A Generic Graph Distance Measure Based on Multivalent Matchings

Sébastien Sorlin, Christine Solnon and Jean-Michel Jolion

Summary. Many applications such as information retrieval and classification, involve measuring graph distance or similarity, i.e., matching graphs to identify and quantify their common features.

Different kinds of graph matchings have been proposed, giving rise to different graph similarity or distance measures. Graph matchings may be *univalent* – when each vertex is associated with at most one vertex of the other graph – or *multivalent* – when each vertex is associated with a set of vertices of the other graph. Also, graph matchings may be *exact* – when all vertex and edge features must be preserved by the matching – or *error-tolerant* – when some vertex and edge features may not be preserved by the matching.

The first goal of this chapter is to propose a new graph distance measure based on the search of a best matching between the vertices of two graphs, i.e., a matching minimizing vertex and edge distance functions. This distance measure is generic in the sense that it allows both univalent and multivalent matchings and it is parameterized by vertex and edge distance functions defined by the user depending on the considered application. The second goal of this chapter is to show how to use this generic measure to model and to solve classical graph matching problems such as (sub-)graph isomorphism problem, error-tolerant graph matching, and nonbijective graph matching.

1 Introduction

In many applications such as information retrieval or classification, measuring object similarity is an important issue [1]. Measuring the similarity of two objects consists in identifying and quantifying their commonalities. A dual problem is to measure the distance of these two objects, i.e., identify and quantify their differences.

Graphs are often used to model structured objects, e.g., scene representation [2–5], design objects [6], molecule representations [7, 8], and web documents [9]. Vertices represent object components while edges represent binary relations between these components. Vertices and edges may be labeled by their features. For example, to represent an image by a graph, one usually associates a vertex with each region of the segmented image, and an edge with each couple of vertices corresponding to two adjacent regions. In order to better represent images, each vertex may be labeled

S. Sorlin et al.: *A Generic Graph Distance Measure Based on Multivalent Matchings*, Studies in Computational Intelligence (SCI) **52**, 151–181 (2007)
www.springerlink.com

by the size and the bounding box of its associated region and each edge may be labeled by a value representing how much two regions are connected (by means of the number of adjacent pixels) [2].

1.1 Graph Matchings and Distance Measures

Computing the distance/similarity of two graphs usually involves finding a "best" matching of the graph vertices (i.e., the one that most preserves vertex and edge features) and then quantifying this set of preserved features. Hence, graph distance measures are closely related to graph matching problems and the capacity of a measure to identify the commonalities of graphs depends on the kind of considered matching.

Graph matchings may be *univalent* – when each vertex is associated with at most one vertex of the other graph – or *multivalent* – when each vertex is associated with a set of vertices of the other graph. Also, graph matchings may be *exact* – when all vertex and edge features must be preserved by the matching – or *error-tolerant* – when some vertex and edge features may not be preserved by the matching.

Examples of univalent exact matchings are:

1. Graph isomorphism, that involves finding a bijection between the graph vertices that preserves all vertex and edge features of the graphs and that is used to prove graph equivalence
2. Subgraph isomorphism, that involves finding an injection from the vertices of the first graph to the vertices of a second graph that preserves all vertex and edge features of the first graph and that is used to prove graph inclusion

In many applications, we are looking for similar objects and not "identical" ones and error-tolerant matchings are needed. Examples of univalent error-tolerant matchings are:

1. Maximum common subgraph [10, 30], that looks for the largest matching (with respect to the number of matched vertices) that preserves all the edges of the matched vertices
2. Graph edit distance [10, 30] that looks for the minimum cost set of operations (i.e., vertex and edge insertion, deletion and relabeling) needed to transform the first graph into a graph that is isomorphic to the second graph

Many applications involve comparing objects described at different granularity levels and multivalent matchings are needed. Different graph distance/similarity measures based on multivalent error-tolerant graph matchings have been proposed:

1. Champin and Solnon [6] measure the similarity of design[ed] objects where one single component of an object may play the same role as that of a set of components of another object, depending on the granularity of object description. Therefore, the graph similarity measure is based on multivalent matchings where one vertex in a graph may be associated with a set of vertices of the other graph.
2. Boeres et al. [4] and Deruyver et al. [12] use graph matching to match an image to its model. In this application, the model has a schematic aspect easy to segment while the image is noised and usually over-segmented. Therefore, scene

recognition is better expressed as a multivalent matching problem where a set of vertices of the scene may be matched with a same vertex of the model.

3. Ambauen et al. [2] propose a new graph edit distance to overcome the problem of comparing over and under segmented images. This distance is based on multivalent matchings: two new edit operations – vertex splitting and merging – are introduced in order to merge or to split over- or under-segmented regions.

1.2 Motivation and Outline of the Chapter

Many different graph distance/similarity measures have been proposed in the literature [13, 14]. These measures are based on different definitions of a "best" matching between two graphs depending on the considered application. For example, the graph similarity measure of Boeres et al. [4] is specific to the recognition of brain images, and in this context specific constraints are added (e.g., all model vertices must be mapped and each image vertex must be mapped to exactly one model vertex). Therefore, it is difficult to use this measure in other applications.

Ambauen et al. defines [2] a more generic graph distance measure: the measure is parameterized by the cost of each possible operation and these costs can be chosen depending on the considered application. As in [4], this measure adds an image recognition specific constraint on the considered multivalent matching: the multivalent matching operations (vertex merging and splitting) must be nonoverlapping, i.e., if one wants to link two vertices u and v of one graph to another vertex u', one has to merge u and v and as a consequence, it will not be possible anymore to link u with a vertex v' without linking v to v'. If this constraint makes sense in a context of over-segmented regions, it is not a desirable property in all applications (in particular for the application of [6]). Also, graph distance measure of [2] is not generic enough to express all kinds of multivalent matching problems: for example, it cannot be used to model the problem described in [4].

In [15] Sorlin and Solnon prove that the similarity measure of Champin and Solnon [6] is generic, i.e., it can be used to compute many other similarity measures (including measures of [4] and [2]). However, if it is generic, it is not always straightforward to use. This measure deals with multilabeled graphs and the similarity of two multilabeled graphs is computed with respect to the set of identical labels that are associated by a mapping. These labels are discrete values, and each label is either recovered or lost by a mapping. However, in many applications and in particular in an image recognition context, one has to compare continuous values. For example, the size of a region of an image is a continuous value and in order to compare two regions, one has to compute the difference between their sizes. Furthermore, when two components are merged, one needs an operator to aggregate these continuous values (for example, the sum of the sizes or the average color of a set of merged regions). Finally, some constraints on matchings are difficult to express in [6]. For example, it is difficult to constrain a vertex to be linked to vertices having a given property only. To express these kinds of constraints on matchings, we show in [15] that one can label the graph vertices in such a way that the original matching can be reconstituted from the set of recovered labels. As a consequence, the similarity

of [6] can be used to compute any other similarity measures based on a best graph matching, whatever the constraints on the matching are.

Our goal is to propose a generic graph distance measure, i.e., a unifying framework for all graph matchings and distance measures. This framework offers a better understanding of the different existing matchings and distance measures. It also allows us to define generic algorithms that can be used to compute any kind of graph distance/similarity measures. Indeed, many algorithms have been proposed for computing graph distance measures or solving graph matching problems. However, all these algorithms are dedicated to one problem and cannot be used to solve other kinds of graph matching problems.

Our generic distance has the same power of expression than the similarity measure of Champin and Solnon [6]. However, it is more flexible: it is based on a multivalent matching of the graph vertices like in [6] but it is parameterized by vertex and edge distance functions that can more easily deal with vertex and edge properties (such as labels, real values, etc.).

In Sect. 2, we introduce some definitions and notations needed to define our distance measure. In Sect. 3, we propose a new generic graph distance measure. In Sect. 4, we compare this measure with some classical graph matching problems. In Sect. 5, we prove that our distance and the graph similarity measure of Champin and Solnon [6] are equivalent in the sense that they have the same power of expression. We conclude in Sect. 6 with some computational issues.

2 Definitions and Notations

2.1 Graph

A *graph* is a pair $G = (V, E)$ such that:

1. V is a finite set of *vertices*
2. $E \subseteq V \times V$ is a set of oriented couples of vertices called *edges*

Given an edge $(u, v) \in E$, the vertices u and v are called the *endpoints* of the edge (u, v).

Partial Subgraph and Induced Subgraph

A graph $G' = (V', E')$ is a *partial subgraph* of a graph $G = (V, E)$ (noted $G' \subseteq_p G$) if and only if $V \subseteq V'$ and $E' \subseteq E \cap (V \times V')$.

A graph $G' = (V', E')$ is an *induced subgraph* of a graph $G = (V, E)$ (noted $G' \subseteq_i G$) if and only if $V \subseteq V'$ and $E' = E \cap (V' \times V')$. An induced subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is the graph that contains all the edges of G having their endpoints into V'. As a consequence, an induced subgraph is always a partial subgraph of G (Fig. 1).

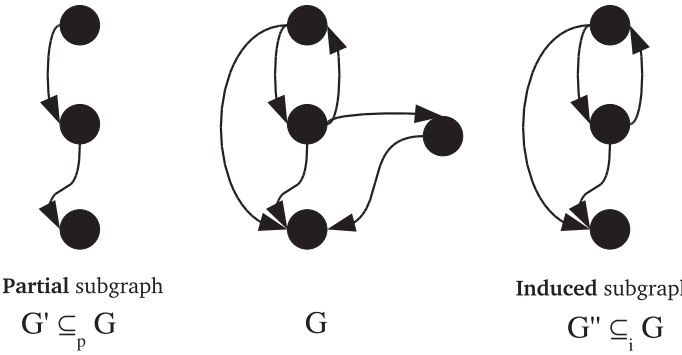

Fig. 1. Examples of a graph G, a partial subgraph G' of G, and an induced subgraph G'' of G

Graph Matching

Given two graphs $G = (V, E)$ and $G' = (V', E')$, a *multivalent matching* m between G and G' is a relation between V and V', i.e., $m \subseteq V \times V'$. Without loss of generality, we shall suppose that $V \cap V' = \emptyset$.

Given a matching m, we note $m(v)$ the set of vertices matched with a vertex v. More formally, we define:

$$\forall v \in V, m(v) \doteq \{v' \in V' | (v, v') \in m\}$$
$$\forall v' \in V', m(v') \doteq \{v \in V | (v, v') \in m\}$$

By extension, when the set of vertices matched with a vertex v is a singleton (i.e., $|m(v)| = 1$), we shall also use $m(v)$ to denote the single vertex that belongs to $m(v)$.

When there is no constraint on the matching, i.e., each vertex may be associated in m with 0, 1 or several vertices, the matching is said to be *multivalent*.

However, one may add constraints on the number of vertices a vertex may be matched with, thus defining matchings that are *partial functions*, *total functions*, *univalent matchings*, *injective matchings*, and *bijective matchings*. Given two graphs $G = (V, E)$ and $G' = (V', E')$, a matching $m \subseteq V \times V'$ is said to be:

1. A *partial function* from G to G' if m links each vertex of V to at most one vertex of G', i.e.:

$$\forall v \in V \ , |m(v)| \leq 1$$

2. A *total function* from G to G' if m links each vertex of V to exactly one vertex of G', i.e.:

$$\forall v \in V \ , |m(v)| = 1$$

3. A *univalent matching* between G and G' if m links each vertex of V and V' to at most one vertex, i.e.:

$$\forall v \in V, |m(v)| \leq 1 \wedge \forall v' \in V', |m(v)| \leq 1$$

4. An *injective matching* from G to G' if m links each vertex of V to a different vertex of V', i.e.:

$$\forall v \in V, |m(v)| = 1 \wedge \forall (u,v) \in V \times V, u \neq v \Rightarrow m(u) \neq m(v)$$

Another definition of an injective matching from G to G' is a matching m such that:

$$\forall v \in V, |m(v)| = 1 \wedge \forall v' \in V', |m(v')| \leq 1$$

5. A *bijective matching* between G and G' if m links each vertex of V (resp. V') to a different vertex of V' (resp. V), i.e.:

$$\forall v \in V, |m(v)| = 1 \wedge \forall (u,v) \in (V \times V), u \neq v \Rightarrow m(u) \neq m(v)$$
$$\forall v' \in V', |m(v')| = 1 \wedge \forall (u',v') \in (V' \times V'), u' \neq v' \Rightarrow m(u') \neq m(v')$$

Another definition of a bijective matching between G and G' is a matching m such that m links each vertex of V and V' to exactly one vertex, i.e.:

$$\forall v \in V, |m(v)| = 1 \wedge \forall v' \in V', |m(v')| = 1$$

Edges Matched by a Matching

Given a matching m of the vertices of two graphs $G = (V, E)$ and $G' = (V', E')$, an edge $(u, v) \in E$ is said to be matched with another edge $(u', v') \in E'$ if and only if $\{(u, u'), (v, v')\} \subseteq m$. By extension, we shall note $m(u, v)$ the set of edges matched with the edge (u, v) by the matching m, i.e.:

$$\forall (u,v) \in E, m(u,v) \doteq \{(u',v') \in E' | u' \in m(u), v' \in m(v)\}$$
$$\forall (u',v') \in E', m(u',v') \doteq \{(u,v) \in E | u \in m(u'), v \in m(v')\}$$

Subgraph Induced by a Matching

Given a matching m of two graphs $G = (V, E)$ and $G' = (V', E')$, the subgraph of G (resp. G') induced by m is noted $G_m = (V_m, E_m)$ (resp. $G'_m = (V'_m, E'_m)$) where V_m and E_m (resp. V'_m and E'_m) are the sets of vertices and edges of G (resp. G') matched with at least one vertex or edge of G' (resp. G), i.e.:

$$V_m = \{v \in V / m(v) \neq \emptyset\} \ , \ E_m = \{(u,v) \in E / m(u,v) \neq \emptyset\}$$
$$V'_m = \{v' \in V' / m(v') \neq \emptyset\} \ , \ E'_m = \{(u',v') \in E' / m(u',v') \neq \emptyset\}$$

Given a matching m of two graphs $G = (V, E)$ and $G' = (V', E')$, if the subgraph of G induced by m, $G_m = (V_m, E_m)$, is equal to G, then, m is an homomorphism between G and G', i.e., m is a function that links each edge of G to an edge of G' (Fig. 2).

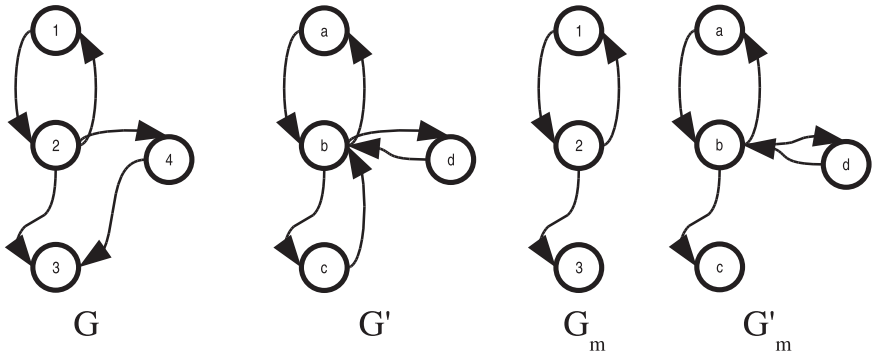

Fig. 2. Two graphs G and G' and their subgraphs induced by the matching $m = \{(1, a), (1, d), (2, b), (3, c)\}$

3 A New Graph Distance Measure

3.1 Vertex and Edge Distance Functions

The first step when computing the distance between two graphs is to match their vertices in order to identify their commonalities. We consider here multivalent graph matchings, i.e., each vertex of a graph may be matched with a – possibly empty – set of vertices of the other graph.

Given a matching m, one has to know for each vertex and each edge how much its properties are recovered by m. Therefore, we assume the existence of a vertex (resp. edge) distance function δ_{vertex} (resp. δ_{edge}) that gives for each vertex v (resp. edge (u, v)) of the two graphs and each set of vertices s_v (resp. set of edges s_e) of the other graph a real value from the interval $[0, +\infty[$ expressing the distance between v (resp. (u, v)) and the set s_v (resp. s_e). More formally, we assume the existence of the two following functions:

$$\delta_{vertex} : (V, \wp(V')) \cup (V', \wp(V)) \rightarrow [0, +\infty[$$
$$\delta_{edge} : (E, \wp(E')) \cup (E', \wp(E)) \rightarrow [0, +\infty[$$

Roughly speaking, the functions δ_{vertex} and δ_{edge} express the *local preferences* on the way to match a vertex and an edge. These functions depend on the considered application and are used to reflect both the similarity knowledge and constraints that a matching must satisfy.

Generally, the distance is equal to $+\infty$ if the vertex v (resp. the edge (u, v)) is not comparable with the set of vertices s_v (resp. the set of edges s_e), i.e., when it is not possible to match v (resp. (u, v)) with s_v (resp. s_e). The distance is equal to 0 when all the properties of v (resp. (u, v)) are recovered by the set s_v (resp. s_e).

For example, if we are looking for an univalent matching (i.e., each vertex is linked to at most one other vertex) that recovers a maximum number of vertices and edges, one can define the functions δ_{vertex} and δ_{edge} as follows:

$$\begin{aligned}
\forall (v, s_v) \in (V \times \wp(V')) \cup (V' \times \wp(V)), \delta_{vertex}(v, s_v) &= 1 \quad \text{if } s_v = \emptyset \\
&= 0 \quad \text{if } |s_v| = 1 \\
&= +\infty \quad \text{otherwise} \\
\forall ((u, v), s_e) \in (E \times \wp(E')) \cup (E' \times \wp(E)), \delta_{edge}((u, v), s_e) &= 1 \quad \text{if } s_e = \emptyset \\
&= 0 \quad \text{if } |s_e| = 1 \\
&= +\infty \quad \text{otherwise}
\end{aligned}$$

3.2 Graph Distance

Given a matching $m \subseteq V \times V'$ of two graphs $G = (V, E)$ and $G' = (V'E')$ and two distance functions δ_{vertex} and δ_{edge}, the distance of these two graphs with respect to the matching m depends on the distance between each vertex (resp. edge) and the set of vertices (resp. edges) they are matched with, i.e.:

$$\begin{aligned}
\delta_m(G, G') = \otimes(\{(v, \delta_{vertex}(v, m(v)))/v \in V \cup V'\} \cup \\
\{((u, v), \delta_{edge}((u, v), m(u, v)))/(u, v) \in E \cup E'\})
\end{aligned} \tag{1}$$

where $\otimes$ is an application-dependent function which is used to aggregate the different vertex and edge distances. Roughly speaking, the function $\otimes$ is used to express the global preferences on the distances of the vertices and the edges of the graphs. The function $\otimes$ should be defined in such a way that the minimal distance between two graphs with respect to a matching is equal to 0 and if the distance between two graphs G and G' is equal to $+\infty$, the matching of these two graphs is not acceptable with respect to the considered application. In most cases, the function $\otimes$ is defined as a sum or a weighted sum of the distances of each component. However, in order to express more sophisticated distances, we do not restrict ourself to this particular case. For example, the function $\otimes$ may be defined in such a way that the distance between two graphs depends on the number of vertices that have at most one incoming or outgoing edge having a distance higher than a threshold.

Formula (1) defines the distance of two graphs with respect to a given matching m between the graph vertices. Now, we define the distance of two graphs G and G' as the distance induced by the best matching, i.e., the matching giving rise to a minimal distance:

$$\delta(G, G') = \min_{m \subseteq V \times V'} \delta_m(G, G') \tag{2}$$

Finally, given two graphs G and G', a distance measure between G and G' is defined as a triple $\delta =< \delta_{vertex}, \delta_{edge}, \otimes >$ where δ_{vertex} is the vertex distance function, δ_{edge} the edge distance function, and $\otimes$ is the function used to aggregate the distances of all vertices and edges of the graphs.

Note that the word "distance" is used here in its common sense: the distance of two graphs is low when the two graphs share a lot of common properties and is equal to 0 (the minimum) when we can find a "perfect" matching of the two graphs

(with respect to the considered application). In the general case, our distance measure does not have the mathematical properties of a classical distance measure and is not a metric. As a consequence, the distance between two graphs may have an infinite value, it may not respect the triangular inequality, nor be symmetric and the distance between a graph and itself may not be equal to 0. However, depending on the functions δ_{vertex}, δ_{edge}, and $\otimes$, our distance measure may be a metric.

3.3 Graph Similarity

We have chosen to define the distance of two graphs but distance and similarity measures are two dual concepts and we could use this graph distance measure to define a graph similarity measure of two graphs. For example, in many applications, the distance between two graphs G and G' is always lower or equal to the sum of the distance between each graph and the empty graph $G_\emptyset$ (i.e., $G_\emptyset = (\emptyset, \emptyset)$). As a consequence, we could define a graph similarity measure using this property:

$$sim(G, G') = 1 - \frac{\delta(G, G')}{\delta(G, G_\emptyset) + \delta(G', G_\emptyset)}$$

4 Equivalence with Other Graph Matchings and Distance/Similarity Measures

In this section, we show how our graph distance measure can be used to solve classical graph matching problems.

In this section, the function $\otimes$ is always defined by the function $\otimes_{\sum}$ that returns the sum of the distances of each vertex and each edge of the two graphs. More formally, we define $\otimes_{\sum} : (V \cup V' \cup E \cup E') \times [0, +\infty[\rightarrow [0, +\infty[$ by:

$$\otimes_{\sum}(S) = \sum_{(u,d) \in S} d + \sum_{((u,v),d) \in S} d$$

4.1 Exact Graph Matchings

In this section we show how to reformulate exact graph matching problems with our graph distance measure. For all these kinds of problems, we are looking for an univalent matching between the vertices of two graphs. As a consequence, the vertex and edge distance functions are defined in such a way that a multivalent matching always involves an infinite positive distance. Furthermore, as these problems are satisfaction problems, the objective is always to find a matching m such that $\delta_m(G, G') = 0$.

Graph Isomorphism

Problem Definition

Given two graphs that have the same number of vertices, the graph isomorphism problem consists in deciding if these two graphs are identical minor a renaming of

their vertices. More formally, two graphs $G = (V, E)$ and $G' = (V', E')$ such that $|V| = |V'|$ are isomorphic if and only if there exists a bijective matching $m \subseteq V \times V'$ such that $(u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$.[1]

Measure Definition

To solve the graph isomorphism problem using our distance measure, we have to define vertex and edge distance functions such that these functions return 0 if the vertex or edge is matched with exactly one element and $+\infty$ otherwise (in order to forbid nonbijective matchings). More formally:

$$\begin{aligned} \forall v \in V \cup V', \forall s_v \subseteq V \cup V', \delta^{iso}_{vertex}(v, s_v) &= \quad 0 \quad \text{if } |s_v| = 1 \\ &= +\infty \quad \text{otherwise} \\ \forall (u, v) \in E \cup E', \forall s_e \subseteq E \cup E', \delta^{iso}_{edge}(u, v, s_e) &= \quad 0 \quad \text{if } |s_e| = 1 \\ &= +\infty \quad \text{otherwise} \end{aligned}$$

$$\delta^{iso} = < \delta^{iso}_{vertex}, \delta^{iso}_{edge}, \otimes_{\sum} >$$

Theorem 1. *Given two graphs $G = (V, E)$ and $G' = (V', E')$, the two following properties are equivalent:*

1. *G and G' are isomorphic*
2. *$\delta^{iso}(G, G') = 0$*

Proof. (1) $\Rightarrow$ (2). By definition, if the two graphs are isomorphic, there exists a bijective matching $m \subseteq V \times V'$ such that $(u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$. As a consequence, $\forall v \in V \cup V', |m(v)| = 1$ (because m is a bijective matching) and $\forall (u, v) \in E \cup E', |m(u, v)| = 1$ (because $\forall (u, v) \in V \times V, (u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$). So, $\delta^{iso}_m(G, G') = 0$ and therefore $\delta^{iso}(G, G') = 0$.

(2) $\Rightarrow$ (1). If $\delta^{iso}(G, G') = 0$, there exists a matching m such that $\delta^{iso}_m(G, G') = 0$. Given the definition of δ^{iso}_{vertex}, m is such that $\forall v \in V \cup V', |m(v)| = 1$. As a consequence, the matching m is a bijective matching. Furthermore, if $\delta^{iso}_m(G, G') = 0$, then, $\forall (u, v) \in E \cup E', |m(u, v)| = 1$. As a consequence, each edge of both graphs is matched with exactly one edge of the other graph, so $(u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$. So, m defines an isomorphic matching between the two graphs and G and G' are isomorphic.

Partial Subgraph Isomorphism (or Monomorphism)

Problem Definition

Given two graphs $G = (V, E)$ and $G' = (V', E')$ such that $|V| \leq |V'|$, the partial subgraph isomorphism problem (or monomorphism problem) consists in deciding

[1] Let us recall that for univalent matchings, when the set of vertices matched with a vertex v is a singleton, i.e., $|m(v)| = 1$, we note $m(v)$ to denote the single vertex, which is an element of $m(v)$.

if the graph G is isomorphic to a partial subgraph of the graph G', i.e., in finding an injective matching $m \subseteq V \times V'$ such that $\forall (u,v) \in V \times V, (u,v) \in E \Rightarrow (m(u), m(v)) \in E'$. The partial subgraph isomorphism problem is used to decide if a graph is included into another graph.

Measure Definition

To solve the partial subgraph isomorphism problem using our distance measure, we have to define vertex and edge distance functions such that these functions return 0 if an element of G is matched with one element (in order to preserve vertices and edges of G) and $+\infty$ otherwise (in order to avoid noninjective matching). Distance functions for vertices and edges of G' just forbid nonunivalent matchings. More formally:

$$G \begin{cases} \forall v \in V, \forall s_v \subseteq V', \delta^{psub}_{vertex}(v, s_v) = 0 & \text{if } |s_v| = 1 \\ & = +\infty \text{ otherwise} \\ \forall (u,v) \in E, \forall s_e \subseteq E', \delta^{psub}_{edge}(u, v, s_e) = 0 & \text{if } |s_e| = 1 \\ & = +\infty \text{ otherwise} \end{cases}$$

$$G' \begin{cases} \forall v \in V', \forall s_v \subseteq V, \delta^{psub}_{vertex}(v, s_v) = 0 & \text{if } |s_v| \leq 1 \\ & = +\infty \text{ otherwise} \\ \forall (u,v) \in E', \forall s_e \subseteq E, \delta^{psub}_{edge}(u, v, s_e) = 0 & \text{if } |s_e| \leq 1 \\ & = +\infty \text{ otherwise} \end{cases}$$

$$\delta^{psub} = < \delta^{psub}_{vertex}, \delta^{psub}_{edge}, \otimes_{\sum} >$$

Theorem 2. *Given two graphs $G = (V, E)$ and $G' = (V', E')$, the two following properties are equivalent:*

1. *The graph G is a partial subgraph of G'*
2. $\delta^{psub}(G, G') = 0$

Proof. (1) $\Rightarrow$ (2). By definition, if G is a partial subgraph of G', there exists an injective matching $m \subseteq V \times V'$ such that $\forall (u,v) \in V \times V, (u,v) \in E \Rightarrow (m(u), m(v)) \in E'$. As a consequence, $\forall v \in V, |m(v)| = 1, \forall v \in V', |m(v)| \leq 1$, and $\forall (u,v) \in E', |m(u,v)| \leq 1$ (because m is an injective matching). Furthermore, $\forall (u,v) \in E, |m(u,v)| = 1$ (because $(u,v) \in E \Rightarrow (m(u), m(v)) \in E'$). So, given the definition of δ^{psub}_{vertex} and δ^{psub}_{edge}, $\delta^{psub}_{m}(G, G') = 0$ and therefore $\delta^{psub}(G, G') = 0$.

(2) $\Rightarrow$ (1). If $\delta^{psub}(G, G') = 0$, then, there exists a matching m such that $\delta^{psub}_{m}(G, G') = 0$. Given the definition of δ^{psub}_{vertex}, $\forall v \in V, |m(v)| = 1$ and $\forall v \in V', |m(v)| \leqslant 1$. As a consequence, m is an injective matching. Furthermore, $\forall (u,v) \in E, |m(u,v)| = 1$. As a consequence, each edge of G is matched with exactly one edge of G' and $(u,v) \in E \Rightarrow (m(u), m(v)) \in E'$. So, there exists an injective matching $m \subseteq V \times V'$ that preserves all the edges of G and, by definition, G is a partial subgraph of G'.

Induced Subgraph Isomorphism

Problem Definition

Given two graphs $G = (V, E)$ and $G' = (V', E')$ such that $|V| \leq |V'|$, the induced subgraph isomorphism problem consists in deciding if the graph G is isomorphic to an induced subgraph of G', i.e., in finding an injective matching $m \subseteq V \times V'$ such that $\forall (u, v) \in V \times V, (u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$. The induced subgraph isomorphism problem is a special case of partial subgraph isomorphism: it adds the constraint that for each couple $(u, v) \in V^2$, if (u, v) is not an edge of G, then, the corresponding vertices in m must neither be an edge of G'.

Measure Definition

The induced subgraph problem between $G = (V, E)$ and $G' = (V', E')$ adds a constraint on each couple of vertices of V (to be or not matched with an edge of G'). To check these constraints, the edge distance function δ_{edge} has to be defined for each couple $(u, v) \in V \times V$ of vertices of G and each subset $s_e \subseteq E'$ of edges of G'. As a consequence, one has to compare the complete graph $G'' = (V, V \times V)$ to the graph $G' = (V', E')$. The vertex distance function must return $+\infty$ if the matching is not injective (rules a, d, and e) and 0 otherwise. The edge distance function must return $+\infty$ if an edge of G is not matched (rule b) or if a couple (u, v) of vertices of G which is not an edge is matched with an edge of G' (rule c) and 0 otherwise. More formally, given a graph $G = (V, E)$ and a graph $G' = (V', E')$, we have to compare the graphs $G'' = (V, V \times V)$ and G' with the two following distance functions:

$$
G'' \begin{cases} a & \forall v \in V, \forall s_v \subseteq V', \delta^{sub}_{vertex}(v, s_v) = 0 \quad \text{if } |s_v| = 1 \\ & \qquad\qquad\qquad\qquad\quad = +\infty \quad \text{otherwise} \\ b & \forall (u, v) \in V^2, \forall s_e \subseteq E', \delta^{sub}_{edge,G}(u, v, s_e) = 0 \quad \text{if } (u, v) \in E \wedge |s_e| = 1 \\ c & \qquad\qquad\qquad\qquad\quad = 0 \quad \text{if } (u, v) \notin E \wedge s_e = \emptyset \\ & \qquad\qquad\qquad\qquad\quad = +\infty \quad \text{otherwise} \end{cases}
$$

$$
G' \begin{cases} d & \forall v \in V', \forall s_v \subseteq V, \delta^{sub}_{vertex}(v, s_v) = 0 \quad \text{if } |s_v| \leq 1 \\ & \qquad\qquad\qquad\qquad\quad = +\infty \quad \text{otherwise} \\ e & \forall (u, v) \in E', \forall s_e \subseteq E, \delta^{sub}_{edge,G}(u, v, s_e) = 0 \quad \text{if } |s_e| \leq 1 \\ & \qquad\qquad\qquad\qquad\quad = +\infty \quad \text{otherwise} \end{cases}
$$

$$
\delta^{sub}_{G} = < \delta^{sub}_{vertex}, \delta^{sub}_{edge,G}, \otimes_{\sum} >
$$

Theorem 3. *Given two graphs $G = (V, E)$ and $G' = (V', E')$, the two following properties are equivalent:*

1. *The graph G is an induced subgraph of G'*
2. *$\delta^{sub}_{G}(G'', G') = 0$, where $G'' = (V, V \times V)$*

Proof. (1) $\Rightarrow$ (2). By definition, if G is a subgraph of G', there exists an injective matching $m \subseteq V \times V'$ such that $\forall (u, v) \in V \times V, (u, v) \in E \Leftrightarrow (m(u), m(v)) \in E'$.

As a consequence, $\forall v \in V, |m(v)| = 1$, $\forall v \in V', |m(v)| \leq 1$, and $\forall (u,v) \in E', |m(u,v)| \leq 1$ (because m is an injective matching). Furthermore, $\forall (u,v) \in E, |m(u,v)| = 1$ (because $\forall (u,v) \in V \times V, (u,v) \in E \Rightarrow (m(u), m(v)) \in E'$) and $\forall (u,v) \in (V \times V) - E, m(u,v) = \emptyset$ (because $\forall (u,v) \in V \times V, (u,v) \notin E \Rightarrow (m(u), m(v)) \notin E'$). So, given the definition of δ_{vertex}^{sub} and $\delta_{edge,G}^{sub}$, $\delta_{mG}^{sub}(G'', G') = 0$ and $\delta_{G}^{sub}(G'', G') = 0$.

(2) $\Rightarrow$ (1). If $\delta_{G}^{sub}(G'', G') = 0$, there exists a matching m such that $\delta_{mG}^{sub}(G'', G') = 0$. Given the definition of δ_{vertex}^{sub}, $\forall v \in V, |m(v)| = 1$ and $\forall v \in V, |m(v)| \leq 1$. As a consequence, m is an injective matching. Furthermore, if m involves a distance equal to 0, then, $\forall (u,v) \in E, |m(u,v)| = 1$. As a consequence, each edge of G is matched with exactly one edge of G', so $\forall (u,v) \in V \times V, (u,v) \in E \Rightarrow (m(u), m(v)) \in E'$. Finally, $\forall (u,v) \in (V \times V) - E, m(u,v) = \emptyset$, and each couple of vertices of G that is not an edge of G is linked to a couple of vertices of G' that is neither an edge of G'. As a consequence, m is an injective matching such that $\forall (u,v) \in V \times V, (u,v) \in E \Leftrightarrow (m(u), m(v)) \in E'$ and G is an induced subgraph of G'.

Generalization of the Subgraph Isomorphism Problem

Problem Definition

Zampelli et al. propose [16] a generalization of the subgraph isomorphism problem. This problem is called "approximate subgraph matching" and consists in looking for a pattern graph into a target graph. It is used for the analysis of biochemical networks. The specificity of this problem is that the pattern graph is composed of mandatory vertices and edges (i.e., vertices and edges that must be preserved by the matching), optional vertices (i.e., vertices that may not be matched), and forbidden edges (i.e., edges that must not be preserved by the matching). Note that an edge having an optional endpoint is optional until its endpoints are matched.[2] More formally, an approximate pattern graph is defined by a tuple $G_p = (V_p, O_p, E_p, F_p)$ where (V_p, E_p) is a graph, $O_p \subseteq V_p$ is the set of optional nodes, and $F_p \subseteq (V_p \times V_p) - E_p$ is the set of forbidden edges. An approximate subgraph matching m between an approximate pattern graph $G_p = (V_p, O_p, E_p, F_p)$ and a target graph $G_t = (V_t, E_t)$ is an univalent matching $m \subseteq V_p \times V_t$ such that:

1. $\forall v \in V_p - O_p, |m(v)| = 1$
2. $\forall (u,v) \in V_p \times V_p, |m(u)| = 1 \wedge |m(v)| = 1$
   $\wedge (u,v) \in E_p \Rightarrow (m(u), m(v)) \in E_t$
3. $\forall (u,v) \in V_p \times V_p, |m(u)| = 1 \wedge |m(v)| = 1$
   $\wedge (u,v) \in F_p \Rightarrow (m(u), m(v)) \notin E_t$
4. $\forall v' \in V_T, |m(v')| \leq 1$

[2] This notion of optional vertices is only useful when we look for a matching satisfying some other constraints. Otherwise, we just have to remove optional vertices and their edges from the pattern graph.

Measure Definition

Solving an approximate subgraph matching problem consists in finding an univalent matching m between G_p and the graph $G' = (V_t, V_t \times V_t)$ such that each mandatory vertex is matched with exactly one vertex (rule a), each optional vertex is matched with at most one vertex (rule b), each edge (u, v) is either matched with a couple of vertices (u', v') of G' which is an edge of G_t (rule d) or is not matched at all if one of its optional endpoints is not matched (rule c), each forbidden edge is not matched (rule e). Finally, the matching must be univalent (rules f and g). More formally, one has to compute the distance between $G = (V_p, E_p \cup F_p)$ and $G' = (V_t, E' = V_t \times V_t)$ with the following vertex and edge distance functions:

$$
G \begin{cases}
a & \forall v \in V_p - O_p, \forall s_v \subseteq V_t, \delta^{agm}_{vertex}(v, s_v) = 0 \quad \text{if } |s_v| = 1 \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise} \\
b & \forall v \in O_p, \forall s_v \subseteq V_t, \delta^{agm}_{vertex}(v, s_v) = 0 \quad \text{if } |s_v| \leq 1 \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise} \\
 & \forall (u, v) \in E_p, \forall s_e \subseteq V_t \times V_t, \\
c & \delta^{agm}_{edge,G_t}(u, v, s_e) = 0 \quad \text{if } s_e = \emptyset \\
d & \qquad\qquad\qquad\qquad\qquad\qquad = 0 \quad \text{if } s_e = \{(u', v')\} \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \wedge (u', v') \in E_t \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise} \\
e & \forall (u, v) \in F_p, \forall s_e \subseteq E', \delta^{agm}_{edge,G_t}(u, v, s_e) = 0 \quad \text{if } s_e = \{(u', v')\} \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \wedge (u', v') \notin E_t \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise}
\end{cases}
$$

$$
G' \begin{cases}
f & \forall v \in V_t, \forall s_v \subseteq V_p, \delta^{agm}_{vertex}(v, s_v) = 0 \quad \text{if } |s_v| \leq 1 \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise} \\
g & \forall (u, v) \in E', \forall s_e \subseteq E_p \cup F_p, \delta^{agm}_{edge,G_t}(u, v, s_e) = 0 \quad \text{if } |s_e| \leq 1 \\
 & \qquad\qquad\qquad\qquad\qquad\qquad = +\infty \quad \text{otherwise}
\end{cases}
$$

$$
\delta^{agm}_{G_t} = < \delta^{agm}_{vertex}, \delta^{agm}_{edge,G_t}, \otimes_{\sum} >
$$

Theorem 4. *Given an approximate pattern graph $G_p = (V_p, O_p, E_p, F_p)$, a target graph $G_t = (V_t, E_t)$ and a mapping $m \subseteq V \times V'$, the two following properties are equivalent:*

1. *m is a solution of the approximate subgraph matching problem between the approximate pattern graph $G_p = (V_p, O_p, E_p, F_p)$ and the target graph $G_t = (V_t, E_t)$*
2. *$\delta^{agm}_{m,G_t}(G, G') = 0$ where $G = (V_p, E_p \cup F_p)$ and $G' = (V_t, V_t \times V_t)$*

Proof. (1) $\Rightarrow$ (2). If m is a solution of the approximate subgraph matching problem then $\forall v \in V_p - O_p, |m(v)| = 1$ (condition 1), $\forall v \in V_t, |m(v)| \leq 1$ and $\forall (u, v) \in V_t \times V_t, |m(u, v)| \leq 1$ (condition 4), $\forall (u, v) \in E_p, m(u, v) = \{(u', v')\} \wedge (u', v') \in E_t$ (condition 2), and $\forall (u, v) \in F_p, m(u, v) = \{(u', v')\} \wedge (u', v') \notin E_t$ (condition

3). As a consequence, given the definition of the vertex and edge distance functions, $\delta^{agm}_{m,G_t}(G, G') = 0$.

(2) $\Rightarrow$ (1). If the distance $\delta^{agm}_{m,G_t}(G, G') = 0$, then the matching m is univalent because, given the vertex and edge distance functions, all nonunivalent matchings give rise to an infinite distance. Furthermore, if $\delta^{agm}_{m,G_t}(G, G') = 0$, then $\forall v \in V_p - O_p, |m(v)| = 1$ so that m respects condition 1. Furthermore, $\forall (u, v) \in E_p, (m(u) \neq \emptyset \wedge m(v) \neq \emptyset) \Rightarrow (m(u, v) = \{(u', v')\} \wedge (u', v') \in E_t)$ and as a consequence, m respects condition 2. Finally, $\forall (u, v) \in F_p, (m(u) \neq \emptyset \wedge m(v) \neq \emptyset) \Rightarrow (m(u, v) = \{(u', v')\} \wedge (u', v') \notin E_t)$ and as a consequence, m respects condition 3 and m is a solution of the approximate subgraph matching problem.

4.2 Error Tolerant Graph Matchings

In this section we show how to model error tolerant graph matching problems as graph distance measures. For all these problems, we are looking for an univalent matching between the vertices of two graphs. As a consequence, the vertex and edge distance functions are chosen in such a way that a nonunivalent matching always gives an infinite positive distance. Furthermore, as these problems are optimization problems, the objective is always to find the matching that gives the lower distance.

Maximum Common Partial Subgraph

Problem Definition

Given two graphs G and G' the maximum common partial subgraph problem consists in finding the size of the largest partial subgraph G'' of G that is isomorphic to a partial subgraph of G'. For this problem, the size of a graph $G = (V, E)$ is defined by the number of its vertices and edges, i.e., $|G| = |V| + |E|$. The maximum common partial subgraph problem is used to quantify the intersection of two graphs and therefore, it can be used to define a graph similarity measure. Indeed, the similarity of two objects a and b is usually defined as $size(a \cap b)/size(a \cup b)$ [17, 18].

Measure Definition

We have to use vertex and edge distance functions that forbid multivalent matchings while encouraging vertices and edges of G and G' to be matched. As a consequence, the vertex and edge distance functions must return $+\infty$ if the element is matched with more than one element, 1 if it is not matched and 0 if the element is matched with exactly one element, i.e.:

$$\begin{aligned} \forall v \in V \cup V', \forall s_v \subseteq V \cup V', \delta^{mcps}_{vertex}(v, s_v) = &\quad 1 \quad \text{if } s_v = \emptyset \\ = &\quad 0 \quad \text{if } |s_v| = 1 \\ = &+\infty \quad \text{otherwise} \end{aligned}$$

$$\forall (u,v) \in E \cup E', \forall s_e \subseteq E \cup E', \delta_{edge}^{mcps}(u,v,s_e) = \begin{array}{ll} 1 & \text{if } s_e = \emptyset \\ 0 & \text{if } |s_e| = 1 \\ +\infty & \text{otherwise} \end{array}$$

$$\delta^{mcps} = < \delta_{vertex}^{mcps}, \delta_{edge}^{mcps}, \otimes_{\sum} >$$

Theorem 5. *Given two graphs $G = (V, E)$ and $G' = (V', E')$, and a mapping $m \subseteq V \times V'$, the two following properties are equivalent:*

1. *m is a mapping that minimizes the distance δ_m^{mcps}*
2. *The subgraph G_m of G induced by the matching m is a maximum common partial subgraph of G and G'*

Proof. The proof is decomposed into two steps, we first show that for every matching $m \subseteq V \times V'$ such that $\delta_m^{mcps}(G, G') = d \neq +\infty$, the induced subgraph G_m of G is a common partial subgraph of G and G' and $|G_m| = (|G| + |G'| - d)/2$. In a second step, we show that, if there exists a subgraph G'' of G isomorphic to a partial subgraph of G', then, we can find a matching m having a distance d equal to $|G| + |G'| - 2 * |G''|$ and such that $G'' = G_m$, the subgraph induced by the mapping m. Then, as we prove that each common partial subgraph G'' corresponds to a mapping inducing a noninfinite distance inverse to the size of G'' (and conversely), the property holds.

$\delta_m^{mcps}(G, G') = d \neq +\infty \Rightarrow G_m$ is a common subgraph of G and G' such that $|G_m| = (|G| + |G'| - d)/2$. Given the vertex and edge distance functions, if $\delta_m^{mcps}(G, G') \neq +\infty$ then m is an univalent matching (because all nonunivalent matchings give an infinite distance). By definition, the subgraph $G_m = (V_m, E_m)$ of G induced by m is a partial subgraph of G and the subgraph $G'_m = (V'_m, E'_m)$ of G' induced by m is a partial subgraph of G'. Given the definition of an induced subgraph and knowing that the mapping is univalent, the matching m is a bijective matching between the vertices of G_m and G'_m such that $(u, v) \in E_m \Leftrightarrow (m(u), m(v)) \in E'_m$. As a consequence, G_m and G'_m are isomorphic and G_m is a common partial subgraph of both G and G'. Given the vertex and edge distance functions, if $\delta_m^{mcps}(G, G') = d \neq +\infty$ then $d = |G| + |G'| - |G_m| - |G'_m|$. As G_m and G'_m are isomorphic, then $|G_m| = |G'_m|$. As a consequence, $|G_m| = (|G| + |G'| - d)/2$ and the property holds.

G'' is a common subgraph of G and $G' \Rightarrow \exists m$ such that $\delta_m^{mcps}(G, G') = |G| + |G'| - 2 * |G''|$ and $G'' = G_m$. If there exists a common subgraph $G'' = (V'', E'')$ of $G = (V, E)$ and $G' = (V', E')$, then, by definition of a common subgraph, there exists at least one graph $G''' = (V''' \subseteq V', E''' \subseteq E')$ and a bijective matching $m \subseteq V'' \times V'''$ such that $(u, v) \in E'' \Leftrightarrow (m(u), m(v)) \in E'''$. As a consequence, the matching m is such that $\forall v \in V'' \cup V''', |m(v)| = 1$ (because m is a bijective matching), $\forall (u, v) \in E'' \cup E''', |m(u, v)| = 1$ (because m is such that $(u, v) \in E'' \Leftrightarrow (m(u), m(v)) \in E'''$. Furthermore, by definition, m is such that $\forall v \in V - V'', m(v) = \emptyset$, $\forall v \in V' - V''', m(v) = \emptyset$, $\forall (u, v) \in E - E'', m(u, v) = \emptyset$, and $\forall (u, v) \in E' - E''', m(u, v) = \emptyset$. As a consequence, $\delta_m^{mcps}(G, G') = |G| + |G'| -$

$|G''| - |G'''|$. G'' and G''' are isomorphic, so, $|G''| = |G'''|$ and $\delta_m^{mcps}(G, G') = |G| + |G'| - 2 * |G''|$. The property holds.

Maximum Common Induced Subgraph

Problem Definition

Given two graphs G and G' the maximum common induced subgraph problem consists in finding the largest induced subgraph G'' of G that is isomorphic to an induced subgraph of G'. For this problem, the size of a graph $G = (V, E)$ is defined by the number of its vertices, i.e., $|G| = |V|$. As the maximum common partial subgraph, the maximum common induced subgraph problem is used to define an intersection between two graphs and a corresponding graph similarity measure [10].

Measure Definition

To solve the maximum common subgraph problem using our distance measure, we have to use vertex and edge distance functions encouraging vertices of G to be matched while forbidding matchings that do not correspond to common induced subgraph. So, similarly to the induced subgraph isomorphism problem, the edge distance function must check a constraint (and so be defined) for each couple of vertices of both the graphs. As a consequence, complete graphs must be compared. The vertex distance function encourages the vertices of G to be matched (rule a) and the edge distance function returns $+\infty$ when a couple of vertices (u, v) of G (resp. (u', v') of G') is linked to a couple of vertices (u', v') of G' (resp. (u, v) of G) such that $(u, v) \in E \not\Leftrightarrow (u', v') \in E'$ (rule b) (resp. rule d). Finally, the matching must be univalent (rule c). More formally, we have to compute the distance of the graph $G_2 = (V, V \times V)$ with the graph $G'_2 = (V', V' \times V')$ by using the following vertex and edge distance functions:

$$G_2 \begin{cases} a\ \forall v \in V, \forall s_v \subseteq V', \delta^{mcs}_{vertex}(v, s_v) = & 1 & \text{if } s_v = \emptyset \\ & = & 0 & \text{if } |s_v| = 1 \\ & = & +\infty & \text{otherwise} \\ & \forall (u, v) \in V^2, \forall s_e \subseteq V'^2, & & \\ b & \delta^{mcs}_{edge,GG'}(u, v, s_e) = & 0 & \text{if } s_e = \emptyset \\ & = & 0 & \text{if } s_e = \{(u', v')\} \\ & & & \wedge((u, v) \in E \Leftrightarrow (u', v') \in E') \\ & = & +\infty & \text{otherwise} \end{cases}$$

$$G'_2 \begin{cases} c\ \forall v \in V', \forall s_v \subseteq V, \delta^{mcs}_{vertex}(v, s_v) = & 0 & \text{if } |s_v| \leq 1 \\ & = & +\infty & \text{otherwise} \\ & \forall (u, v) \in V'^2, \forall s_e \subseteq V^2, & & \\ & \delta^{mcs}_{edge,GG'}(u, v, s_e) = & 0 & \text{if } s_e = \emptyset \\ & = & 0 & \text{if } s_e = \{(u', v')\} \\ & & & \wedge((u, v) \in E' \Leftrightarrow (u', v') \in E) \\ & = & +\infty & \text{otherwise} \end{cases}$$

$$\delta^{mcs}_{GG'} = < \delta^{mcs}_{vertex}, \delta^{mcs}_{edge,GG'}, \otimes_{\sum} >$$

Theorem 6. *Given two graphs $G = (V, E)$ and $G' = (V', E')$, and a mapping $m \subseteq V \times V'$, the two following properties are equivalent:*

1. *m is a mapping that minimizes the distance $\delta^{mcs}_{m,GG'}$*
2. *The subgraph G_m of G induced by the matching m is a maximum common induced subgraph of G and G'*

Proof. The proof is decomposed into two steps. We first show that, for every matching $m \subseteq V \times V'$ such that $\delta^{mcs}_{GG'm}(G, G') = d \neq +\infty$, the subgraph G_m of G induced by the mapping m is an induced common subgraph of G and G' such that $|G_m| = |G| - d$. In a second step, we show that, if there exists an induced subgraph G'' of G isomorphic to an induced subgraph of G', then, we can find a matching m having a distance d equal to $|G| - |G''|$ and such that $G'' = G_m$, the subgraph of G induced by the matching m. Then, as we prove that each common induced subgraph G'' corresponds to a mapping inducing a noninfinite distance inverse to the size of G'' (and reversely), the property holds.

$\delta^{mcs}_{mGG'}(G_2, G'_2) = d \neq +\infty \Rightarrow G_m$ is a common induced subgraph of G and G' such that $|G_m| = |G| - d$. Given the vertex and edge distance functions, if $\delta_{mGG'}(G_2, G'_2) \neq +\infty$ then m is an univalent matching (because all nonunivalent matchings give a distance equal to $+\infty$). By definition, the subgraph $G_{2m} = (V_{2m}, E_{2m})$ of G_2 induced by m is a partial subgraph of G_2 and of G. Furthermore, given the definition of the edge distance function, $(u, v) \in E_{2m} \Rightarrow (u, v) \in E$ and $(u, v) \notin E_{2m} \Rightarrow (u, v) \notin E$. As a consequence, G_{2m} is an induced (i.e., a nonpartial) subgraph of G and $G_{2m} = G_m$. In the same way, we can also prove that the subgraph $G'_{2m} = (V'_{2m}, E'_{2m})$ of G'_2 induced by m is an induced subgraph of G' and that $G'_{2m} = G'_m$. Finally, m is a univalent matching and, given the definitions of the vertex and edge distance functions, m is such that $(u, v) \in E_m \Leftrightarrow (m(u), m(v)) \in E'_m$ so, m defines an isomorphism matching between G_m and G'_m. As a consequence G_m is a common induced subgraph of G and G'. Finally, as only the number of nonrecovered vertices of G influences (positively) the distance, $|G_m| = |G| - d$.

G'' is a common induced subgraph of G and $G' \Rightarrow \exists m$ such that $\delta^{mcs}_{mGG'}(G_2, G'_2) = |G| - |G''|$ and such that $G_m = G''$. If there exists a common induced subgraph $G'' = (V'', E'')$ of $G = (V, E)$ and $G' = (V', E')$, then, by definition of an induced common subgraph, there exists at least one induced subgraph $G''' = (V''', E''')$ of G' and one bijective matching $m \subseteq V'' \times V'''$ such that $(u, v) \in E'' \Leftrightarrow (m(u), m(v)) \in E'''$. Given the vertex and edge distance functions, we can see that the distance $\delta^{mcs}_{mGG'}(G_2, G'_2)$ is equal to $|G| - |G''|$ and that $G_m = G''$.

Graph Edit Distance (ged)

Problem Definition

Given two labeled graphs G and G' (i.e., graphs where a label is associated with each vertex and each edge), the graph edit distance of G and G' is the minimum cost

set of weighted operations needed to transform G into G'. Considered operations are insertions, substitutions (i.e., relabeling), and deletions of vertices and edges. Bunke shows in [10] that, when considering appropriate weight definitions, ged is closely related to the maximum common subgraph, and therefore it is also closely related to the similarity measure based on it.

Bunke and Jiang define formally the graph edit distance in [19]. A *labeled graph* is defined by a tuple $G = (V, E, L, \alpha, \beta)$ where V is a set of vertices, E is a set of edges, L is a set of labels, $\alpha : V \rightarrow L$ is a total function labeling the vertices of G and $\beta : E \rightarrow L$ is a total function labeling the edges of G. Given two labeled graphs $G = (V, E, L, \alpha, \beta)$ and $G' = (V', E', L', \alpha', \beta')$, an *error tolerant graph matching* is an univalent matching $m \subseteq V \times V'$. The vertex $u \in V$ is *substituted* by the vertex v if $m(u) = v$. If $\alpha(u) = \alpha'(m(u))$, the substitution is called an *identical substitution*, otherwise, it is a *nonidentical substitution*. Each vertex $v \in V$ such that $m(v) = \emptyset$ is *deleted* by m and each vertex $v' \in V'$ such that $m(v') = \emptyset$ is *inserted* by m. The same terms are used for the substituted, deleted, and inserted edges of the graphs. A cost c_{vs} (resp. c_{vi} and c_{vd}) is associated with the nonidentical vertex substitutions (resp. insertions and deletions) and a cost c_{es} (resp. c_{ei} and c_{ed}) is associated with the nonidentical edge substitutions (resp. insertions and deletions). Once the six operation costs are set, the *cost of an error tolerant graph matching* m is defined as the sum of the costs of each operation induced by m. Finally, the *graph edit distance* between two graphs is defined as the minimum cost error-tolerant graph matching.

Measure Definition

Each univalent graph matching of our model corresponds to an error-tolerant graph matching of Bunke and Jiang [19]. As a consequence, if the vertex and edge distance functions are defined in such a way that they reproduce the cost of each operation while forbidding nonunivalent matchings, the distance between G_1 and G_2 with respect to an univalent mapping m corresponds to the cost of the error-tolerant graph matching defined by m. More formally, to compute the graph edit distance between two labeled graphs $G = (V, E, L, \alpha, \beta)$ and $G' = (V', E', L', \alpha', \beta')$, we have to compare the graphs $G_1 = (V, E)$ and $G_2 = (V', E')$ with the following vertex and edge distance functions:

$$
G_1 \begin{cases}
\forall v \in V, \forall s_v \subseteq V', & \\
\quad \delta^{ged}_{vertex,GG'}(v, s_v) = c_{vd} & \text{if } s_v = \emptyset \\
\qquad\qquad\qquad\quad = 0 & \text{if } s_v = \{v'\} \wedge \alpha(v) = \alpha'(v') \\
\qquad\qquad\qquad\quad = c_{vs} & \text{if } s_v = \{v'\} \wedge \alpha(v) \neq \alpha'(v') \\
\qquad\qquad\qquad\quad = +\infty & \text{if } |s_v| > 1 \\
\forall (u, v) \in E, \forall s_e \subseteq E', & \\
\quad \delta^{ged}_{edge,GG'}(u, v, s_e) = c_{ed} & \text{if } s_e = \emptyset \\
\qquad\qquad\qquad\quad = 0 & \text{if } s_e = \{(u', v')\} \wedge \beta((u, v)) = \beta'((u', v')) \\
\qquad\qquad\qquad\quad = c_{es} & \text{if } s_v = \{(u', v')\} \wedge \beta((u, v)) \neq \beta'((u', v')) \\
\qquad\qquad\qquad\quad = +\infty & \text{if} |s_e| > 1
\end{cases}
$$

$$G_2 \begin{cases} \forall v \in V', \forall s_v \subseteq V, \delta^{ged}_{vertex,GG'}(v, s_v) = c_{vi} & \text{if } s_v = \emptyset \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad = 0 & \text{if } |s_v| = 1 \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad = +\infty & \text{if } |s_v| > 1 \\ \forall (u, v) \in E', \forall s_e \subseteq E, \delta^{ged}_{edge,GG'}(u, v, s_e) = c_{ei} & \text{if } s_e = \emptyset \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad = 0 & \text{if } |s_e| = 1 \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad = +\infty & \text{if } |s_e| > 1 \end{cases}$$

$$delta^{ged}_{GG'} = < \delta^{ged}_{vertex,GG'}, \delta^{ged}_{edge,GG'}, \otimes_{\sum} >$$

Theorem 7. *Given two labeled graphs G and G' ($G = (V, E, L, \alpha, \beta)$ and $G' = (V', E', L', \alpha', \beta')$), the graph edit distance of Bunke and Jiang [19] is equal to the distance $\delta^{ged}_{GG'}(G_1, G_2)$, where $G_1 = (V, E)$ and $G_2 = (V', E')$.*

Proof. The proof of correctness is trivially done first by proving the equivalence between the set of error-tolerant graph matchings and the set of univalent graph matchings and second, by proving that, given an univalent matching m, the computed distance with respect to m is equal to the cost of the error-tolerant graph matching m.

4.3 Multivalent Graph Matchings

In this section we show how to model different multivalent graph matching problems as graph distance measures. As these problems are optimization problems, the objective is always to find the matching that gives the lowest distance.

Extended Graph Edit Distance

Problem Definition

Ambauen et al. [2] propose to extend the graph edit distance with two new operations: vertex splitting – to split one vertex of G into several vertices of G' – and vertex merging – to merge several vertices of G into one single vertex of G'. These two new operations are added in order to merge over-segmented regions and to split under-segmented regions. Each of these new operations is weighted by a cost c_{split} and c_{merge} (but, in [2], these costs are set to 0). Finally, nonoverlapping constraints are added on the two kinds of "multivalent matching" operations (vertex merging and splitting): if one wants to link two vertices u and v of one graph to another vertex u', one has to merge u and v. As a consequence, it will not be possible anymore to link u with a vertex v' without linking v to v'.

Measure Definition

We cannot model the extended graph edit distance in the same way as that for (nonextended) graph edit distance: the nonoverlapping constraint could not be checked. To take into account this constraint, the matching m must represent the operations that are done. We introduce an "operation graph" $G_O = (V_O, E_O = V_O \times V_O)$. This

graph is a complete graph that has as many vertices as the two graphs to compare, i.e., $|V_O| = |V| + |V'|$. Its vertices must be matched with the vertices of the two graphs to compare, i.e., we are looking for a matching $m \subseteq V_O \times (V \cup V')$. Depending on the way the vertices of G_O are matched with the vertices of G and G', the matching m represents a set of edit operations between G and G'. When a vertex of G_O is only matched with a vertex v of G, the vertex v is deleted. When a vertex of G_O is only matched with a vertex v' of G', the vertex v' is inserted. When a vertex of G_O is matched with a vertex v of G and a vertex v' of G', the vertex v is substituted by the vertex v'. In the same way, the edges of G_O model the edge deletions, insertions, and substitutions. When a vertex of G_O is matched with some vertices of G (resp. G'), these vertices are merged (resp. splitted). If the vertices of G and G' must be matched with exactly one vertex of G_O, every matching satisfying this constraint corresponds to a set of edition operations of the extended graph edit distance satisfying the nonoverlapping constraint.

More formally, to model the extended graph edit distance between $G = (V, E, L, \alpha, \beta)$ and $G' = (V', E', L', \alpha', \beta')$, with our generic graph distance measure, one have to compare the graph $G'' = (V'' = V \cup V', E'' = E \cup E')$ (let us recall that $V \cap V' = \emptyset$) and the complete graph $G_O = (V_O, E_O = V_O \times V_O)$ such that $|V_O| = |V| + |V'|$ (because there is at most one edition operation for each vertex of G and G'). The distance functions δ^{eged}_{vertex} and δ^{eged}_{edge} must constrain the vertices of G and G' to be matched with exactly one vertex of G_O. The cost of the edition operations must be computed on the vertices of the graph G_O:

$$\begin{cases} \forall v \in V'', \forall s_v \subseteq V_O, \delta^{eged}_{vertex}(v, s_v) = 0 & \text{if } |s_v| = 1 \\ \qquad\qquad\qquad\qquad = +\infty & \text{otherwise} \\ \forall (u,v) \in E'', \forall s_e \subseteq E_O, \delta^{eged}_{edge}((u,v), s_e) = 0 & \\ \qquad \forall v_o \in V_O, \forall s_v \subseteq V'', & \\ \qquad \delta^{eged}_{vertex}(v_o, s_v) = 0 & \text{if } s_v = \emptyset \\ \qquad\qquad = match_v(s_v \cap V_1, s_v \cap V_2) & \text{otherwise} \\ \forall (u_o, v_o) \in E_O, \forall s_e \subseteq E'', & \\ \qquad \delta^{eged}_{edge}((u_o, v_o), s_e) = 0 & \text{if } s_e = \emptyset \\ \qquad\qquad = match_e(s_e \cap E_1, s_e \cap E_2) & \text{otherwise} \end{cases}$$

$$\delta^{eged} = < \delta^{eged}_{vertex}, \delta^{eged}_{edge}, \otimes_{\sum} >$$

where $match_v(s_v, s'_v)$ (resp. $match_e(s_e, s'_e)$) is the cost needed to match the (possibly empty) set of vertices s_v (resp. edges s_e) of G_1 to the (possibly empty) set of vertices s'_v (resp. edges s'_e) of G_2. More formally, the functions $match_v : \wp(V_1) \times \wp(V_2) \rightarrow [0, +\infty[$ et $match_e : \wp(E_1) \times \wp(E_2) \rightarrow [0, +\infty[$ are defined by:

$$\begin{array}{lll} a\ \forall s_v \subseteq V, \forall s'_v \subseteq V', & & \\ \quad match_v(s_v, s'_v) & = merge(s_v) + merge(s'_v) & \\ & \quad + subst_v(s_v, s'_v) & \text{if } s_v \neq \emptyset \wedge s'_v \neq \emptyset \\ b & = merge(s_v) + del_v(s_v) & \text{if } s_v \neq \emptyset \wedge s'_v = \emptyset \\ c & = merge(s'_v) + ins_v(s'_v) & \text{if } s_v = \emptyset \wedge s'_v \neq \emptyset \end{array}$$

$$
\begin{array}{lll}
d\ \forall s_e \subseteq V, \forall s'_e \subseteq V', & & \\
\quad match_e(s_e, s'_e) & = subst(s_e, s'_e) & \text{if } s_e \neq \emptyset \wedge s'_e \neq \emptyset \\
e & = del_e(s_e) & \text{if } s_e \neq \emptyset \wedge s'_e = \emptyset \\
f & = ins_e(s'_e) & \text{if } s_e = \emptyset \wedge s'_e \neq \emptyset
\end{array}
$$

The function $merge(s_v)$ is the cost needed to merge the vertices of the set s_v, the function $subst_v(s_v, s'_v)$ (resp. $subst_e(s_e, s'_e)$) is the cost needed to substitute the vertices (resp. the edges) of the set s_v (resp. s_e) by the vertices (resp. the edges) of the set s'_v (resp. s'_e). $ins_v(s_v)$ (resp. $ins_e(s_e)$) is the cost need to insert the vertices (resp. edges) of the set s_v (resp. s_e) and $del_v(s_v)$ (resp. $del_e(s_e)$) is the cost of their deletion.

Theorem 8. *Given two (mono)-labeled graphs $G = (V, E, L, \alpha, \beta)$ and $G' = (V', E', L', \alpha', \beta')$, the extended graph edit distance is equal to $\delta^{eged}(G_O, G'')$ where $G'' = (V_1 \cup V_2, E_1 \cup E_2)$ and $G_O = (V_O, V_O \times V_O)$ such that $|V_O| = |V_1| + |V_2|$.*

Proof. The proof of correctness is easy: each matching m giving rise to a noninfinite distance correspond to a sequence of edition operations of the extended graph edit distance (and reversely). Furthermore, the vertex and edge distance functions are defined in such a way that the cost of this sequence is equal to the distance induced by m.

Nonbijective Graph Matching Problem

Definition

Boeres et al. [4] propose a nonbijective graph similarity measure to compare medical images of brains to an image model of a brain. The model has a schematic aspect easy to segment whereas the real image is noised and generally over-segmented. As a consequence, when comparing the image graph to the model graph, one has to use a nonbijective graph matching where the vertices of the model graph may be linked to a set of vertices of the image graph in order to merge over-segmented regions of the image graph. The similarity between an image graph and its model is computed with respect to vertex and edge similarity matrices and the problem consists in finding the best matching (the one with the highest similarity) that satisfies application-dependent constraints. More formally, two graphs are used to represent the problem: the model graph $G = (V, E)$ and the image graph $G' = (V', E')$ (with $|V| \leq |V'|$). A solution is a matching $m \subseteq V \times V$ between G and G' such that each vertex of G is linked to a nonempty set of connected vertices of G' (i.e., $\forall v \in V, |m(v)| \geq 1$ and the subgraph induced by $m(v)$ is a connected graph), and each vertex of G' is linked to exactly one vertex of G (i.e., $\forall v \in V', |m(v)| = 1$). Finally, some couples of vertices cannot be matched together. Given any matching that respects these constraints, a similarity measure $sim[Boeres]_m$ is computed with respect to a vertex and an edge similarity function $sm_v : V \times V' \rightarrow [0, 1]$ and $sm_e : E \times E' \rightarrow [0, 1]$ as follows:

$$sim[Boeres]_m = \frac{\sum_{(u,v)\in m} sm_v(u,v)}{|V|.|V'|} + \frac{\sum_{(u,v)\in (V\times V')-m} 1 - sm_v(u,v)}{|V|.|V'|} + \frac{\sum_{((u,u'),(v,v'))\in E\times E', \{(u,v),(u',v')\}\in m} sm_e((u,u'),(v,v'))}{|E|.|E'|} + \frac{\sum_{((u,u'),(v,v'))\in E\times E', \{(u,v),(u',v')\}\notin m} 1 - sm_e((u,u'),(v,v'))}{|E|.|E'|}$$

Measure Definition

By properly choosing vertex and edge distance functions δ_{vertex} and δ_{edge}, we can model the similarity of Boeres et al. as a graph distance measure. The vertex distance function returns $+\infty$ when the matching violates a constraint and both the vertex and edge distance functions reproduce the similarity matrices sm_v and sm_e. More formally:

$$G \begin{cases} \forall v \in V, \forall s_v \subseteq V', \delta^{nbgm}_{vertex}(v, s_v) = \sum_{v'\in s_v} 1 - sm_v(v,v') \\ \qquad\qquad + \sum_{v'\in V'-s_v} sm_v(v,v') \\ \qquad\qquad \text{if } connected(s_v) \\ \qquad\qquad = +\infty \text{ otherwise} \\ \forall (u,v) \in E, \forall s_e \subseteq E', \\ \qquad \delta^{nbgm}_{edge}((u,v), s_e) = \sum_{(u',v')\in s_e} 1 - sm_e((u,v),(u',v')) \\ \qquad\qquad + \sum_{(u',v')\in E'-s_e} sm_e((u,v),(u',v')) \end{cases}$$

$$G' \begin{cases} \forall v \in V', \forall s_v \subseteq V, \delta^{nbgm}_{vertex}(v, s_v) = 0 \text{ if } allowed(v, s_v) \\ \qquad\qquad = +\infty \text{ otherwise} \\ \forall (u',v') \in E', \forall s_e \subseteq E, \delta^{nbgm}_{edge}((u',v'), s_e) = 0 \end{cases}$$

$$\delta^{nbgm} = < \delta^{nbgm}_{vertex}, \delta^{nbgm}_{edge}, \otimes_{\sum} >$$

where $connected$ and $allowed$ are two predicates introduced to check the constraints. $connected$ is false when a vertex of the model is not matched or when it is matched with a nonconnected set of vertices and true otherwise. $allowed$ is false when a vertex of the image is not matched with only one allowed vertex of the model and true otherwise:

$$\forall v \in V, \forall s_v \subseteq V', connected(s_v) = \text{true if } s_v \text{ is a nonempty set of connected vertices; false otherwise}$$

$$\forall v \in V', \forall s_v \subseteq V, allowed(v, s_v) = \text{true if } s_v = \{v'\} \wedge (v,v') \text{ is allowed; false otherwise}$$

Theorem 9. *If the matching m minimizing the distance $\delta_m^{nbgm}(G, G')$ gives rise to a noninfinite distance, then m is the matching that maximizes the similarity of Boeres et al. otherwise, there does not exist a mapping that satisfies the hard constraints of the similarity of Boeres et al.*

Proof. We can easily prove that, thanks to the predicates $connected$ and $allowed$, the distance between G and G' with respect to a matching m is equal to $+\infty$ if and only if m is a matching that violates at least one hard constraint. Finally, by decomposing the vertex and edge distance functions, we can prove that the distance δ^{nbgm} is inverse to the similarity of [4] and as a consequence, the matching minimizing the distance δ^{nbgm} is the matching that maximizes the similarity of Boeres et al.

5 Comparison with the Graph Similarity Measure of Champin and Solnon

In [15], we show that the similarity of Champin and Solnon [6] is generic in the sense that, by properly instantiating parameters of this measure, it can be used to solve all the graph matching problems listed earlier. In this section, we briefly present the graph similarity measure of Champin and Solnon and we show that this measure and our graph distance measure are equivalent.

5.1 Definition of the Graph Similarity of Champin and Solnon

The measure of Champin and Solnon is defined for multilabeled graphs, i.e., graphs where a nonempty set of labels is associated with each vertex and each edge of the graphs. More formally, given a set L_V of vertex labels and a set L_E of edge labels, a multilabeled graph G is defined by a tuple $G = \langle V, r_V, r_E \rangle$ such that:

- V is a finite set of vertices
- $r_V \subseteq V \times L_V$ is a relation associating labels to vertices, i.e., r_V is the set of couples (v_i, l) such that vertex v_i is labeled by l
- $r_E \subseteq V \times V \times L_E$ is a relation associating labels to edges, i.e., r_E is the set of triples (v_i, v_j, l) such that edge (v_i, v_j) is labeled by l. Note that the set E of edges of the graph can be defined by $E = \{(v_i, v_j) | \exists l, (v_i, v_j, l) \in r_E\}$

The first step for measuring graph similarity of two graphs $G = \langle V, r_V, r_E \rangle$ and $G' = \langle V', r_{V'}, r_{E'} \rangle$ defined over the same set L_V and L_E of vertex and edge labels is to match their vertices. The matching m considered here is multivalent, i.e., $m \subseteq V \times V'$.

Once a multivalent mapping is defined, the next step is to identify the set of features that are common to the two graphs with respect to this matching. This set contains all the features from both G and G' whose vertices (resp. edges) are matched by m to at least one vertex (resp. edge) that has the same feature. More formally, the set of common features $G \sqcap_m G'$, with respect to a matching m, is defined as follows:

$$
\begin{aligned}
G \sqcap_m G' \doteq & \{(v,l) \in r_V | \exists v' \in m(v), (v',l) \in r_{V'}\} \\
& \cup \{(v',l) \in r_{V'} | \exists v \in m(v'), (v,l) \in r_V\} \\
& \cup \{(v_i,v_j,l) \in r_E | \exists (v'_i,v'_j) \in m(v_i,v_j), (v'_i,v'_j,l) \in r_{E'}\} \\
& \cup \{(v'_i,v'_j,l) \in r_{E'} | \exists (v_i,v_j) \in m(v'_i,v'_j), (v_i,v_j,l) \in r_E\}
\end{aligned}
$$

Given a multivalent matching m, we also have to identify the set of split vertices, i.e., the set of vertices that are matched with more than one vertex, each split vertex v being associated with the set s_v of its mapped vertices:

$$splits(m) = \{(v, m(v)) | v \in V \cup V', |m(v)| \geq 2\}$$

The *similarity* of G and G' with respect to a mapping m is then defined by:

$$sim_m(G,G') = \frac{f(G \sqcap_m G') - g(splits(m))}{f(r_V \cup r_E \cup r_{V'} \cup r_{E'})} \tag{3}$$

where f and g are two functions that are introduced to weight features and splits, depending on the considered application.

Finally, the *absolute similarity* $sim(G,G')$ of two graphs G and G' is the highest similarity with respect to all possible mappings:

$$sim(G,G') = \max_{m \subseteq V \times V'} \frac{f(G \sqcap_m G') - g(splits(m))}{f(r_V \cup r_E \cup r_{V'} \cup r_{E'})} \tag{4}$$

5.2 Our Graph Distance Measure and the Graph Similarity of Champin and Solnon

Both our graph distance measure and the graph similarity of Champin and Solnon have been shown to be generic in the sense that they can be used to model many other graph distance/similarity measures from the literature. We show here that these two measures have the same ability to represent graph matching problems.

Theorem 10. *Given two sets of vertex and edge labels L_V and L_E and two functions f and g that define a graph similarity measure, there exists a distance measure $\delta =< \delta_{vertex}, \delta_{edge}, \otimes >$ such that for any pair of labeled graphs $G_1 = \langle V_1, r_{V1}, r_{E1} \rangle$ and $G_2 = \langle V_2, r_{V2}, r_{E2} \rangle$ defined over L_V and L_E, the matching $m \subseteq V_1 \times V_2$ that maximizes $sim_m(G_1,G_2)$ also minimizes $\delta_m(G'_1,G'_2)$ where G'_1 and G'_2 are the nonlabeled graphs corresponding to G_1 and G_2, i.e., $G'_1 = (V_1,E_1)$ and $G'_2 = (V_2,E_2)$ with $E_1 = \{(u,v)/\exists (u,v,l) \in r_{E1}\}$ and $E_2 = \{(u,v)/\exists (u,v,l) \in r_{E2}\}$.*

Proof. In order to make the proof, we show that it is possible to define the distance functions δ_{vertex} and δ_{edge} in such a way that the arguments of the function $\otimes$ contains all the information required to reconstitute the matching done. As a consequence, the function $\otimes$ can be defined with the functions f and g.

Let us define a bijective function $num : \wp(V_2) \rightarrow N$ that associates an unique integer value with every different subset of vertices of G'_2. The function num is used by the vertex distance function δ_{vertex} to return the set of vertices of G'_2 matched with each vertex of G'_1:

$$\begin{aligned}\forall v \in V_1, \forall s_v \subseteq V_2, \delta_{vertex}(v, s_v) = num(s_v)\\ \forall v \in V_2, \forall s_v \subseteq V_1, \delta_{vertex}(v, s_v) = 0\\ \forall (u, v) \in E_1, \forall s_e \subseteq E_2, \delta_{edge}((u, v), s_e) = 0\\ \forall (u', v') \in E_2, \forall s_e \subseteq E_1, \delta_{edge}((u', v'), s_e) = 0\end{aligned}$$

With such vertex and edge distance functions, the function $\otimes_{sim}$ can be defined with the functions f and g of the similarity measure:[3]

$$\otimes_{sim}(S) = g(split(m_s)) - f(G_1 \sqcap_{m_S} G_2)$$

where m_S is defined as follows:

$$m_S = \{(u, u')/\exists (u, d) \in S \wedge u \in V_1 \wedge u' \in num^{-1}(d)\}$$

Theorem 11. *Given a distance definition $\delta =< \delta_{vertex}, \delta_{edge}, \otimes >$, there exists a graph similarity measure sim of Champin and Solnon (defined by the two functions f and g) such that for any pair of graphs G_1 and G_2, the matching $m \subseteq V_1 \times V_2$ that minimizes the distance $\delta_m(G_1, G_2)$ also maximize $sim_m(G'_1, G'_2)$, where G'_1 and G'_2 are two labeled graphs corresponding to G_1 and G_2.*

Proof. In order to make the proof, we show that, by properly choosing the multi-labeled graphs G_1 and G_2 to compare, the set $G_1 \sqcap_m G_2$ can contain all the information required to know the matching m done. As a consequence, the function f that takes this set as parameter can be defined with the functions δ_{vertex}, δ_{edge}, and $\otimes$ of the graph distance measure.

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we define the multilabeled graphs $G'_1 = \langle V_1, r_{V1}, r_{V2} \rangle$ and $G'_2 = \langle V_2, r_{V2}, r_{E2} \rangle$ and the sets L_V and L_E of vertex and edge labels such that:

$$\begin{aligned}L_V = \{(u, v), u \in V_1, v \in V_2\}\ ,\ L_E = \{l_e\}\\ r_{V1} = \{(u, (u, v)), u \in V_1, v \in V_2\}\ ,\ r_{E1} = \{(u, v, l_e), (u, v) \in E_1\}\\ r_{V2} = \{(v, (u, v)), u \in V_1, v \in V_2\}\ ,\ r_{E2} = \{(u, v, l_e), (u, v) \in E_2\}\end{aligned}$$

With such labeled graphs, the function f can be defined with the functions δ_{vertex}, δ_{edge} and $\otimes$:

$$\begin{aligned}f(S) = - \otimes (\{(v, \delta_{vertex}(v, m_S(v)))/v \in V_1 \cup V_2\}\\ \cup \{((u, v), \delta_{edge}((u, v), m_S(u, v)))/(u, v) \in E_1 \cup E_2\})\end{aligned}$$

where the matching m_S is defined by:

$$m_S = \{(u, v)/\exists (u, (u, v)) \in S\}$$

[3] Note that in one case the problem is to minimize the distance and in the other case, the problem is to maximize the similarity. So, the function $\otimes$ must be defined in such a way that $\forall m \subseteq V_1 \times V_2, \delta_m(G_1, G_2) = -sim_m(G_1, G_2)$.

6 Computing the Distance Between two Graphs

All matching problems described in Sect. 4 are NP-complete or NP-hard problems, except for the graph isomorphism problem, the complexity of which is not exactly stated.[4] As a consequence, computing the distance between two graphs is also a NP-hard problem in the general case.

Complete algorithms have been proposed for computing the matching which maximizes the similarity of Champin and Solnon [6] and for computing the extended graph edit distance of Ambauen et al. [2]. This kind of algorithms based on an exhaustive exploration of the search space combined with pruning techniques, guarantees solution optimality. However, these algorithms are limited to very small graphs. Therefore, incomplete algorithms, that do not guarantee optimality but have a polynomial time complexity, appear to be good alternatives. We propose in [6, 15, 20, 21] three incomplete algorithms for computing the similarity of Champin and Solnon. These algorithms may be adapted to our graph distance in a very straightforward way.

Greedy Algorithm

We propose in [6] a greedy algorithm. The algorithm starts from an empty matching $m = \emptyset$, and iteratively adds to m couples of vertices chosen within the set of candidate couples $cand = V \times V' - m$. This greedy addition of couples to m is iterated until m is locally optimal, i.e., until no more couple addition can increase the similarity. At each step, the couple to be added is randomly chosen within the set of couples that most increase the similarity. This greedy algorithm has a polynomial time complexity of $\mathcal{O}((|V| \times |V'|)^2)$, provided that the computation of the f and g functions have linear time complexities with respect to the size of the matching.

Reactive Tabu Search

The greedy algorithm of [6] returns a "locally optimal" matching in the sense that adding or removing one couple of vertices to this matching cannot improve it. However, it may be possible to improve it by adding and/or removing more than one couple to this matching. In order to improve the matching returned by the greedy algorithm, we propose in [6, 15] a reactive tabu local search.

A local search [25, 26] tries to improve a solution by locally exploring its neighborhood: the neighbors of a matching m are the matchings which can be obtained by adding or removing one couple of vertices to m.

From an initial matching, computed by the greedy algorithm, the search space is explored from neighbor to neighbor until the optimal solution is found (when the optimal value is known) or until a maximum number of moves have been performed.

[4] For particular graphs (such as trees or planar graphs) the graph isomorphism problem is polynomial [22–24]; in general case, the graph isomorphism problem clearly belongs to NP but has neither be proven to belong in P nor to be NP-complete.

The tabu metaheuristic [25, 27] is used to choose the next neighbor to move on. At each step, the best neighbor, i.e., the one that most increase the similarity, is chosen. To avoid staying around locally optimal matchings by always performing the same moves, a tabu list is used. This list has a length k and memorizes the last k moves (i.e., the last k added/removed couples of vertices) in order to forbid backward moves (i.e., to remove/add a couple recently added/removed).

The length k of the tabu list is a critical parameter that is hard to set: if the list is too long, search diversification is too strong so that the algorithm converges too slowly; if the list is too short, intensification is too strong so that the algorithm may be stuck around local maxima and fail in improving the current solution. To solve this parameter tuning problem, Battiti and Protasi [28] introduced *Reactive Search* where the length of the tabu list is dynamically adapted during the search. We have used the same idea to build a reactive tabu search algorithm to compute our generic graph distance measure.

Ant Colony Optimization

We also proposed in [20,21] to use the Ant Colony Optimization (ACO) metaheuristic approach to compute the similarity of Champin and Solnon. The ACO metaheuristic is a bioinspired approach [29, 30] that has been used to solve many hard combinatorial optimization problems. The main idea is to model the problem to solve as a search for an optimal path in a graph – called the construction graph – and to use artificial ants to search for "good" paths.

The behavior of artificial ants mimics the behavior of real ones: (1) ants lay pheromone trails on the components of the construction graph to keep track of the most promising components, (2) ants construct solutions by moving through the construction graph and choose their path with respect to probabilities which depend on the pheromone trails previously laid, and (3) pheromone trails decrease at each cycle simulating in this way the evaporation phenomena observed in the real world.

Given two graphs $G = (V, E)$ and $G' = (V', E')$ to match, the construction graph is the complete nondirected graph that associates a vertex $< (u, u') >$ with each couple $(u, u') \in V \times V'$. Each elementary path into this graph represents a matching $m \subseteq V \times V'$.

At each cycle, each ant of a colony constructs a matching in a randomized greedy way: starting from an empty matching $m = \emptyset$, the ant iteratively adds couples of vertices that are chosen within the set $cand = \{(u, u') \in V \times V' - m\}$. As usually in ACO algorithm, the choice of the next couple to be added to m is done with respect to a probability that depends on pheromone and heuristic factors (i.e., the similarity improvement when adding the couple). A simple local search procedure may be applied on built matchings to improve their quality.

Once each ant of the colony has built a matching, pheromone trails are updated according to the best matching found. Pheromone is laid on each vertex $< (u, u') >$ of the best found matching in a quantity proportional to the similarity induced by the matching. As a consequence, the amount of pheromone on a vertex $< (u, u') >$ represents the learnt desirability to match u with u'. This process stops iterating

either when an ant has found an optimal matching, or when a maximum number of cycles has been performed.

Experimental Results

These three algorithms have been experimentally compared on three different test suites: graph and subgraph isomorphism problems, randomly generated multivalent problems, and the nonbijective graph matching problems of Boeres et al. [4]. Each of these problems has been transformed into our generic graph similarity measure computing problem and, as a consequence, we always use exactly the same code whatever the problem to solve is.

Experimental results showed us that on graph and subgraph isomorphism problems, our algorithms are not competitive with dedicated algorithms: our reactive tabu search and ACO algorithms are able to solve these problems but are clearly longer than dedicated algorithms such as Nauty [31] or VFLIB [32, 33]. These results can be explained by the fact that our algorithms do not use any kind of filtering techniques and potentially explore all kinds of mappings, even multivalent ones. On the seven instances of the nonbijective graph matching problem, our algorithms obtain better results than $LS+$, the reference algorithm of [4] (six instances over seven are better solved by reactive tabu search and seven instances over seven are better solved by ACO algorithm). On all these instances, ACO obtains better results than reactive tabu search but reactive tabu search finds the solutions in shorter times than ACO. On multivalent graph matching problems, reactive tabu search and ACO obtain similar results. However, reactive tabu search finds the solutions in shorter times than ACO.

As a consequence, ACO usually obtains better results but is slower than reactive tabu search. These two algorithms are complementary: if we have to quickly compute a "good" solution of hard instances or if instances are easy, we can use reactive tabu search but if we have more time to spend on computation or if we want to solve very hard instances, we can use ACO.

7 Conclusion

In this chapter, we propose a graph distance measure. This distance is generic: it is based on multivalent matchings of the graph vertices and it is parameterized by two distance functions δ_{vertex} and δ_{edge} used to introduce the application-dependent distance knowledge on vertices and edges and a function $\otimes$ used to aggregate these local preferences. We have shown that we can use our graph distance measure to solve many graph matching problems including the problem of computing the generic graph similarity of Champin and Solnon. We quickly describe three algorithms to compute this generic distance measure: a greedy algorithm, a reactive tabu local search, and an Ant Colony Optimization algorithm. These algorithms are generic so that they can be used to solve any kind of graph matching problem.

References

1. H. Bunke. Graph matching: Theoretical foundations, algorithms, and applications. In *Proceedings on Vision Interface 2000*, Montreal, pages 82–88, 2000
2. R. Ambauen, S. Fischer, and H. Bunke. Graph edit distance with node splitting and merging, and its application to diatom identification. In E. Hancock and M. Vento, editors, *IAPR-TC15 Wksp on Graph-based Representation in Pattern Recognition*, volume 2726 of *LNCS*, Springer, Berlin Heidelberg New York, pages 95–106, 2003
3. R. Baeza-Yates and G. Valiente. An image similarity measure based on graph matching. In *Proceedings of 7th International Symposium on String Processing and Information Retrieval*, pages 28–38. IEEE Computer Science Press, 2000
4. M. Boeres, C. Ribeiro, and I. Bloch. A randomized heuristic for scene recognition by graph matching. In *WEA 2004*, pages 100–113, 2004
5. A. Hlaoui and S. Wang. A new algorithm for graph matching with application to content-based image retrieval. *LNCS*, volume 2396, 2002
6. P.-A. Champin and C. Solnon. Measuring the similarity of labeled graphs. In *5th International Conference on Case-Based Reasoning (ICCBR 2003)*, volume Lecture Notes in Artificial Intelligence 2689-Springer, Berlin Heidelberg New York, pages 80–95, 2003
7. T. Akutsu. Protein structure alignment using a graph matching technique, citeseer.nj. nec.com/akutsu95protein.html, 1995
8. L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–602, 1996
9. A. Schenker, M. Last, H. Bunke, and A. Kandel. Classification of web documents using graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3): 475–496, 2004
10. H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18: 689–694, 1997
11. D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3): 265–298, 2004
12. A. Deruyver, Y. Hod, E. Leammer, and J.-M. Jolion. Adaptive pyramid and semantic graph: Knowledge driven segmentation. In L. Brun and M. Vento, editors, *Graph-Based Representations in Pattern Recognition: 5th IAPR International Workshop, GbRPR 2005, Poitiers, France, April 11–13, 2005. Proceedings*, volume 3434 of *LNCS*, page 213. Springer, Berlin Heidelberg New York, 2005
13. H. Bunke. Recent developments in graph matching. In *ICPR 2000*, pages 2117–2124, 2000
14. J.M. Jolion. Graph matching: what are we really talking about? In *3rd IAPR-TC15 workshop on Graph-based Representations in Pattern Recognition*, pages 170–175, 2001
15. S. Sorlin and C. Solnon. Reactive tabu search for measuring graph similarity. In L. Brun and M. Vento, editors, *5th IAPR-TC-15 workshop on Graph-based Representation in Pattern Recognition*, Springer, Berlin Heidelberg New York, pages 172–182, 2005
16. S. Zampelli, Y. Deville, and P. Dupont. Approximate constrained subgraph matching. In *11th International Conference on Principles and Practice of Constraint Programming*, number 3709 in LNCS, Springer, Berlin Heidelberg new York, pages 832–836, 2005
17. D. Lin. An Information-theoretic definition of similarity. In *Proceedings of ICML 1998, 15th International Conference on Machine Learning*, Morgan Kaufmann, Los Altos, CA, pages 296–304, 1998
18. A. Tversky. Features of Similarity. In *Psychological Review*, volume 84, American Psychological Association Inc., pages 327–352, 1977

19. H. Bunke and X. Jiang. *Graph matching and similarity*, In H.-N. Teodorescu, D. Mlynek, A. Kandel, and H.-J. Zimmermann, editors, *Intelligent Systems and Interfaces*, Chapter 1. Kluwer, Dordrecht, 2000
20. O. Sammoud, C. Solnon, and K. Ghdira. Ant algorithm for the graph matching problem. In *5th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2005)*, volume 3448 of *LNCS*, Springer, Berlin Heidelberg New York, pages 213–223, April 2005
21. O. Sammoud, S. Sorlin, C. Solnon, and K. Ghdira. A comparative study of ant colony optimization and reactive search for graph matching problems. In *6th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2006)*, volume to appear of *LNCS*. Springer, Berlin Heidelberg New York, April 2006
22. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison Wesley, Reading, MA, USA, 1974
23. J.E. Hopcroft and J.-K. Wong. Linear time algorithm for isomorphism of planar graphs. 6^{th} *Annual ACM Symposium on Theory of Computing*, pages 172–184, 1974
24. E.M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer System Science*, pages 42–65, 1982
25. F. Glover. Tabu search – part I. *Journal on Computing*, pages 190–260, 1989
26. S. Kirkpatrick, S. Gelatt, and M. Vecchi. Optimisation by simulated annealing. In *Science*, volume 220, pages 671–680, 1983
27. S. Petrovic, G. Kendall, and Y. Yang. A Tabu Search Approach for Graph-Structured Case Retrieval. In *STAIRS 2002*, pages 55–64, 2002
28. R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. In Springer, editor, *Algorithmica*, volume 29, pages 610–637, 2001
29. M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization.* McGraw Hill, London, UK, pages 11–32, 1999
30. M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT, Cambridge, MA, 2004
31. B.D. McKay. Practical graph isomorphism. *Congressus Numerantium*, Nauty, 1981
32. L.P. Cordella, P. Foggia, and M. Vento C. Sansone. An improved algorithm for matching large graphs. *3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, 2001
33. L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. Performance evaluation of the vf graph matching algorithm. In *Proceedings of the 10th International Conference on Image Analysis and Processing (ICIAP'99)*, IEEE, New York, page 1172, 1999