A Generic Graph Distance Measure Based on Multivalent Matchings

Sébastien Sorlin, Christine Solnon and Jean-Michel Jolion

LIRIS, CNRS UMR5205, bât. Nautibus, University of Lyon I
43 Bd du 11 novembre, 69622 Villeurbanne cedex, France
{sebastien.sorlin,christine.solnon,jean-michel.jolion}@liris.cnrs.fr

Abstract. Many applications such as *e.g.*, information retrieval and classification, involve measuring graphs 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. Exact graph matchings such as graph or subgraph isomorphism can be used in order to show graph equivalence or inclusion. However, in many applications, the assumption of the existence of such an "exact" matching is too strong. As a consequence, error-tolerant graph matchings such as maximum common subgraph and graph edit distance have been proposed. Such matchings drop the condition that the matching must preserve all vertices and edges and look for a "best" matching, *i.e.*, one which preserves a maximum number of vertices and edges. Most recently, three different approaches proposed to go one step further by introducing multivalent matchings where a vertex may be matched with a set of vertices. This kind of matching handles the fact that, due to different description granularity levels, one object component may "play the same role" than a set of components of another object.
A first goal of this paper 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. A second goal of this paper 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 non bijective graph matching.

1 Introduction

In many applications such as information retrieval or classification, measuring object similarity is an important issue [8]. 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 [3, 6, 17, 4], design objects [11], molecules representations [2, 18], web documents [27]. Vertices represent object components while edges represent binary relations between these components. Vertices and edges may be labelled 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 to each couple of vertices corresponding to two adjacent regions. In order to better represent images, each region (*i.e.*, each vertex) may be labelled by its size and its bounding box and each edge may be labelled by a value representing how much two regions are connected (by means of the number of adjacent pixels) [3].

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 matching considered.

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:

- 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
- subgraph isomorphism, that involves finding an injection between 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:

- maximum common subgraph [7, 12] that looks for the largest matching (with respect to the number of matched vertices) that preserves all the edges of the matched vertices
- graph edit distance [7, 12] that looks for the minimum cost set of operations (*i.e.*, vertex and edge insertion, deletion and relabelling) 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. Four recent papers proposed

graph distance/similarity measures based on multivalent error-tolerant graph matchings:

- Champin and Solnon [11] measure the similarity of design objects where one single component of an object may play the same role than a set of components of another object, depending on the granularity of object description. Therefore, the graph similarity measure is based on multivalent matchings so that one vertex in a graph may be associated with a set of vertices of the other graph.
- Boeres and al. [6] use graph matching for model-based pattern recognition of brain images. 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 to a same vertex of the model. Deruyver and al. [13] use graph matching for image segmentation: the vertices of a graph that represents an initial over-segmented image are merged until the resulting graph matchs another graph that semantically describes the image. As a consequence, as in [6], the graph matching is multivalent.
- Ambauen and al. [3] 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

A wide number of graph distance and similarity measures have been proposed in the litterature [20, 10]. These measures are based on different definitions of a "best" matching between two graphs depending on the considered application.

The graph similarity measure of [6] is specific to the addressed problem: it is used for matching brain images to models, and in this context they added specific constraints (*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 for another application.

Ambauen and al. defines [3] a more generic graph similarity 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 [6], this measure adds an image recognition specific constraint on the considered multivalent matching. The multivalent matching operations (vertex merging and splitting) need to be non-overlapping: if we want to link two vertices u and v of one graph to another vertex u', we need 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 where we need to merge over-segmented region, it is not a desirable property in all applications (in particular for the application of [11]). Finally, the measure introduced in [3] is not generic enough

to express all kinds of multivalent matching problems: for example, it cannot be used to model the problem described in [6].

In [28] it has been proven that the similarity measure of Champin and Solnon [11] is generic in the sense that, thanks to two similarity functions, it can be used to compute many other similarity measures (including measures of Boeres and al. [6] and Ambauen and al. [3]). However, if it has been proven generic, it is not always straightforward to use. The measure of Champin and Solnon [11] deals with multi-labelled graphs and the similarity of two multi-labelled graphs is computed with respect to the set of the common labels identified by a mapping. As a consequence, the comparison of the graph components is a binary operation: a label is a discrete value so that the label is recovered or not. However, in many applications and in particular in an image recognition context, one needs to represent and 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 needs to compute the difference between their sizes. Furthermore, when two components are merged, one needs to have an operator to aggregate these continuous values. For example, one needs to compute the sum of the sizes or the average color of a set of merged regions. Finally, some constraints on the allowed matchings are difficult to express in [11]. For example, it is difficult to constrain a vertex to be only linked to vertices having a particular property. To express these kinds of constraints on matchings, we show in [28] that one can label the graph vertices in such a way that one can reconstitute the original matching from the set of recovered labels. As a consequence, the similarity of [11] 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 matching 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 [11]. However, it has been designed to be more flexible: this distance is based on a multivalent matching of the graph vertices like in [11] but it is parameterized by vertex and edge distance functions that can easily express many different vertex and edge properties (such as labels, real values...).

In section 2, we introduce some definitions and notations needed to define our distance measure. In section 3, we propose a new generic graph distance measure. In section 4, we compare our graph distance regarding some classical graph matching problems. In section 5, we prove that our distance and the graph similarity measure of Champin and Solnon [11] are equivalent in the sense that they have the same power of expression. We conclude in section 6 with some computational issues.

2 Definitions and notations

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

- V is a finite set of *vertices*
- $E \subseteq V \times V$ is a set of oriented pairs 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, induced subgraph. A graph $G' = (V', E')$ is a *partial subgraph* of a graph $G = (V, E)$ (noted $G' \subseteq G$) 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 $G' \sqsubseteq G$) 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.

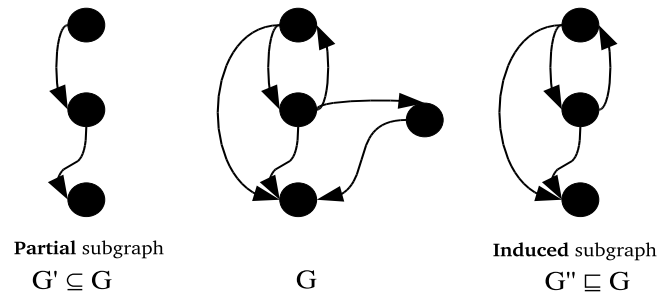

Fig. 1. Example of a graph G, a partial subgraph G' of G and an induced subgraph G'' of G

Graphs 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 to 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 is element of $m(v)$.

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 matching*. Given two graphs $G = (V, E)$ and $G' = (V', E')$ and a matching $m \subseteq V \times V'$, m is said to be:

- a *partial function* from G to G' if m links each vertex of V to at most one vertex of G', *i.e.*:
$$\forall v \in V \ , |m(v)| \leq 1$$
- a *total function* from G to G' if m links each vertex of V to exactly one vertex of G', *i.e.*:
$$\forall v \in V \ , |m(v)| = 1$$
- a *univalent matching* between G and G' if m links each vertex of V and V' to at most one vertex, *i.e.*:
$$\forall v \in V \cup V' \ , |m(v)| \leq 1$$
- an *injective matching* from G to G' if m links each vertex of V to a different vertex of V', *i.e.*:
$$\forall (u, v) \in V \times V, |m(u)| = |m(v)| = 1 \wedge u \neq v \Rightarrow m(u) \neq m(v)$$
Another definition of an injective matching from G to G' is a matching m such that:
$$\forall v \in V \ , |m(v)| = 1$$
$$\forall v \in V' \ , |m(v)| \leq 1$$
- 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.*:
$$\forall (u, v) \in (V \times V) \cup (V' \times V') \ , |m(u)| = |m(v)| = 1 \wedge u \neq v \Rightarrow m(u) \neq m(v)$$
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.*:
$$\forall v \in V \cup V' \ , |m(v)| = 1$$

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 to 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 to 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 to at least one vertex or edge of G' (resp. G), *i.e.*:

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

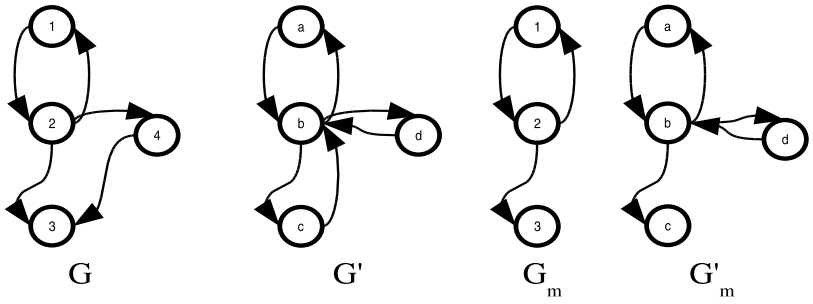

Fig. 2. Two graphs G and G' and their subgraphs induced by the matching $m = \{(1, a), (1, d), (2, b), (3, c)\}$

3 A new graph distance measure

In this section, we introduce a new generic graph distance measure. This measure deals with graphs that may have properties or not. It is tuned by vertex and edge distance functions expressing local preferences on vertex and edge matchings.

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 matching, *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, we need 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_v (resp. δ_e) giving 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 number from the intervall $[0, +\infty[$ 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:

$$\delta_v : (V, \wp(V')) \cup (V', \wp(V)) \rightarrow [0, +\infty[$$
$$\delta_e : (E, \wp(E')) \cup (E', \wp(E)) \rightarrow [0, +\infty[$$

Generally, the distance will be 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 the matching violates a hard constraint. 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).

Roughly speaking, the function δ_v (resp. δ_e) expresses the *local preferences* on the way to match a vertex (resp. an edge). The vertex and edge distance functions depend on the considered application and are used to reflect both the similarity knowledge and some of the constraints that a matching must respect.

For example, if we are looking for a 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_v and δ_e as follows:

$$\begin{aligned}\forall v \in V \cup V', \forall s_v \in \wp(V) \cup \wp(V'), \delta_v(v, s_v) &= 1 - |s_v| \text{ if } |s_v| \leq 1\\ &\quad +\infty \text{ otherwise}\\ \forall (u,v) \in E \cup E', \forall s_e \in \wp(E) \cup \wp(E'), \delta_e((u,v), s_e) &= 1 - |s_e| \text{ if } |s_e| \leq 1\\ &\quad +\infty \text{ otherwise}\end{aligned}$$

3.2 Graph distance

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

$$\begin{aligned}\delta_m(G, G') = \otimes(\{(v, \delta_v(v, m(v)))/v \in V \cup V'\} \cup \\ \{((u,v), \delta_e((u,v), m(u,v)))/(u,v) \in E \cup E'\})\end{aligned} \quad (1)$$

where $\otimes$ is an application-dependant function which is used to aggregate the computed 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 matching. 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 $\otimes$ could be defined to make the distance between two graphs depending 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 graphs vertices. Now, we can 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:

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

Finally, given two graphs G and G', a distance measure between G and G' is defined as a triple $\delta =< \delta_v, \delta_e, \otimes >$ where δ_v is the vertex distance function, δ_e the edge distance function and $\otimes$ is a function used to aggregate the distances of all vertices and edges of both 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 chosen definitions of δ_v, δ_e 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 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 defined by the function $\otimes_\sum$ that simply makes the sum of the distances. 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 matching

In this subsection we show how to solve exact graph matching problems. For all these kinds of problems, we are looking for a univalent matching between the

vertices of two graphs. As a consequence, the vertex and edge similarity functions are defines in such a way that a multivalent matching always involves an infinite positive distance. Furthermore, these problems being 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 having the same number of vertices, the graph isomorphism problem consists in proving that 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]. The graph isomorphism problem is used to prove that two graphs are structurally identical.

Measure definition. To solve the graph isomorphism problem using our distance measure, we need to choose vertex and edge distance functions such that these functions returns 0 if the vertex or edge is matched to exactly one element and $+\infty$ otherwise (in order to avoid non bijective matchings). More formally:

$$\forall v \in V \cup V', \forall s_v \subseteq V \cup V', \quad \delta_v^{iso}(v, s_v) = \begin{array}{l} 0 \text{ if } |s_v| = 1 \\ +\infty \text{ otherwise} \end{array}$$

$$\forall (u, v) \in E \cup E', \forall s_e \subseteq E \cup E', \delta_e^{iso}(u, v, s_e) = \begin{array}{l} 0 \text{ if } |s_e| = 1 \\ +\infty \text{ otherwise} \end{array}$$

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. *the distance $\delta^{iso} =< \delta_v^{iso}, \delta_e^{iso}, \otimes_{\sum} >$ between G and G' is equal to 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, given the definition of δ_v^{iso} and δ_e^{iso}, the distance of G and G' with respect to m is equal to 0 and the distance between G and G' is equal to 0.

(2) $\Rightarrow$ (1). If the distance between G and G' is equal to 0, then, given the definition of δ_v^{iso}, there exists a matching m such that $\forall v \in V \cup V', |m(v)| = 1$. As a consequence, the matching m is a bijective matching. Furthermore, if m involves a distance equal to 0, then, $\forall (u, v) \in E \cup E', |m(u, v)| = 1$. As a

[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 shall note $m(v)$ to denote the single vertex that is element of $m(v)$.

consequence, each edge of both the graphs is matched to 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

Problem definition. Given two graphs $G = (V,E)$ and $G' = (V',E')$ such that $|V| \leq |V'|$, the partial subgraph isomorphism problem consists in showing that 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 prove that a graph is included into another graph.

Measure definition. To solve the partial subgraph isomorphism problem using our distance measure, we need to choose vertex and edge distance functions such that these functions return 0 if an element of G is matched to one element (in order to preserve the vertices and the edges of G) and $+\infty$ otherwise (in order to avoid non injective matching). Distance functions for vertices and edges of G' just need to avoid non univalent matching. More formally:

$$G \begin{cases} \forall v \in V, \forall s_v \subseteq V', \delta_v^{psub}(v,s_v) = 0 \text{ if } |s_v| = 1 \\ \qquad\qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \\ \forall (u,v) \in E, \forall s_e \subseteq E', \delta_e^{psub}(u,v,s_e) = 0 \text{ if } |s_e| = 1 \\ \qquad\qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \end{cases}$$

$$G' \begin{cases} \forall v \in V', \forall s_v \subseteq V, \delta_v^{psub}(v,s_v) = 0 \text{ if } |s_v| \leq 1 \\ \qquad\qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \\ \forall (u,v) \in E', \forall s_e \subseteq E, \delta_e^{psub}(u,v,s_e) = 0 \text{ if } |s_e| \leq 1 \\ \qquad\qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \end{cases}$$

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. *the distance $\delta^{psub} =< \delta_v^{psub}, \delta_e^{psub}, \otimes_{\sum} >$ between G and G' is equal to 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_v^{psub} and δ_e^{psub}, the similarity of G and G' with respect to m is equal to 0 and the distance between G and G' is equal to 0.

(2) $\Rightarrow$ (1). If the distance between G and G' is equal to 0, then, given the definition of δ_v^{psub}, there exists a matching m such that $\forall v \in V, |m(v)| = 1$ 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 to 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 showing that 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 to an edge of G'). To check these constraints, the edge distance function δ_e needs 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 needs 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 (1, 3 and 4) and 0 otherwise. The edge distance function must return $+\infty$ if an edge of G is not matched (2.1) or if a couple (u,v) of vertices of G which is not an edge is matched to an edge of G' (2.2) and 0 otherwise. More formally, given a graph $G = (V, E)$ and a graph $G' = (V', E')$, we need to compare the graphs $G'' = (V, V \times V)$ and G' with the two following distance functions:

$$
G'' \begin{cases}
1 & \forall v \in V, \forall s_v \subseteq V', \delta_v^{sub}(v, s_v) = 0 \text{ if } |s_v| = 1 \\
 & \qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \\
2.1 & \forall (u,v) \in V^2, \forall s_e \subseteq E', \delta_{eG}^{sub}(u, v, s_e) = 0 \text{ if } (u,v) \in E \wedge |s_e| = 1 \\
2.2 & \qquad\qquad\qquad\qquad\quad = 0 \text{ if } (u,v) \notin E \wedge s_e = \emptyset \\
 & \qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise}
\end{cases}
$$

$$
G' \begin{cases}
3 & \forall v \in V', \forall s_v \subseteq V, \delta_v^{sub}(v, s_v) = 0 \text{ if } |s_v| \leq 1 \\
 & \qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise} \\
4 & \forall (u,v) \in E', \forall s_e \subseteq E, \delta_{eG}^{sub}(u, v, s_e) = 0 \text{ if } |s_e| \leq 1 \\
 & \qquad\qquad\qquad\qquad\quad +\infty \text{ otherwise}
\end{cases}
$$

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. *the distance* $\delta_G^{sub} =< \delta_v^{sub}, \delta_{eG}^{sub}, \otimes_{\sum} >$ *between* $G'' = (V, V \times V)$ *and* G' *is equal to* 0

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_v^{sub} and δ_{eG}^{sub}, the distance of G'' and G' with respect to m is equal to 0 and the distance between G'' and G' is equal to 0.

(2) $\Rightarrow$ (1). If the distance between G'' and G' is equal to 0, then, given the definition of δ_v^{sub}, there exists a matching m such that $\forall v \in V, |m(v)| = 1$ 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 to 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'

Approximate graph matching

Problem definition. Zampelli and al. propose [30] a problem named "approximate subgraph matching" that consists in looking for a pattern graph into a target graph and that 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 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 a univalent matching[3] $m \subseteq V_p \times V_t$ such that:

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

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

[3] In [30], an approximate subgraph matching is defined as a function $f : V_p \rightarrow V_t$ but for a reason of homogeneity, we define it as a univalent matching

Measure definition. Solving an approximate subgraph matching problem consists in finding a univalent matching m between G_p and the graph $G' = (V_t, V_t \times V_t)$ such that each mandatory vertex is matched to exactly one vertex (1), each optional vertex is matched to at most one vertex (2), each edge (u, v) is either matched to a couple of vertices (u', v') of G' which is an edge of G_t (3.2) or is not matched at all if one of its optional endpoints is not matched (3.1), each forbidden edge is not matched (4). Finally, the matching must be univalent (5 and 6). More formally, one needs 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:

$$
\begin{array}{ll}
G & \begin{cases}
1 & \forall v \in V_p - O_p, \forall s_v \subseteq V_t, \delta_v^{agm}(v, s_v) = 0 \text{ if } |s_v| = 1 \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise} \\
2 & \forall v \in O_p, \forall s_v \subseteq V_t, \delta_v^{agm}(v, s_v) = 0 \text{ if } |s_v| \leq 1 \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise} \\
3.1 & \forall (u, v) \in E_p, \forall s_e \subseteq V_t \times V_t, \delta_{eG_t}^{agm}(u, v, s_e) = 0 \text{ if } s_e = \emptyset \\
3.2 & \qquad\qquad\qquad\qquad 0 \text{ if } s_e = \{(u', v')\} \wedge (u', v') \in E_t \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise} \\
4 & \forall (u, v) \in F_p, \forall s_e \subseteq E', \delta_{eG_t}^{agm}(u, v, s_e) = 0 \text{ if } s_e = \{(u', v')\} \wedge (u', v') \notin E_t \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise}
\end{cases} \\
G' & \begin{cases}
5 & \forall v \in V_t, \forall s_v \subseteq V_p, \delta_v^{agm}(v, s_v) = 0 \text{ if } |s_v| \leq 1 \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise} \\
6 & \forall (u, v) \in E', \forall s_e \subseteq E_p \cup F_p, \delta_{eG_t}^{agm}(u, v, s_e) = 0 \text{ if } |s_e| \leq 1 \\
 & \qquad\qquad\qquad\qquad +\infty \text{ otherwise}
\end{cases}
\end{array}
$$

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. *the distance $\delta_{G_t}^{agm} =< \delta_v^{agm}, \delta_{eG_t}^{agm}, \otimes_{\sum} >$ induced by the matching m between the graphs $G = (V_p, E_p \cup F_p)$ and $G' = (V_t, V_t \times V_t)$ is equal to 0*

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 2), $\forall v \in V_t, |m(v)| \leq 1$ and $\forall (u, v) \in V_t \times V_t, |m(u, v)| \leq 1$ (condition 1), $\forall (u, v) \in E_p, m(u, v) = \{(u', v')\} \wedge (u', v') \in E_t$ (condition 3) and $\forall (u, v) \in F_p, m(u, v) = \{(u', v')\} \wedge (u', v') \notin E_t$ (condition 2 and 4). As a consequence, given the definition of the vertex and edge distance functions, the distance $\delta_{G_t}^{agm} =< \delta_v^{agm}, \delta_{eG_t}^{agm}, \otimes_{\sum} >$ induced by the matching m between the graphs $G = (V_p, E_p \cup F_p)$ and $G' = (V_t, V_t \times V_t)$ is equal to 0.

(2) $\Rightarrow$ (1). If the distance $\delta_{eG_t}^{agm} =< \delta_v^{agm}, \delta_{eG_t}^{agm}, \otimes_{\sum} >$ induced by the matching m between the graphs $G = (V_p, E_p \cup F_p)$ and $G' = (V_t, V_t \times V_t)$ is equal to 0, then the matching m is univalent (condition 1) because, given the vertex

and edge distance functions, all non-univalent matching give rise to an infinite distance. Furthermore, if the distance induced by m is equal to 0, then $\forall v \in V_p - O_p, |m(v)| = 1$ so that m respects the condition 2. 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 the condition 3. 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 the condition 4 and m is a solution of the approximate subgraph matching problem.

4.2 Error tolerant graph matching

In this subsection we show how to model error tolerant graph matching problems as graph distance measures. For all these problems, we are looking for a 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 non univalent matching always gives an infinite positive distance. Furthermore, these problems being optimization problems, the objective is always to find the matching giving 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 $size(a \cap b)/size(a + b)$ [29, 22].

Measure definition. We need to use vertex and edge distance functions forbidding multivalent matching while encouraging vertices and edges of G and G' to be linked. As a consequence, the vertex and edge distance functions must return $+\infty$ if the element is matched to more than one element, 1 if it is not matched and 0 is the element is matched to exactly one element, *i.e.*:

$$\begin{aligned} \forall v \in V \cup V', \forall s_v \subseteq V \cup V', \quad \delta_v^{mcps}(v, s_v) = & \quad 1 - |s_v| \text{ if } |s_v| \leq 1 \\ & \quad +\infty \text{ otherwise} \\ \forall (u,v) \in E \cup E', \forall s_e \subseteq E \cup E', \delta_e^{mcps}(u, v, s_e) = & \quad 1 - |s_e| \text{ if } |s_e| \leq 1 \\ & \quad +\infty \text{ otherwise} \end{aligned}$$

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 $\delta^{mcps} =< \delta_v^{mcps}, \delta_e^{mcps}, \otimes_{\sum} >$*

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 non infinite distance inverse to the size of G'' (and reversely), the property holds.

$\delta_m^{mcps}(G, G') = d < +\infty \Rightarrow G_m$ is a common subgraph of G and G' such that $|G_m| = (|G| + |G'| - d)/2$. Given the vertex and edge distance functions, if $\delta_m^{mcps}(G, G') < +\infty$ then m is a univalent matching (because all non univalent 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 $\delta_m^{mcps}(G, G') = d < +\infty$ 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 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 [7].

Measure definition. To solve the maximum common subgraph problem using our distance measure, we need to use vertex and edge distance functions encouraging vertices of G to be linked while forbidding matching that does 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 (1) 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 $(u, v) \in E \not\Leftrightarrow (u', v') \in E'$ (2) (resp. (4)). Finally, the matching must be univalent (3). More formally, we need 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:

$$G_2 \begin{cases} 1 & \forall v \in V, \forall s_v \subseteq V', \delta_v^{mcs}(v, s_v) = 1 - |s_v| \text{ if } |s_v| \leq 1 \\ & \qquad +\infty \text{ otherwise} \\ 2 & \forall (u, v) \in V^2, \forall s_e \subseteq V'^2, \delta_{eGG'}^{mcs}(u, v, s_e) = 0 \text{ if } s_e = \emptyset \\ & \qquad 0 \text{ if } s_e = \{(u', v')\} \\ & \qquad \wedge((u, v) \in E \Leftrightarrow (u', v') \in E') \\ & \qquad +\infty \text{ otherwise} \end{cases}$$

$$G'_2 \begin{cases} 3 & \forall v \in V', \forall s_v \subseteq V, \delta_v^{mcs}(v, s_v) = 0 \text{ if } |s_v| \leq 1 \\ & \qquad +\infty \text{ otherwise} \\ 4 & \forall (u, v) \in V'^2, \forall s_e \subseteq V^2, \delta_{eGG'}^{mcs}(u, v, s_e) = 0 \text{ if } s_e = \emptyset \\ & \qquad 0 \text{ if } s_e = \{(u', v')\} \wedge \\ & \qquad ((u, v) \in E' \Leftrightarrow (u', v') \in E) \\ & \qquad +\infty \text{ otherwise} \end{cases}$$

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 $\delta^{mcs} =< \delta_v^{mcs}, \delta_e^{mcs}, \otimes_{\sum} >$*
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 of correctness is decomposed into two steps, we first show that, for every matching $m \subseteq V \times V'$ such that $\delta_m^{mcs}(G, G') = d \neq +\infty$, 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 non infinite distance inverse to the size of G'' (and reversely), the property holds.

$\delta^{mcs}_{mGG'}(G_2, G'_2) = d < +\infty \Rightarrow G_m$ is a common induced subgraph of G and G' such that $|G_m| = |G| - d$. Given the vertex and edge distance function, if $\delta_{mGG'}(G_2, G'_2) < +\infty$ then m is a univalent matching (because all non univalent 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 non partial) 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 definition 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 define 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 non-recovered 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 $\delta_{mGG'}(G_2, G'_2) = |G| - |G''|$ 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 $\delta_{mGG'}(G_2, G'_2)$ is equal to $|G| - |G''|$ and that $G_m = G''$.

Graph edit distance (*ged*)

Problem definition. Given two labelled graphs G_1 and G_2 (*i.e.*, graphs where a label is associated to each vertex and each edge), the graph edit distance of G_1 and G_2 is the minimum cost set of weighted operations needed to transform G_1 into G_2. Considered operations are insertions, substitutions (*i.e.*, relabelling), and deletions of vertices and edges. Bunke shows in [7] 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 [9]. A *labelled 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 labelling the vertices of G and $\beta : E \rightarrow L$ is a total function labelling the edges of G. Given two labelled graphs $G_1 = (V_1, E_1, L_1, \alpha_1, \beta_1)$ and $G_2 = (V_2, E_2, L_2, \alpha_2, \beta_2)$, an *error tolerant graph matching* is a univalent matching[4] $m \subseteq V_1 \times V_2$. The vertex u is *substituted* by the vertex v if $m(u) = v$. If $\alpha_1(u) = \alpha_2(m(u))$, the substitution is called an *identical substitution*, otherwise, it is a *non-identical substitution.* Every vertex $v \in V_1$ such that $m(v) = \emptyset$ is *deleted* by m and every vertex

[4] In [9], an error tolerant graph matching is defined as a partial injective function $f : V_1 \rightarrow V_2$ but for a reason of homogeneity, we define it as a univalent matching

$v' \in V_2$ 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 to the non-identical vertex substitutions (resp. insertions and deletions) and a cost c_{es} (resp. c_{ei} and c_{ed}) is associated to the non-identical 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 [9]. 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 non-univalent matchings, the distance between G'_1 and G'_2 with respect to a 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 labelled graphs $G_1 = (V_1, E_1, L_1, \alpha_1, \beta_1)$ and $G_2 = (V_2, E_2, L_2, \alpha_2, \beta_2)$, we need to compare the graphs $G'_1 = (V_1, E_1)$ and $G'_2 = (V_2, E_2)$ with the following vertex and edge distance functions:

$$G'_1 \begin{cases} \forall v \in V_1, \forall s_v \subseteq V_2, \ \delta^{ged}_{vG_1G_2}(v, s_v) = & c_{vd} \text{ if } s_v = \emptyset \\ & 0 \text{ if } s_v = \{v'\} \wedge \alpha_1(v) = \alpha_2(v') \\ & c_{vs} \text{ if } s_v = \{v'\} \wedge \alpha_1(v) \neq \alpha_2(v') \\ & +\infty \text{ if} |s_v| > 1 \\ \forall (u,v) \in E_1, \forall s_e \subseteq E_2, \delta^{ged}_{eG_1G_2}(u, v, s_e) = & c_{ed} \text{ if } s_e = \emptyset \\ & 0 \text{ if } s_e = \{(u', v')\} \wedge \beta_1((u, v)) = \beta_2((u', v')) \\ & c_{es} \text{ if } s_v = \{(u', v')\} \wedge \beta_1((u, v)) \neq \beta_2((u', v')) \\ & +\infty \text{ if} |s_e| > 1 \end{cases}$$

$$G'_2 \begin{cases} \forall v \in V_2, \forall s_v \subseteq V_1, \ \delta^{ged}_{vG_1G_2}(v, s_v) = & c_{vi} \text{ if } s_v = \emptyset \\ & 0 \text{ if } |s_v| = 1 \\ & +\infty \text{ if} |s_v| > 1 \\ \forall (u,v) \in E_2, \forall s_e \subseteq E_1, \delta^{ged}_{eG_1G_2}(u, v, s_e) = & c_{ei} \text{ if } s_e = \emptyset \\ & 0 \text{ if } |s_e| = 1 \\ & +\infty \text{ if} |s_e| > 1 \end{cases}$$

Theorem 7. *Given two labelled graphs* $G_1 = (V_1, E_1, L_1, \alpha_1, \beta_1)$ *and* $G_2 = (V_2, E_2, L_2, \alpha_2, \beta_2)$*, the graph edit distance of Bunke and Jiang [9] is equal to the distance* $\delta^{ged}_{G_1G_2} =< \delta^{ged}_{vG_1G_2}, \delta^{ged}_{eG_1G_2}, \otimes_{\sum} >$ *between the graph* $G'_1 = (V_1, E_1)$ *and the graph* $G'_2 = (V_2, E_2)$*.*

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 a 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 matching

In this subsection we show how to model different multivalent graph matching problems as graph distance measures. These problems being optimisation problems, the objective is always to find the matching giving the lowest distance.

Extended graph edit distance

Problem definition. In order to compare over- and under-segmented images, Ambauen et al. [3] propose to extend ged 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 one of these new operations is weighted by a cost c_{split} and c_{merge} (but, in [3], these costs are set to 0). Finally, non-overlapping constraints are added on the two kinds of "multivalent matching" operations (vertex merging and splitting): if we want to link two vertices u and v of one graph to another vertex u', we need 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 can model the extended graph edit distance with our graph distance measure in the same way that for the (non extended) graph edit distance. The vertex and edge distance functions are similar but must not return $+\infty$ when a multivalent matching is considered and the vertex distance function δ_v must take into account the vertex merging and splitting operation costs. However, this modelisation does not care of the non-overlapping constraint. To modelize exactly the graph edit distance, we need to use a more sophisticated vertex distance function and a function $\otimes$ different of the function $\otimes_{\sum}$ in order to check the non-overlapping constraint. The idea is to define distance functions in such a way that the matching m can be reconstructed from the distances, and then to check in the $\otimes$ function that the considered matching satisfies the non-overlapping constraints. We do not present here this more complicated modelisation because we propose a modelisation of the graph similarity of Champin and Solnon [11] in section 5 and that it has been shown in Sorlin and Solnon [28] that one can compute the extended graph edit distance by computing this graph similarity measure.

However, if we consider the extended graph edit distance without the non-overlapping constraints, the proof of correctness can be trivially done in the same way than for non-extended graph edit distance: each multivalent matching corresponds to an extended error-tolerant graph matching and our distance function can weight this graph matching in the same way than Ambauen and al. did [3].

Non bijective graph matching problem

Definition. Boeres and al. [6] propose a non-bijective 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 needs to use a non-bijective 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 respects application dependant constraints. More formally, two graphs are used to represent the problem: the model graph $G_1 = (V_1, E_1)$ and the image graph $G_2 = (V_2, E_2)$ (with $|V_1| \leq |V_2|$). A solution is a matching $m \subseteq V_1 \times V_2$ between G_1 and G_2 such that each vertex of G_1 is linked to a non-empty set of connected vertices of G_2 (*i.e.*, $\forall v \in V_1, |m(v)| \geq 1$ and the vertices of $m(v)$ are connected by edges of E_2) –in order to only merge connected regions–, and each vertex of G_2 is linked to exactly one vertex of G_1 (*i.e.*, $\forall v \in V_2, |m(v)| = 1$). Finally, some couples of vertices cannot be matched together. Given any matching that respects these constraints, a similarity measure $sim[6]_m$ is computed with respect to a vertex and an edge similarity function $sm_v : V_1 \times V_2 \rightarrow [0,1]$ and $sm_e : E_1 \times E_2 \rightarrow [0,1]$ as follows:

$$sim[6]_m = \frac{\sum_{(u,v)\in m} sm_v(u,v)}{|V_1|.|V_2|} + \frac{\sum_{(u,v)\in (V_1\times V_2)-m} 1 - sm_v(u,v)}{|V_1|.|V_2|} + \frac{\sum_{((u,u'),(v,v'))\in E_1\times E_2, \{(u,v),(u',v')\}\in m} sm_e((u,u'),(v,v'))}{|E_1|.|E_2|} + \frac{\sum_{((u,u'),(v,v'))\in E_1\times E_2, \{(u,v),(u',v')\}\notin m} 1 - sm_e((u,u'),(v,v'))}{|E_1|.|E_2|}$$

Measure definition. By properly choosing vertex and edge distance functions δ_v and δ_e, we can model the similarity of Boeres and 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 function reproduce the similarity matrices sm_v and sm_e. More formally:

$$G_1 \begin{cases} \forall v \in V_1, \forall s_v \subseteq V_2, \delta_v^{nbgm}(v, s_v) = \sum_{v'\in s_v} 1 - sm_v(v,v') \\ \qquad\qquad + \sum_{v'\in V_2 - s_v} sm_v(v,v') \\ \qquad\qquad \text{if } connected(v, s_v) \\ \qquad\qquad +\infty \text{ otherwise} \\ \forall (u,v) \in E_1, \forall s_e \subseteq E_2, \delta_e^{nbgm}((u,v), s_e) = \sum_{(u',v')\in s_e} 1 - sm_e((u,v),(u',v')) \\ \qquad\qquad + \sum_{(u',v')\in E_2 - s_e} sm_e((u,v),(u',v')) \end{cases}$$

$$G_2 \begin{cases} \forall v \in V_2, \forall s_v \subseteq V_1, \delta_v^{nbgm}(v, s_v) = 0 \text{ if } allowed(v, s_v) \\ \qquad\qquad\qquad\qquad\qquad\qquad +\infty \text{ otherwise} \\ \forall (u', v') \in E_2, \forall s_e \subseteq E_1, \delta_e^{nbgm}((u', v'), s_e) = 0 \end{cases}$$

where *connected* and *allowed* are two functions introduced to check the constraints. *connected* is a function returning $+\infty$ when a vertex of the model is not matched or when it is matched to a non-connected set of vertices and 0 otherwise. *allowed* is a function returning $+\infty$ when a vertex of the image is not matched to only one allowed vertex of the model and 0 otherwise:

$$\begin{aligned} \forall v \in V_1, \forall s_v \subseteq V_2, connected(v, s_v) = &\text{ true if } s_v \text{ is a non-empty set of} \\ &\text{connected vertices} \\ &\text{false otherwise} \\ \forall v \in V_2, \forall s_v \subseteq V_1, allowed(v, s_v) = &\text{ true if } s_v = \{v'\} \wedge (v, v') \text{is allowed} \\ &\text{false otherwise} \end{aligned}$$

Theorem 8. *If the matching m minimizing the distance $\delta^{nbgm} =< \delta_v^{nbgm}, \delta_e^{nbgm}, \otimes_{\sum} >$ gives rise to a non-infinite 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 functions *connected* and *allowed*, the distance between G_1 and G_2 with respect to a matching m is equal to $+\infty$ if and only if m is a matching that violates at least one constraint. Finally, by decomposing the vertex and edge distance functions, we can prove that the distance δ^{nbgm} is inverse to the similarity of [6] and as a consequence, the matching minimizing the distance δ^{nbgm} is the matching that maximizes the similarity of Boeres et al.

5 Graph similarity of Champin and Solnon [11]

In [28], we show that the similarity of Champin and Solnon [11] is generic in the sense that, by properly instanciating parameters of this measure, it can be used to solve all the graph matching problems listed above. 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 in the sense that they can express the same graph matchings problems.

5.1 Definition of the graph similarity of Champin and Solnon [11]

The measure of Champin and Solnon is defined for multi-labelled graphs, *i.e.*, graphs where a non-empty set of labels is associated to 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 multi-labelled 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 to 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 **maximal 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 generic in the sense that they can be used to model many other graph distance and similarity measures from the litterature. We show here that these two measures have the same ability to represent graph matching problems.

Theorem 9. *Given a graph similarity measure of Champin and Solnon (defined by the two functions f and g) of two multi-labelled graphs $G_1 = \langle V_1, r_{V1}, r_{E1} \rangle$ and $G_2 = \langle V_2, r_{V2}, r_{E2} \rangle$ defined over the same sets L_V and L_E of vertex and edge labels, there exists a distance definition $\delta =< \delta_v, \delta_e, \otimes >$ between two graphs $G'_1 = (V_1, E_1)$ and $G'_2 = (V_2, E_2)$ such that the matching m which minimizes the distance δ between G'_1 and G'_2 is the matching which maximizes the similarity of G_1 and G_2.*

Proof. In order to make the proof, we first show that a multi-labelled graph $G = \langle V, r_V, r_E \rangle$ can be modelled by a graph $G' = (V, E)$ with two functions l_v and l_e labelling the vertices and the edges of G'. Then, we show that it is possible to define the distance functions δ_v and δ_e in such a way that the arguments of the function $\otimes$ contains the matching done (*i.e.*, all the information required to compute the arguments of the functions f and g). As a consequence, the function $\otimes$ can compute the same value than the functions f and g in the similarity of Champin and Solnon.

For each multi-labelled graph $G = \langle V, r_V, r_E \rangle$ defined over the sets L_V and L_E of vertex and edge labels, one can define the graph $G' = (V, E)$ and two labelling functions:

- $E = \{(u, v), \exists (u, v, l) \in r_E\}$
- $l_v : V \rightarrow \wp(L_V), \forall v \in V, l_v(v) = \{l / (v, l) \in r_V\}$
- $l_e : E \rightarrow \wp(L_E), \forall (u, v) \in E, l_e(u, v) = \{l / (u, v, l) \in r_E\}$

Once the graphs to be compared are defined, one needs to define the vertex and edge distance functions. These functions must return a value corresponding to the matching done. More formally, as the set of vertices is finite, we can define a bijective function $num : \wp(V'_2) \rightarrow N$ that associates a unique integer value to every different subset of vertices of G'_2. The function num is used by the vertex distance function δ_v to return the set of vertices of G'_2 matched to each vertex of G'_1:

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

With such a function δ_v, the set of tuples $S = \{(u, \delta_v(u, m(u))) / u \in V\}$ can be used to reconstitute the matching m done:

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

The set S of these tuples being a subset of its arguments, the function $\otimes$ can be defined in such a way that $\otimes$ reconstitutes the matching m and computes the sets $G \sqcap_m G'$ and $splits(m)$. As a consequence, the function $\otimes$ can reconstitute the arguments of the functions f and g[5].

Theorem 10. *Given a distance definition $\delta =< \delta_v, \delta_e, \otimes >$ between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, there exists a graph similarity measure of Champin and Solnon (defined by the two functions f and g) of two multi-labelled graphs $G'_1 = \langle V_1, r_{V1}, r_{E1} \rangle$ and $G'_2 = \langle V_2, r_{V2}, r_{E2} \rangle$ such that the matching m which minimizes the distance δ between $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the matching which maximizes the similarity of G'_1 and G'_2.*

Proof. In order to make the proof, we show that, by properly choosing the multi-labelled 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 compute the functions δ_v, δ_e and $\otimes$.

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we define the multi-labelled 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:

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

With such multi-labelled graphs, the set $G'_1 \sqcap_m G'_2$ of common labels recovered by m contains all the information required to reconstitute the matching m done:

$$m = \{(u, v) / \exists (u, (u, v)) \in G'_1 \sqcap_m G'_2\}$$

The set $G'_1 \sqcap_m G'_2$ being its argument, the function f can be defined in such a way that f reconstitutes the matching m and computes the values of the functions δ_v, δ_e and $\otimes$. The property holds.

The measure of Champin and Solnon and our distance measure have the same ability to represent graph matching problems. However, our distance measure is more flexible. If vertices and edges have properties represented by continuous values, our distance measure is able to compare these values in a straightforward way while the similarity of Champin and Solnon can only binarily compare properties represented by discrete values. Furthermore, some constraints on the matching are easier to express with our distance than with the similarity of Champin and Solnon. For example, the matching constraint "to be only matched to vertices having a given property" is easily expressed into the function δ_v of our distance and is difficult to express with the similarity of Champin and Solnon because this similarity is defined with respect to the set of labels recovered by the matching.

[5] Note however 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 consequence.

6 Computing the distance between two graphs

All matching problems described in section 4 are NP-complete or NP-hard problems, except for the graph isomorphism problem, the complexity of which is not exactly stated[6]. 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 [11] and for computing the extended graph edit distance of Ambauen and al. [3]. 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 [11, 28, 25, 26] three incomplete algorithms for computing the similarity of Champin and Solnon. These algorithms can easily be adapted to compute our graph distance.

Greedy algorithm. We propose in [11] 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 [11] 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 [11, 28] a reactive tabu search.

A local search [16, 21] tries to improve a solution by locally exploring its neighborhood: the neighbours 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 neighbour to neighbour until the optimal solution is found (when the optimal value is known) or until a maximum number of moves have been performed. The tabu meta-heuristic [16, 24] is used to choose the next neighbour to move on. At each step, the best neighbour, *i.e.*, the one that most increase the similarity, is chosen. To avoid staying around locally optimal matchings by

[6] For particular graphs (such as trees or planar graphs) the graph isomorphism problem is polynomial ([1, 19, 23]) ; in general case, the graph isomorphism problem clearly belongs to NP but has not be proved to belong in P neither to be NP-complete.

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 [5] introduced *Reactive Search* where the length of the tabu list is dynamically adapted during the search.

Ant Colony Optimization. We also proposed in [25, 26] to use the Ant Colony Optimization (ACO) meta-heuristic approach to compute the similarity of Champin and Solnon. The ACO meta-heuristic is a bio-inspired approach [15, 14] 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: *(i)* ants lay pheromone trails on the components of the construction graph to keep track of the most promising components, *(ii)* 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 *(iii)* 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 non-directed graph that associates a vertex $< (u, u') >$ to 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 added 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 path 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 non-bijective graph matching problems of Boeres et al. [6]. Each one of these problems has been transformed into a graph similarity measure computing problem and 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. These results can be explained by the fact that our algorithms do not use any kind of filtering techniques and potentialy explore all kinds of mappings, even multivalent ones. On the 7 instances of the non-bijective graph matching problem, our algorithms obtain better results than $LS+$, the reference algorithm of [6] (6 instances on 7 are better solved by reactive tabu search and 7 instances on 7 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 the same 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 need to quickly compute a "good" solution of hard instances or if instances are easy, we can use tabu 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 paper, 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_v and δ_e used to introduce the application dependant 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 which is used as a starting point of the two other algorithms, a reactive tabu local search and an Ant Colony Optimization algorithm to improve the solutions obtained by the greedy algorithm. These two last algorithms obtain complementary results. These algorithms are generic so that they can be used to solve any kind of graph matching problem.

References

1. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms.* Addison Wesley, 1974.

2. T. Akutsu. Protein structure alignment using a graph matching technique, citeseer.nj.nec.com/akutsu95protein.html, 1995.
3. 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*, pages 95–106. Springer, 2003.
4. R. Baeza-Yates and G. Valiente. An image similarity measure based on graph matching. In *Proceedings of 7th Int. Symp. String Processing and Information Retrieval*, pages 28–38. IEEE Computer Science Press, 2000.
5. R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. In Springer-Verlag, editor, *Algorithmica*, volume 29, pages 610–637, 2001.
6. M. Boeres, C. Ribeiro, and I. Bloch. A randomized heuristic for scene recognition by graph matching. In *WEA 2004*, pages 100–113, 2004.
7. H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18:689–694, 1997.
8. H. Bunke. Graph matching : Theoretical foundations, algorithms, and applications. In *Proc. Vision Interface 2000, Montreal*, pages 82–88, 2000.
9. H. Bunke and X. Jiang. *Graph Matching and Similarity*, volume Teodorescu, H.-N., Mlynek, D., Kandel, A., Zimmermann, H.-J. (eds.): Intelligent Systems and Interfaces, chapter 1. Kluwer Academic Publishers, 2000.
10. Horst Bunke. Recent developments in graph matching. In *ICPR 2000*, pages 2117–2124, 2000.
11. 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-Verlag, pages 80–95, 2003.
12. 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.
13. A. Deruyver, Y. Hodé, E. Leammer, and J.-M. Jolion. Adaptive pyramid and semantic graph: Knowledge driven segmentation. In Luc Brun and Mario 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, 2005.
14. 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.
15. M. Dorigo and T. Stützle. Ant colony optimization. *MIT Press*, 2004.
16. F. Glover. Tabu search - part I. *Journal on Computing*, pages 190–260, 1989.
17. A. Hlaoui and S. Wang. A new algorithm for graph matching with application to content-based image retrieval. *LNCS*, Volume 2396, 2002.
18. L. Holm and C. Sander. Mapping the protein universe. *Science 273*, pages 595–602, 1996.
19. J.E. Hopcroft and J-K Wong. Linear time algorithm for isomorphism of planar graphs. 6^{th} *Annu. ACM Symp. theory of Comput.*, pages 172–184, 1974.
20. 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.
21. S. Kirkpatrick, S. Gelatt, and M. Vecchi. Optimisation by simulated annealing. In *Science*, volume 220, pages 671–680, 1983.
22. D. Lin. An Information-Theoretic Definition of Similarity. In *proc. of ICML 1998, 15th Inter. Conf. on Machine Learning*, pages 296–304. M. Kaufmann, 1998.

23. E.M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer System Science*, pages 42–65, 1982.
24. S. Petrovic, G. Kendall, and Y. Yang. A Tabu Search Approach for Graph-Structured Case Retrieval. In *STAIRS 2002*, pages 55–64, 2002.
25. O. Sammoud, C. Solnon, and K. Ghédira. Ant algorithm for the graph matching problem. In *5th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2005)* -, volume 3448 of *LNCS*, pages 213–223. Springer, April 2005.
26. O. Sammoud, S. Sorlin, C. Solnon, and K. Ghédira. 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, April 2006.
27. 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.
28. S. Sorlin and C. Solnon. Reactive tabu search for measuring graph similarity. In Luc Brun and Mario Vento, editors, *5th IAPR-TC-15 workshop on Graph-based Representation in Pattern Recognition*, pages 172–182. Springer Verlag, 2005.
29. A. Tversky. Features of Similarity. In *Psychological Review*, volume 84, pages 327–352. American Psychological Association Inc., 1977.
30. 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, pages 832–836. Springer, 2005.