \in \{1, 2\}, \begin{cases} \left|V_i \setminus \hat{V}_i\right| = |V_i| - |\hat{V}_i| = |V_i| - |V| \\ \left|E_i \setminus \hat{E}_i\right| = |E_i| - |\hat{E}_i| = |E_i| - |E| \end{cases}$$

The cost $\gamma_{struc}(P)$ may thus be written as:

$$\gamma_{struc}(P) = |V_1|c_{vd} + |V_2|C_{vi} + |E_1|c_{ed} + |E_2|c_{ei} - |V|(c_{vd} + c_{ci}) - |E|(c_{ed} + c_{ei})$$

We must emphasize that the unlabeled graph G is uniquely determined by the structure (V_1, E_1) and (V_2, E_2) of G_1 and G_2. Therefore, if these structures belong to a finite set of know structures, G and $\gamma_{\text{struct}}(P)$ may be pre-computed.

5.2 Computation of the Substitution Cost

Let $\{\hat{G}_1^1, \ldots, \hat{G}_1^n\}$ denote the set of sub graphs of G_1 structurally isomorphic to G. In the same way, let us denote by $\{\hat{G}_2^1, \ldots, \hat{G}_2^m\}$ the set of sub graphs of G_2 isomorphic to G. If the structures (V_1, E_1) and (V_2, E_2) of G_1 and G_2 are known, both sets may be pre-computed since they correspond to the occurrences of G into (V_1, E_1) and (V_2, E_2). Any sub graph $\hat{G}_1^i$ may be obtained from G_1 by a same amount of vertex and edge removals. In the same way, G_2 may be constructed from any $\hat{G}_2^j$ with a same number of vertex and edge insertions. Therefore, any edit path transforming G_1 into G_2 and passing through $\hat{G}_1^i$ and $\hat{G}_2^j$ will be associated to a same structural cost and the edit path P mentioned in the previous section is just one of them.

Let Φ_{12} denote the set of automorphisms of G. For each $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$ the automorphism $\varphi \in \Phi$ corresponds to a mapping of the vertices and edges of $\hat{G}_1^i$ into the one of $\hat{G}_2^j$. Such a mapping induces $V_f^{ij}(\varphi)$ and $E_f^{ij}(\varphi)$ substitutions of vertices and edges with non identical labels. More precisely we have:

$$\left\{\begin{array}{lcl} V_f^{ij}(\varphi) & = & \left|\{v \in \hat{V}_1^i \mid \mu_1(v) \neq \mu_2(\varphi(v))\}\right| \\ E_f^{ij}(\varphi) & = & \left|\{e \in \hat{E}_1^i \mid \nu_1(e) \neq \nu_2(\varphi(e))\}\right| \end{array}\right.$$

The substitution cost of any edit path $P_{i,j,\varphi}$ passing through $\hat{G}_1^i$ and $\hat{G}_2^j$ is equal to:

$$\gamma_{\text{label}}(P_{i,j,\varphi}) = V_f^{ij}(\varphi)c_{vs} + E_f^{ij}(\varphi)c_{es}$$

Let us denote by P_{opt} the edit path belonging to the family

$$(P_{i,j,\varphi})_{(i,j,\varphi)\in\{1,\dots,n\}\times\{1,\dots,m\}\times\Phi_{12}}$$

such that:

$$\gamma_{\text{label}}(P_{opt}) = \min_{(i,j)\in\{1,\dots,n\}\times\{1,\dots,m\}} \min_{\varphi\in\Phi_{12}} \gamma_{\text{label}}(P_{i,j,\varphi}) \tag{4}$$

Since $\gamma_{\text{struct}}(P_{i,j,\varphi}) = \gamma_{\text{struct}}(P)$ for any $(i,j,\varphi) \in \{1,\dots,n\}\times\{1,\dots,m\}\times\Phi_{12}$ we have:

$$\left.\begin{array}{l} \forall i \in \{1,\dots,n\} \\ \forall j \in \{1,\dots,m\} \\ \forall \varphi \in \Phi_{12} \end{array}\right\} \gamma(P_{i,j,\varphi}) = \gamma_{\text{struct}}(P) + \gamma_{\text{label}}(P_{i,j,\varphi}) \geq \gamma(P_{opt})$$

5.3 Computation of the Edit Distance

If the elementary operations do not depend on their parameters and if the ratios between $c_{vd}, c_{vi}, c_{ed}, c_{ei}$ and c_{vs}, c_{es} are such that any optimal edit path passes trough a maximum common sub graph of G_1 and G_2 then the optimal edit path should be one of the edit paths $P_{i,j,\varphi}$ mentioned in the previous section. The path P_{opt} being the path of minimal cost among all $P_{i,j,\varphi}$ we have:

$$d(G_1, G_2) = \gamma(P_{opt})$$

If the structures (V_1, E_1) and (V_2, E_2) of G_1 and G_2 are known, their maximum common sub graph G may be pre-computed together with both sets of sub graphs $\{\hat{G}_1^1,\dots,\hat{G}_1^n\}$ and $\{\hat{G}_2^1,\dots,\hat{G}_2^m\}$ and $\gamma_{\text{struct}}(P)$. Then given any label function applied to G_1 and G_2 the optimal edit distance between G_1 and G_2 is computed using equation 4 which requires $\mathcal{O}(nm|\Phi_{12}|)$ calculus. Note that using a finite family of structure graphs, the values of n, m and $|\Phi_{12}|$ are bounded.

6 Conclusion

We have studied in this paper the conditions, on the costs of the elementary operations of edit paths, under which the optimal edit path encoding the edit distance should pass through a maximum unlabeled common sub graph of two graphs. To our knowledge, this point has been the subject of very few researches. This apparent lack of interest is certainly due to the fact that the only bounds that we may find to "force" an optimal edit path to pass through a maximum unlabeled common sub graphs are relative to the two graphs being compared. Such a negative property requires to take the maximum of these bounds for all pairs of graphs in a given data set. However a remarkable property is that these bounds depend only on the structure of the graphs and not on their labels. Therefore, in applications where we are dealing with a small set of structures, but with a possibly infinite number of labellings of these structures, bounds may be restricted to small values. One of the major result of this technical report is to show that in such a case, the computation of the optimal edit distance may be performed efficiently.

In our future work we plan to apply these results to the graph kernel framework where we often have to deal with a small number of structural patterns extracted from graph databases. The extraction process associates labels to each structural pattern whose number of structures remains small.

References

[1] Horst Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):917–922, 1999.

[2] M. Neuhaus and H. Bunke. *Bridging the gap between graph edit distance and kernel machines*. World Scientific Pub Co Inc, 2007.