MULTIPLE GRAPH MATCHING AND APPLICATIONS

Albert Sole Ribalta

Dipòsit Legal: T.1211-2012

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Albert Solé Ribalta

MULTIPLE GRAPH MATCHING AND APPLICATIONS

TESIS DOCTORAL

dirigida pel Dr. Francesc Serratosa Casanelles

Departament
d'Enginyeria informàtica i matemàtiques

UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
DEPARTAMENT D'ENGINYERIA INFORMÀTICA I MATEMÀTIQUES
Av. Països Catalans, 26
43007 Tarragona
Tel. 34 977 559 703
Fax 34 977 559 710
secdeim@urv.cat

FAIG CONSTAR que aquest treball titulat "Multiple Graph Matching and Applications", que presenta Albert Solé Ribalta per a l'obtenció del títol de Doctor, ha estat realitzat sota la direcció del Dr. Francesc Serratosa Casanelles del Departament d'Enginyeria Informàtica i Matemàtiques de la Universitat Rovira i Virgili.

Tarragona, 4 de Juliol de 2012

El director de la tesis doctoral

Dr. Francesc Serratosa i Casanelles
Universitat Rovira i Virgili

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Dedication

To everyone who has contributed on it.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Acknowledgments

Suposo que tothom qui ha passat pel camí sap que una tesis doctoral no només és esforç d'una persona sinó que també de tot l'entorn que et rodeja incloent director, companys de treball, família i amics. Per tant m'agradaria abans de començar la descripció de la feina donar especial gracies al director Francesc Serratosa per totes les hores que ha passat a la pissarra discutint amb mi com també per les hores que ha passat corregint documents. Als companys de treball, en especial al Gerard Sanromà per la feina feta junts i als nous companys de la Università "Ca' Foscari" di Venezia, en especial a Nicola Rebagliati, Samuel Rota Bulò, Marcello Pelillo i Andrea Marin per la gran acceptació que m'han donat. Finalment, també m'agradaria agrair a la meva parella Dèlia, als pares Josep i Dolors, a la meva germana Anna i a tota la família el suport donat en tot moment durant el llarg camí fins arribar aquí.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Acknowledgments

This research is supported by “Consolider Ingenio 2010”: project CSD2007-00018, by the CICYT project DPI2010-17112 and by the Universitat Rovira I Virgili through a PhD research grant.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Abstract

In pattern recognition, the use of graphs is, to a great extend, appropriate and advantageous. Usually, vertices of the graph represent local parts of an object while edges represent relations between these local parts. However, its advantages come together with a sever drawback, the distance between two graph cannot be optimally computed in polynomial time. Taking into account this special characteristic the use of graph prototypes becomes ubiquitous. The applicability of graphs prototypes is extensive, being the most common applications clustering, classification, object characterization and graph databases to name some. However, the objective of a graph prototype is equivalent to all applications, the representation of a set of graph. To synthesize a prototype all elements of the set must be mutually labeled. This mutual labeling consists in identifying which nodes of which graphs represent the same information in the training set. Once this mutual labeling is done the set can be characterized and combined to create a graph prototype. We call this initial labeling a common labeling. Up to now, all state of the art algorithms to compute a common labeling lack on either performance or theoretical basis. In this thesis, we formally describe the common labeling problem and we give a clear taxonomy of the types of algorithms. Six new algorithms that rely on different techniques are described to compute a suboptimal solution to the common labeling problem. The performance of the proposed algorithms is evaluated using an artificial and several real datasets. In addition, the algorithms have been evaluated on several real applications. These applications include graph databases and group-wise image registration. In most of the tests and applications evaluated the presented algorithms have showed a great improvement in comparison to state of the art applications.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Contents

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 1

INTRODUCTION

1. MOTIVATION

In pattern recognition, the use of graphs, or similar relational structures, as a framework to model problems (where relational information between sets of objects is of main importance) is, to a great extend, appropriate and advantageous. Consider, as an example, an object recognition application. Usually, vertices of the graph represent local parts of an object while edges represent relations between these local parts. Because of this clear and intuitive representation, since the 80s, graphs have become increasingly important. One of the key advantages of using graphs structures is that the same representational model is able to fit a wide range of problems from image understanding to interaction networks. Consequently, algorithms based on graph structures are suitable in a very large problem space. There is an interesting review of graph representation models, graph matching algorithms and applications in (Conte, Foggia et al. 2004), in addition, to have a complete overview of the state of the art one should also consider embedding and kernel methods (Bunke and Riesen 2012).

Given a set of objects/elements which are represented using attributed graphs (Wong and You 1985), a graph prototype (Wong and You 1985; Wong, Constant et al. 1990; Jiang, Münger et al. 2001) of them is a structure addressed to represent or summarize this set. Usually, the graph prototype is computed or selected to minimize the distance to all elements it represents.

The applicability of graphs prototypes is extensive, being the most common applications clustering (Bunke, Foggia et al. 2003; Hlaoui and Wang 2006; Torsello and Hancock 2007; Xia and Hancock 2008; Lozano,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Escolano et al. 2009; Xia and Hancock 2009) and classification (Wong and You 1985; Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004; Lozano and Escolano 2006; Ferrer, Valveny et al. 2010) in several fields. In addition, graph prototypes can be used in numerous other problems like object characterization (Serratosa, Alquézar et al. 2003; Mukherjee, Singh et al. 2009) and database constructions (Berretti, Bimbo et al. 2001; He and Singh 2006; Serratosa, Solé-Ribalta et al. 2010; Serratosa, Solé-Ribalta et al. 2011) among others (Chan and Cheung 1992; Jiang, Münger et al. 2001; Serratosa, Alquézar et al. 2003; Lozano, Escolano et al. 2009) . The advantage of using a graph prototype instead of working with the whole set it represents is twofold. Firstly, it is common to obtain a higher recognition ratio since the implicit noise in the elements of the set is compensated. Secondly, once the prototype is constructed, the application run time is reduced since less computational effort is needed. On the other hand, depending on the application, graph prototypes tend to over-generalize the set they represent distorting application performance, this is clearly seen in (Ferrer, Valveny et al. 2010) where classification results are better using 1-NN classification procedure than the generalized/set median graph. In addition, some prototypes can be computationally hard to construct if the training set is large.

Graph prototypes can be constructed using an unsupervised or semi-supervised learning process. In the unsupervised learning a set of graph is presented to the system and neither information about the number of classes nor the class each graph belongs to is given. The system decides which partitions the set of graphs contain and it construct a prototype for all of them. This idea is much related to central graph clustering. But frequently, graphs prototypes are constructed using a semi-supervised learning procedure. In this type of learning, the system is provided with information about the class each graph belongs and the problem is focused on constructing the prototype.

In both types of learning, all graphs that belong to the same class (being this class hypothetical for the case of unsupervised learning or known for the case of semi-supervised learning) must be mutually labeled before (Bonev, Escolano et al. 2007; Xia and Hancock 2008; Lozano, Escolano et al. 2009; Xia and Hancock 2009) or while (Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004) the prototype is constructed. This mutual labeling consists in identifying which nodes of which graphs represent the same information in the training set. Once this mutual labeling is done, that is, we know the possible values of each node and edge, the set can be characterized and combined to create a graph prototype. This synthesis can be achieved by averaging the attributes (Jiang, Münger et al. 2001), modeling the attributes with random variables (Wong and You 1985; Bagdanov and Worring 2003), creating fuzzy sets (Chan and Cheung 1992), using histograms (Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004) and so on. Note that

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

this initial mutual labeling is crucial to construct a good prototype. In the case that the initial mutual labeling is incorrect, the graph prototype could contain a noisy representation of the data which will hinder the learning process and decrease the performance of the final application. We call the result of this initial mutual graph labeling among all the elements of the training set a *Common Labeling*. Specifically, the *Common Labeling* of a set of graphs is a bijective function from the nodes of the graphs to a set of labels or virtual nodes.

To illustrate the aforementioned procedure, Figure 1-3-1 shows an example of the process of constructing a graph prototype using a semi-supervised learning process. Suppose we have a set of hand-drawn electronic schemas, and we want to digitalize these drawings. The first step is to obtain a model from each component, such as the resistors. To do so, we initially generate a graph that represents each resistor in the training set. Then, with the set of graphs that represent the resistors, we compute a graph prototype which represents the whole set of resistors. To compute the prototype, first it is needed to mutually label all training data (in this case all the example resistors). That is, we need to compute the *Common Labeling* of the training set. In the example, the *Common Labeling* assigns all the nodes of all the graphs to a *virtual node set*. Each node of this virtual set represents a local part of the object. Once this common labeling is known, the prototype can be constructed using any of the state-of-the-art methods.

Since finding the optimal labeling between nodes of two graphs is already in an NP problem (Garey and Johnson 1979) in its simplest expression, we consider that computing the common labeling of a graph set is also at least an NP problem. As a consequence sub-optimal algorithms must be provided.

2. AIMS AND OBJECTIVES OF THE PHD THESIS

Considering, as exposed in section 1, the crucial importance of a good initial common labeling in the construction of a graph prototype and so in any of its applications, the main aim of this PhD thesis is to provide fast and reliable algorithms to compute a consistent multiple isomorphism between a set of graphs. Due to the exponential complexity of the problem, we present six different sub-optimal algorithms that return an approximation of the optimal and common labeling solution in polynomial time.

In the way of this research and with the desire to give solid theoretical basis to the presented methods and algorithms, two other important fields related to structural pattern recognition have been studied. The first one is related to the study in depth of the graph edit distance between two graphs. Although this distance has been used throughout 30 years, some theoretical

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

aspects never have been studied. In this thesis, we move one step further to the comprehension of this distance that has been shown to be appropriate in many applications. Additionally, we will analyze several applications where these new theoretical aspects can be applied. The second research field focuses on a new algorithm to compute the edit distance between two graphs. The developed algorithm relies on the association graph, a very solid theoretical framework to solve the graph isomorphism problem. Related to the main aim of the thesis this pair-wise matching algorithm has been extended to the group-wise case. Since the error tolerant consistent multiple isomorphism is a generalization of the error tolerant isomorphism problem this initial development and evaluation of the pair-wise case represents an advantage to ensure the functionality of the algorithm for the group-wise case. Finally to evaluate the proposed algorithms, we have selected several methods and applications that need to find a consistent multiple isomorphism between a set of graphs. Methods involve important fields such as synthesis of graph class prototypes, graph database construction and group-wise image registration. In this last application the common labeling aids to the increase of the quality of individual graph correspondences given highly noise applications.

3. ORGANIZATION

The thesis document is organized in 7 chapters.

Chapter 2 is devoted to describe the required concepts for the rest of the document. Concepts described overview the basics of graph matching, types of bijections between graphs and the state of the art of algorithms to compute bijections between two graphs. Once these concepts are overviewed, the chapter extends all previous definitions to the group-wise case. To this aim, concepts like the multiple isomorphism, consistent multiple isomorphism and common labeling are formally defined. Next, the state of the art of algorithms to compute a common labeling are detailed. The rest of the chapter concerns the analysis of several applications where the common labeling problem arises. These applications include graph prototype construction and graph databases.

Chapter 3 describes several new properties of the graph edit distance. These new properties, previously unknown, focus on the particularization of a class of costs. The properties describe the shape and the interpretation of a class of costs. The end of the chapter reviews several possible applications of the new properties and give some directions to improve several existing graph matching algorithms.

Chapter 4 describes a new algorithm to compute a bijection between two graphs that minimize the graph edit distance. The algorithm relies on the

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

concept of dominant set, which is a generalization of the association graph framework to detect similar structures between two attributed graphs.

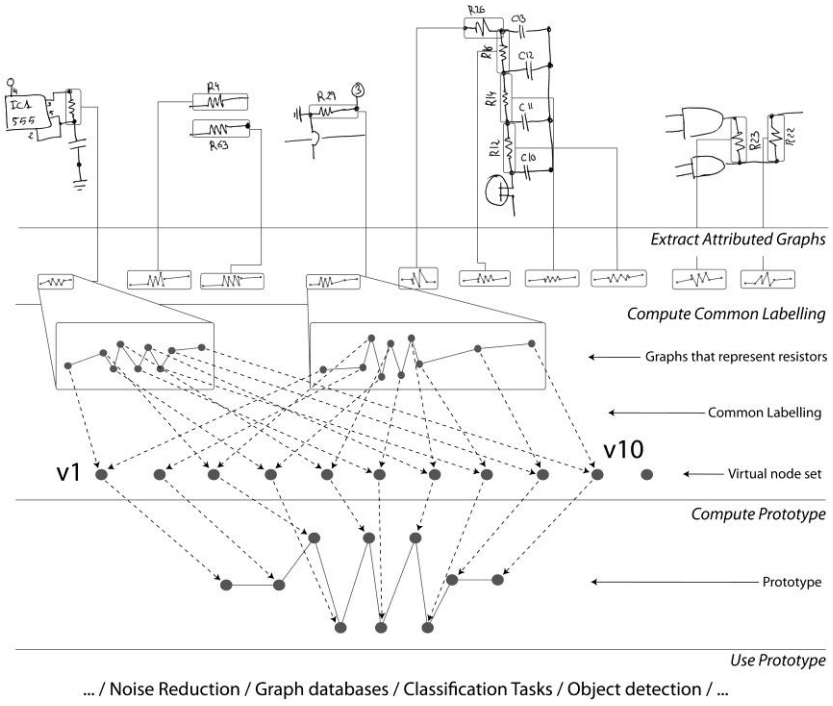


Figure 1-3-1: Graph prototype computation example

Chapter 5 describes six new algorithms to compute a common labeling. The first two are based on a generalization of the graduated assignment, the effectivity of these algorithms is high at the expense of having also a high computational cost. Two of these last algorithms center its objective on minimizing all pair-wise bijections, considering consistency restrictions, among the involved graphs. The other two are focused to minimize explicitly the common labeling.

Chapter 6 evaluates the new proposed methods and compares their results with the state of the art algorithms. The first part of the chapter is addressed to evaluate the cost of the common labeling obtained as a raw measure. The second part evaluates the efficiency of the common labeling on several applications such as prototype construction and graphs databases. The last part of the chapter is dedicated to describe and evaluate how the common labeling can be used as a group-wise image registration algorithm.

Eventually, chapter 7 draws conclusions and gives directions to further work.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 2

STATE OF THE ART AND DEFINITIONS

1. GRAPHS AND ATTRIBUTED GRAPHS

We define a **graph** as an abstract representation of an object. This object is considered to be composed by parts and there exists some relation between the parts. The interconnected parts are represented by abstractions called *vertices (or nodes)*, and the relations that connect some pairs of vertices are called *edges (or arcs)*. More formally, a graph is defined by a tuple $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v = \{v_k \mid k = 1, \dots, N\}$ is the set of vertices (or nodes) and $\Sigma_e \in \left\{e_{ij} \mid i, j \in \{1, \dots, N\}\right\}$ is the set of edges (or arcs). $\gamma_v: \Sigma_v \rightarrow \Delta_v$ and $\gamma_e: \Sigma_e \rightarrow \Delta_e$, where Δ_e and Δ_v represent two possible domains, assigns problem dependent attributes to vertices and edges respectively.

There are several classifications of graphs depending on the characteristics of the composing nodes and vertices. With regard to the information associated to the elements that compose the graphs (nodes and edges), we differentiate between un-attributed and attributed graphs depending if the nodes or edges contain or not contain attributes, that is depending whether functions γ_v and γ_e exist. We also differentiate between directed or undirected graph. Directed graphs are composed by directed edges, which indicate directional information on the relation they encode. Edge directionality can also be encoded into an edge attribute.

From now to the rest of the document, we call attributed graphs to graphs where nodes and edges contain attributes over a possible domain, edges are not directed. In case some directionality needs to be codified, the relation will be encoded into the attribute of the edge.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1.1 Types of Correspondences between graphs

1.1.1 Bijection between two graphs

Let $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q = (\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ be two graphs of order N^p and N^q respectively. A bijection between the two graphs, in case $N^p = N^q$, is assumed to be a function $f^{p,q}: \Sigma_v^p \rightarrow \Sigma_v^q$ which assigns nodes of graph G^p to nodes of graph G^q. The bijection between the graph edges, denoted by $f_e^{p,q}$, must be defined accordingly to the bijection of their terminal nodes. In other words:

$$f_e^{p,q}\left(e_{ab}^p\right) = e_{ij}^q \Rightarrow f^{p,q}\left(v_a^p\right) = v_i^q \wedge f^{p,q}\left(v_b^p\right) = v_j^q \qquad (1.1)$$

In addition, we represent the set of all possible bijections by symbol T.

Depending on the special characteristics of this bijection we can classify them in several types. The following two sections give details of two of them.

1.1.2 Graph isomorphism

Let $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q = (\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ be two un-attributed graphs with the same order, $N^p = N^q$. A bijection $f^{p,q}$ is an isomorphism if and only if for every two nodes $e_{a,b}^p = \left(v_a^p, v_b^p\right)$ adjacent in G^p, nodes $e_{i,j}^q = \left(f^{p,q}\left(v_a^p\right), f^{p,q}\left(v_b^p\right)\right)$ are also adjacent in G^q and vice versa. This type of bijections between two graphs are said to be edge preserving in the sense that every edge in G^p also exist in G^q and vice versa. Formally,

Definition 1-1: given two graphs G^p and G^q, an isomorphism between them is a bijection $f^{p,q}: \Sigma_v^p \rightarrow \Sigma_v^q$ where

$$\left(v_i^p, v_j^p\right) \in \Sigma_e^p \Leftrightarrow \left(f^{p,q}\left(v_i^p\right), f^{p,q}\left(v_j^p\right)\right) \in \Sigma_e^q \qquad (1.2)$$

Two graphs are isomorphic is there exist an isomorphism between them. □

In case, $N^p \neq N^q$ or both graphs are not completely isomorphic, bijection $f^{p,q}$ cannot assign all nodes either in the domain or in the co-domain. In those case the concept of sub-graph isomorphism arises. A sub-graph isomorphism $\hat{f}^{p,q}: \Sigma_v^p \rightarrow \Sigma_v^{q'} \subseteq \Sigma_v^q$ assigns nodes of G^p to a subset of nodes of G^q with the same characteristics of an isomorphism but restricted to the graph induced by $\Sigma_v^{q'}$.

The concept of graph isomorphism in attributed graphs becomes fuzzy, and several authors have slightly different definitions depending on the type of problem. One approach (Jiang and Bunke 1996; Hidovic and Pelillo 2004), which is a strict generalization of the original meaning of graph

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

isomorphism restricts the solution to be isomorphic (edge preserving). In addition, authors give a numerical value to quantify the quality of the bijection with respect to node and edge attributes. Another possible definition (Cordella, Foggia et al. 2004), maybe more intuitive than a formal generalization, defines the compatibility between two nodes or edges based on a semantic similarity measure which asserts if two nodes or edges can be defined as equivalent or not (note that the solution may not be edge preserving in the original sense of graph isomorphism).

1.1.3 Error tolerant bijection

When applying graph matching algorithms over real data on real applications it is possible that, due to the process of sampling, information extraction or modeling, the graph representation of the original data loses some structural information. In these situations, graph isomorphic solutions for the process of computing correspondences between nodes of both graphs may be too restrictive. Is in these situations where error tolerant graph matching (Messmer and Bunke 1997; Bunke 1998; Bunke 2000; Pawar and Zaveri 2011) solutions must be applied. Several other names have been given to the same problem such as error correcting graph matching or error tolerant graph isomorphism. However, the concept is equivalent. Error tolerant graph matching is not only focused in attributed graphs but also in non-attributed graph. In case of non-attributed graphs, the main difference with respect to the original graph isomorphism problem relies on that edge consistency is not necessarily preserved. Equivalently, in attributed graph, edge correspondences given by the solution may not be preserved. In addition, attributes of nodes and edges, in case they exist, they are not mandatory to match the exact value. From now on to the rest of the document, we focus only on attributed graph. However, same methods can be applied for error tolerant graph matching using non-attributed graphs.

Like graph isomorphism, error tolerant bijections relate vertices of one graph to vertices of the other graph. This relation allows applying local distance measures between nodes and edges and also between node and edge attributes. By means of this tight relationship between local costs and the global correspondence between nodes of both graphs, graph matching algorithms usually compute error tolerant bijections by minimizing some objective function which relates vertex and edge correspondences with vertex and edge attributes.

In general, given two graphs $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q = (\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ where $\left|\Sigma_v^p\right| = \left|\Sigma_v^q\right| = N$, the general objective function to optimize corresponds to the quadratic assignment problem (Du and Pardalos 1998) objective function. That is:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$C(G^p, G^q, F^{p,q}) = \sum_{a=1}^{N}\sum_{\substack{b=1\\b\neq a}}^{N}\sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N} F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{E}{}_{ai,bj}^{p,q} + \\ + \sum_{a=1}^{N}\sum_{i=1}^{N} F^{p,q}[a,i]\cdot C_{V}{}_{a,i}^{p,q} \tag{1.3}$$

where F represents a continuous or discrete assignment matrix. Usually, in case of discrete assignments the matrix is described as a permutation matrix:

$$F^{p,q}[a,i] = \begin{cases} 1 & if\ f^{p,q}(v_a^p) = v_i^q \\ 0 & otherwise \end{cases} \tag{1.4}$$

on the other hand, if the matrix is defined in a continuous form, is usually interpreted as a probability of assigning two nodes:

$$F^{p,q}[a,i] = P(v_a^p \to v_i^q) \tag{1.5}$$

In both cases, we can generalize the restriction of this matrix by:

$$\sum_{a=1}^{N} F^{p,q}[a,i] = 1, \forall i \in \{1..N\}$$
$$\sum_{i=1}^{N} F^{p,q}[a,i] = 1, \forall a \in \{1..N\} \tag{1.6}$$
$$F^{p,q}[a,i] > 0, \forall a,i \in \{1..N\}$$

Cost functions $\mathcal{C}_{E}{}_{ai,bj}^{p,q}: \Sigma_e^p \times \Sigma_e^q \to \mathbb{R}$ and $\mathcal{C}_{V}{}_{a,i}^{p,q}: \Sigma_v^p \times \Sigma_v^q \to \mathbb{R}$ assign a cost or compatibility value, depending if the problem is addressed to minimize or to maximize (1.3), between any edge or nodes correspondence. We represent compatibility by C and cost by $\mathcal{C}$. Compatibility and cost values are complementary and can be transformed using several functions:

$$C = e^{-k\cdot\mathcal{C}}$$
$$C = \frac{1}{k+\mathcal{C}} \tag{1.7}$$
$$C = \frac{k-\mathcal{C}}{k}$$

where k represents a scaling constant to reach the zero compatibility when $\mathcal{C}$ is maximum. We prefer to use the second or the third functions of (1.7) due to its lower computational cost.

A more compact representation of objective function in (1.3) can be given by :

$$C(G^p, G^q, F^{p,q}) = \sum_{a=1}^{N}\sum_{\substack{b=1\\b\neq a}}^{N}\sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N} F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \tag{1.8}$$

where the combined compatibility $C_{ai,bj}^{p,q}$ is given by:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$C_{ai,bj}^{p,q} = \frac{1}{N-1}\left(C_V{}_{a,i}^{p,q} + C_V{}_{b,j}^{p,q}\right) + C_{E}{}_{ai,bj}^{p,q} \tag{1.9}$$

since each node forms part of $N-1$ edges (note that we do not count self-loops on vertices).
From now to the rest of the document, we will use this compact representation since it ease the notation and it is numerically equivalent.

1.1.3.1 Graph extension with null nodes

From the formalization of the error tolerant graph matching problem it can be seen that, due to restrictions given in (1.6), the solution is forced to match all vertex set $\Sigma_{\mathrm{v}}^{\mathrm{p}}$ of the first graph to all vertex set of the second graph Σ_v^q. The situation can be involved if both graphs just contain a sub-part in common that have to be corresponded and the rest of both graph must remain unmatched. In addition, the same situation appears when both graphs have different size. Since now, we restricted the size of both involved graphs G^p and G^q to have the same cardinality, $\left|\Sigma_v^p\right| = \left|\Sigma_v^q\right| = N$, which obviously do not solve the aforementioned problems but give a nice and clear optimization problem. To keep these nice properties of the formalization given in (1.3) and (1.6), the most used solution (Wong and You 1985; Bunke 1998; Lohmann and Cramon 2000; Myers and Hancock 2000; Myers and Hancock 2000; Gautama, Bellens et al. 2006; Justice and Hero 2006; Ferrer, Valveny et al. 2009; Riesen and Bunke 2009; Raveaux, Burie et al. 2010) is to extend both graphs with null-nodes that we identify with symbol $\emptyset$. These null nodes do not contain attributes and so the cost $C_V{}_{a,i}^{p,q}$ is defined in a slightly different form. We will not focus in this definition because it is specific of the problem and also on the graph distance we aim to optimize. In section 1.1.4, we will specify this cost for the graph edit distance (Sanfeliu and Fu 1983). Considering this new representation, given two graphs G^p and G^q with $\left|\Sigma_v^p\right| = N^p$ and $\left|\Sigma_v^q\right| = N^q$, the extended graphs should be of size $\left|\Sigma_v^p\right| = N^p + N^q$ and $\left|\Sigma_v^q\right| = N^p + N^q$ to allow all possible node assignations and the extreme case where any node of both graphs is identified to be common. However, this amount of extra nodes increases the effective cost of the computation. To reduce this sever drawback of the solution, usually some heuristics (Myers, Wilson et al. 1999) can be applied to estimate the amount of overlapping between both graphs. Depending on the problem is also common to add null edges between nodes which are not adjacent. In the same way, null edges are not attributed.

1.1.4 Graph edit distance

One of the most widely used methods to evaluate an error correcting graph isomorphism is the Graph Edit Distance (Sanfeliu and Fu 1983). The

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

basic idea behind the Graph Edit Distance is to define a dissimilarity measure between two graphs. This dissimilarity, equivalently to the idea of string edit distance (Navarro 2001), is defined as the minimum amount of distortion required to transform one graph into the other. To this end, a number of distortion or edit operations, consisting of insertion, deletion and substitution of both nodes and edges are defined. Then, for every pair of graphs (G^p and G^q), there is a sequence of edit operations, or an edit path $editPath(G^p, G^q) = (\varepsilon_1, \dots, \varepsilon_k)$ (where each ε_i denotes an edit operation) that transform one graph into the other. In general, several edit paths may exist between two given graphs. This set of edit paths is denoted by ϑ. To quantitatively evaluate which edit path is the best, edit cost functions are introduced. These edit functions assign a penalty cost to each edit operation according to the amount of distortion that they introduce in the transformation.

Each $editPath(G^p, G^q) \in \vartheta$ can be related to a univocal graph isomorphism $f^{p,q} \in T$ between the involved graphs. In this way, each edit operation assigns a node of the first graph to a node of the second graph. Deletion and insertion operations are transformed to assignations of a non-null node of the first or second graph to a null node of the second and first graph. Substitutions simply indicate node-to-node assignations. Using this transformation, given two graphs, G^p and G^q, and a bijection $f^{p,q}$ between their nodes, the graph edit cost is given by (Definition 7 of (Bunke 1999)):

$$\begin{aligned} C_{GED}\,(G^p, G^q, f^{p,q}) = \\ = \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{ns}(v_a^p, v_i^q) + \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \hat{\Sigma}_v^q}} C_{nd}(v_a^p, v_i^q) + \\ + \sum_{\substack{v_a^p \in \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{ni}(v_a^p, v_i^q) + \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{es}\left(e_{ab}^p, e_{ij}^q\right) + \\ + \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \hat{\Sigma}_e^q}} C_{ed}\left(e_{ab}^p, e_{ij}^q\right) + \sum_{\substack{e_{ab}^p \in \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{ei}\left(e_{ab}^p, e_{ij}^q\right) \end{aligned} \tag{1.10}$$

$$f^{p,q}(v_a^p) = v_i^q \text{ and } f_e^{p,q}(e_{ai}^p) = e_{ij}^q \tag{1.11}$$

where $\hat{\Sigma}_v^p$ and $\hat{\Sigma}_e^p$ refer to null nodes and null edges[1], C_{ns} is the cost of substituting node v_a^p of G^p for node $f^{p,q}(v_a^p)$ of G^q, C_{nd} is the cost of

[1] we consider a null edge an edge that does not exist on the graph. Null edges do not contain attributes. Usually, graphs are extended with null edges to be complete and consequently null edges are considered in the cost computation.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

deleting node v_a^p of G^p and $\mathcal{C}_{ni}$ is the cost of inserting node v_i^q of G^q. Equivalently for edges, $\mathcal{C}_{es}$ is the cost of substituting edge e_{ab}^p of graph G^p for edge $f_e^{p,q}(e_{ab}^p)$ of G^q, $\mathcal{C}_{ed}$ is the cost of assigning edge e_{ab}^p of G^p to a non-existing edge of G^q and C_{ei} is the cost of assigning edge e_{ab}^q of G^q to a non-existing edge of G^p.

Finally, the Graph Edit Distance is defined as the minimum cost under the set of all possible bijections T:

$$D_{GED}(G^p, G^q) = \min_{f^{p,q} \in T} \mathcal{C}_{GED}(G^p, G^q, f^{p,q}) \tag{1.12}$$

It is common to encode the graph edit distance problem into an optimization framework and solve the problem using a permutation matrix instead of computing the edit path explicitly (Torsello and Hancock 2003; Justice and Hero 2006; Neuhaus and Bunke 2007; Riesen and Bunke 2009). This approach has the advantage of not dealing with equivalent edit paths, like in (Ferrer, Valveny et al. 2010). Usually, the objective function to optimize is related to the quadratic assignment problem as defined in (1.3) or what is the same as (1.8). In this formulation, as commented previously, it is usual to extend the graph with null nodes to be of order $|\Sigma_v^p| = |\Sigma_v^q| = N^p + N^q$, and the costs $\mathcal{C}_{V_{ai}}^{p,q}$ and $\mathcal{C}_{E_{ai,bj}}^{p,q}$ that compose $\mathcal{C}_{ai,bj}^{p,q}$ are defined as:

$$\mathcal{C}_{V_{ai}}^{p,q} = \begin{cases} \mathcal{C}_{ns}(v_a^p, v_i^q) & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \wedge v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \\ \mathcal{C}_{nd}(v_a^p, v_i^q) & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \wedge v_i^q \in \hat{\Sigma}_v^q \\ \mathcal{C}_{ni}(v_a^p, v_i^q) & v_a^p \in \hat{\Sigma}_v^p \wedge v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \end{cases}$$

$$\mathcal{C}_{E_{ai,bj}}^{p,q} = \begin{cases} \mathcal{C}_{es}(e_{ab}^p, e_{ij}^q) & e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \wedge e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q \\ \mathcal{C}_{ed}(e_{ab}^p, e_{ij}^q) & e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \wedge e_{ij}^q \in \hat{\Sigma}_e^q \\ \mathcal{C}_{ei}(e_{ab}^p, e_{ij}^q) & e_{ab}^p \in \hat{\Sigma}_e^p \wedge e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q \end{cases} \tag{1.13}$$

It is usual that, since null nodes and edges do not contain attributes, to model functions $\mathcal{C}_{nd}$, $\mathcal{C}_{ni}$, $\mathcal{C}_{ed}$ and $\mathcal{C}_{ei}$ with constant costs K_{nd}, K_{ni}, K_{ed} and K_{ei}. However, some other definitions exist. Some of them will be explained in Chapter 3.

1.1.5 Graph matching algorithms

The literature on graph matching algorithms is extensive and algorithms range from the 70th up to now. We will review the basic literature and methods. In addition, since most of the document is focused on the graph edit distance, the end of the section is devoted specifically to algorithms that minimize that distance.

To survey graph matching algorithms one can rely on different taxonomies. These taxonomies involve types of graphs, types of solutions,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

optimality of the solution and so on. This section reviews algorithms from graph isomorphism, error tolerant graph matching, and among them structural matching and attributed graph matching. We will also differentiate between optimal and suboptimal algorithms. Several surveys and related articles can be found, we highlight (Conte, Foggia et al. 2004; Gallagher 2006).

One of the primal problems related to graph matching is the graph isomorphism problem. This problem was one of the first problems to be studied in depth considering its importance for the pattern recognition community and the difficulty to compute solutions in reasonable time.

The first proposed optimal methods where proposed for non-attributed graph. Most of them are based on back-track search (Ullmann 1976) or state space representations of the matching process (Cordella, Foggia et al. 1999). Since the first proposed methods most of the work has been focused on pruning the search space to speed up the matching process (Ballabh 1988; Krissinel and Henrick 2004; Battiti and Mascia 2007). But, the problem remains of exponential computational cost in the worst case scenario. One way to overcome this drawback is to focus on specific graphs such as with graph with unique node labels (Dickinson, Bunke et al. 2003). In this way, the matching process can fall into polynomial time algorithms to compute the optimal solution. Still on the graph isomorphism problem, but moving now to the attributed graph case, we can find some works (McGregor 1982; Cordella, Foggia et al. 1999). Considering the exponential complexity of the general problem several suboptimal solutions have been proposed to compute the isomorphism between two graphs in polynomial time. One of the most popular is related to the association graph and the Motzkin-Straus theorem (Pelillo 1999). This solution will be explained in detail further in this chapter. Some other approaches to sub-optimally compute graph isomorphic solutions could be found (Massaro and Pelillo 2001; Fosser, Glantz et al. 2003; Massaro and Pelillo 2003).

As commented earlier, since some amount of noise is expected in the input data graph, isomorphic solutions applied to pattern recognition problems are usually too restrictive. It is in these situations where models relying on error tolerant solutions improve performance. In both cases, when data is represented by attributed graphs or non-attributed graphs, a large amount of solutions have been proposed. Algorithms proposed for non-attributed graphs, could be based on optimization techniques (Finch, Wilson et al. 1998; Bin and Hancock 2001; Wang and Hancock 2009), spectral graph theory (Caelli and Kosinov 2004; Qiu and Hancock 2006), some heuristic technique such as (Zhu, Qin et al. 2001) based on some incremental matching plus a refinement process, the association graph (Tang, Jiang et al. 2011) or other less popular techniques such as quantum walks (Emms, Wilson et al. 2008). When data is represented by attributed graph, the literature is still more extensive since attributed graphs are often

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

used in pattern recognition applications. Several optimal algorithms exist (Tsai and Fu 1979; Llados, Marti et al. 2001; Cordella, Foggia et al. 2004), which as usual are based on a tree search with some pruning heuristics or similar procedures (Hlaoui and Wang 2002). It is important to see that with attributed graphs, it is possible to obtain a higher degree of pruning since more information is considered (structure and attributes). For particular graphs, such as graph with unique node labels, linear time optimal algorithms are available (Dickinson, Bunke et al. 2003).

Taking into account that optimal labelings are usually too expensive to compute, even though high pruning heuristics have been designed, a large set of suboptimal algorithms have been proliferated. Some of them based on optimization techniques (Christmas, Kittler et al. 1995; Wilson and Hancock 1997; Wyk and Wyk 2004; Schellewald and Schnörr 2005), some others on spectral properties of graphs and their adjacency matrices (Umeyama 1988 ; Zhao, Luo et al. 2007) or less commonly other techniques (Gori, Maggini et al. 2005; Jain and Wysotzki 2005; Othman, Abdullah et al. 2008; Kim, Yun et al. 2010). Up to now, we mainly have considered bijective solutions, but other types of solutions could be considered, such as a many-to-many correspondences (Keselman, Shokoufandeh et al. 2003; Demirci, Osmanlioglu et al. 2011).

Most of the previously cited algorithms are generic graph matching algorithms, but since there are several standard distance measures to compare two graph (Sanfeliu and Fu 1983; Shapiro and Haralick 1985; Bunke and Shearer 1998), there are also several algorithms that specifically minimize these distances. We focus now on algorithms that minimize the graph edit distance (Sanfeliu and Fu 1983). The proposed algorithms that minimize a distance related to graph edit distance between two graphs is considerably large. However, not many algorithms to minimize the specific graph edit distance as proposed in (Bunke 1998) exist. There is a great survey of most of the proposed algorithms in (Gao, Xiao et al. 2010). We highlight some and then we describe in detail the graph matching algorithms that will be used in the document.

Since the graph edit distance is a well established distance measure between two graphs, it has been applied to a large set of problems. We differentiate between two types of input data, trees and graphs. The tree edit distance seems more widely used than the graph edit distance. Target applications could be addressed to match shapes using shock graphs (Klein, Tirthapura et al. 2000; Torsello and Hancock 2003; Sebastian, Klein et al. 2004), images (Todorovic and Ahuja 2007) or to analysis of glycan structures (Fukagawa, Tamura et al. 2011) to name some. In most of the tree edit distance proposed solutions, it is common to carry out the matching process using the tree closure to be resistant to node merging (Torsello and Hancock 2003; Torsello, Robles-Kelly et al. 2007). Thus, the final type of data to match are not trees but directed acyclic graphs. More general

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

approaches exist for graph, such as (Llados, Marti et al. 2001; Ambauen, Fischer et al. 2003; Berretti, Bimbo et al. 2004; Justice and Hero 2006; Neuhaus and Bunke 2007; Riesen, Neuhaus et al. 2007; Riesen and Bunke 2009). Neuhaus (Neuhaus and Bunke 2007) use a quadratic assignment formulation to solve the problem. Justice and Hero (Justice and Hero 2006) approach the problem using a linear formulation obtaining especially good results. However, just discrete or symbolic labels on nodes can be used. Other works (Llados, Marti et al. 2001; Ambauen, Fischer et al. 2003; Berretti, Bimbo et al. 2004) with the aim of obtaining a more flexible matching process enhance the graph edit distance with some extra operations. These operations are related to shift, merge and split node operations. Lately, (Riesen, Neuhaus et al. 2007; Riesen and Bunke 2009) propose a very fast algorithm to compute the graph edit distance. This algorithm is based on the linear assignment problem that can be optimally solved with a cost of at most $O(N^3)$ (Kuhn 1955; Munkres 1957).

In the following subsection, we will explain in detail three graph matching algorithms that will be used throughout the document.

1.1.5.1 Graduated assignment

The Graduated Assignment algorithm (Gold and Rangarajan 1996) is probably the most popular algorithm to compute a sub-optimal solution for the graph isomorphism problem. Its cornerstone is how it reduces the isomorphism problem to the quadratic assignment problem (Garey and Johnson 1979). The proposed development starts by defining the energy of an isomorphism between two graphs. Thus, given two graphs $G^p = \left(\Sigma_v^p, \Sigma_e^p, \gamma_v, \gamma_e\right)$ and $G^q = \left(\Sigma_v^q, \Sigma_e^q, \gamma_v, \gamma_e\right)$ where $\left|\Sigma_v^p\right| = \left|\Sigma_v^q\right| = N$ and F corresponds to a continuous assignation matrix, the objective function becomes:

$$E^{p,q} = -c_1 \sum_{a=1}^{N}\sum_{i=1}^{N}\sum_{\substack{b=1 \\ b\neq a}}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \qquad (1.14)$$
$$c_1 = \frac{1}{N(N-1)}$$

In fact, in the original paper (Gold and Rangarajan 1996), the normalization constant is $1/2$ instead of $1/(N(N-1))$. We prefer to use this constant because it restricts the energy in the range $[-1,0]$ instead of $(-Inf, 0]$.

In (Gold and Rangarajan 1996), $E^{p,q}$ is approximated, at point $(F^{p,q})^0$ using Taylor series expansion as:

$$E(G^p, G^q, F^{p,q}) \approx \tilde{E}(G^p, G^q, F^{p,q}) = \qquad (1.15)$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$= -c_1 \sum_{a=1}^{N}\sum_{i=1}^{N}\sum_{\substack{b=1\\ b\neq a}}^{N}\sum_{j\neq i}^{N}(F^{p,q}[a,i])^0 \cdot (F^{p,q}[b,j])^0 \cdot C_{ai,bj}^{p,q} -$$

$$-c_1 \sum_{a=1}^{N}\sum_{i=1}^{N}\left[\sum_{\substack{b=1\\ b\neq a}}^{N}\sum_{j\neq i}^{N}(F^{p,q}[b,j])^0 \cdot C_{ai,bj}^{p,q}\right](F^{p,q}[a,i] - (F^{p,q}[a,i])^0) \cdot C_{ai,bj}^{p,q}$$

Analyzing the approximation it can be seen that:

$$\begin{aligned} F^{p,q*} = arg\min_{F^{p,q}} E(G^p, G^q, F^{p,q}) &\approx arg\min_{F^{p,q}} \tilde{E}(G^p, G^q, F^{p,q}) \\ &\equiv arg\max_{F^{p,q}} \sum_{a=1}^{N}\sum_{i=1}^{N} Q^{p,q}[a,i] \cdot F^{p,q}[a,i] \end{aligned} \tag{1.16}$$

where

$$Q^{p,q}[a,i] = \sum_{b=1}^{N}\sum_{j=1}^{N}(F^{p,q}[b,j])^0 \cdot C_{ai,bj}^{p,q} \tag{1.17}$$

Note that the same conclusions can be achieved with the objective function in (1.8). That is,

$$\begin{aligned} F^{p,q*} = arg\max_{F^{p,q}} C(G^p, G^q, F^{p,q}) &\approx arg\max_{F^{p,q}} \tilde{C}(G^p, G^q, F^{p,q}) \\ &\equiv arg\max_{F^{p,q}} \sum_{a=1}^{N}\sum_{i=1}^{N} Q^{p,q}[a,i] \cdot F^{p,q}[a,i] \end{aligned} \tag{1.18}$$

The algorithm proposed in (Gold and Rangarajan 1996) minimizes $E(G^p, G^q, F^{p,q})$ under the assumption that it is minimised at the same point in which (1.16) is maximize. Consequently, the problem is equivalent to the linear assignment problem (Du and Pardalos 1998), where $Q^{p,q}$ represents a cost matrix and $F^{p,q}$ represents a doubly stochastic matrix (Sinkhorn 1964) which contains the desired assignation probability.

The Graduated Assignment algorithm proceeds in the following way: start with a valid $(F^{p,q})^0$, compute cost matrix $Q^{p,q}$ given by (1.17), apply Graduated Assignment to compute $F^{p,q}$ and start again. A pseudo-code of the Graduated Assignment is listed in Algorithm 1-1.

The description of the parameters of the algorithm is given below:

- $\boldsymbol{\beta_0}$: initial value of β. If β_0 is low the algorithm is more likely to adapt the isomorphism correctly at the initial steps of the algorithm, since $e^{\beta \cdot Q^{p,q}}$ will have similar values.
- $\boldsymbol{\beta_f}$: latest value of β. If β_f is high, then the exponential function $e^{\beta \cdot Q^{p,q}}$ approaches $F^{p,q}$ to a permutation matrix obtaining only almost 0 or 1 values. Therefore, $F^{p,q}$ becomes almost a doubly stochastic matrix and the discretization of $F^{p,q}$ to obtain the resulting bijection may not be required.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

- $\boldsymbol{\beta_r}$: increment of β. The smaller the increment, the better the algorithm captures the solution with minimum energy. However, the run time is higher.
- $\boldsymbol{I_0}$ and $\boldsymbol{I_1}$: maximum number of iterations of loops in lines 6 and line 10 in case $F^{p,q}$ does not converge given a concrete β value. If these values are too low, the system will not have time to capture the global knowledge. But if they are too high, the run time increases without increasing the quality of the isomorphism.

The proposed values for these constants are suggested in Table 1-1. Function Λ indicates a final discretization process to convert the doubly stochastic matrix to a permutation matrix.

```
1   Algorithm Graduated Assignment( G^p, G^q )
2     F^{p,q} = Initialize();
3     β = β_0
4     repeat until β > β_f
5       t_0 = 0
6       repeat until Q^{p,q} converges or t_0 > I_0
7         Q^{p,q}[a,i] = Σ_{b=1, b≠a}^{N} Σ_{j=1, j≠i}^{N} F^{p,q}[b,j] · C^{p,q}_{ai,bj}
8         F^{p,q}[a,i] = e^{β·Q^{p,q}[a,i]}
9         t_1 = 0
10        repeat until H converges or t_1 > I_1
11          F^{p,q}[a,i] = F^{p,q}[a,i] / Σ_{x=1}^{N} F^{p,q}[x,i]
12          F^{p,q}[a,i] = F^{p,q}[a,i] / Σ_{x=1}^{N} F^{p,q}[a,x]
13          t_1 = t_1 + 1
14        end
15        t_0 = t_0 + 1
16      end
17      β = β · β_r
18    end
19    f = Λ(F^{p,q})
20  Return f
```

Algorithm 1-1: Graduated Assignment algorithm.

	Description	Value
β_0	Start value for Graduated Assignment control parameter	0.05
β_r	Increment rate for Graduated Assignment control parameter	1.075
β_f	Maximum value for Graduated Assignment control parameter	150
I_0	Number of iterations with the same control parameter value	4
I_1	Maximum number of iterations for Sinkhorn method	30

Table 1-1**:** proposed value for graduated assignment algorithm.

1.1.5.2 Bipartite Graph Matching

The bipartite graph matching algorithm (Riesen, Neuhaus et al. 2007; Riesen and Bunke 2009) has lately shown very good performance on several

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

databases. Considering its performance and speed, we consider it a good algorithm to compute the graph edit distance.

The algorithm approximates the graph edit distance (quadratic assignment problem) to the linear assignment problem by using a cost matrix $\mathcal{C}_{a,i}^{p,q}$. They define the cost matrix as follows.

Definition 1-2: given two graphs $G^p = \left(\Sigma_v^p, \Sigma_e^p, \gamma_v, \gamma_e\right)$ and $G^q = \left(\Sigma_v^q, \Sigma_e^q, \gamma_v, \gamma_e\right)$ where $\left|\Sigma_v^p\right| = N$ and $\left|\Sigma_v^q\right| = M$.The cost matrix $\mathcal{C}$ is defined as:

$$\mathcal{C} = \left[\begin{array}{cccc|cccc} \overbrace{\mathcal{C}_{1,1}^{p,q} & \mathcal{C}_{1,2}^{p,q} & \cdots & \mathcal{C}_{1,M}^{p,q}}^{A} & & & & \overbrace{}^{B} \\ \mathcal{C}_{2,1}^{p,q} & \mathcal{C}_{2,2}^{p,q} & \cdots & \mathcal{C}_{2,M}^{p,q} & & & & \\ \vdots & \vdots & \ddots & \vdots & & & & \\ \mathcal{C}_{N,1}^{p,q} & \mathcal{C}_{N,2}^{p,q} & \cdots & \mathcal{C}_{N,M}^{p,q} & & & & \\ \hline K_n & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & K_n & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \cdots & K_n & 0 & 0 & \cdots & 0 \\ \underbrace{}_{R} & & & & \underbrace{}_{S} & & & \end{array}\right] \tag{1.19}$$

where $\mathcal{C}_{a,i}^{p,q} = \mathcal{C}_V{}_{a,i}^{p,q}$ denotes the node substitution cost and K_n the graph edit distance node insertion and deletion constant. □

Using this cost matrix $\mathcal{C}$ either the Hungarian algorithm (Kuhn 1955) or the Munkres algorithm (Munkres 1957) can be used to compute the bijection $f^{p,q}$ that minimize the correspondences.

Note that the cost matrix given in (1.19), nodes do not consider the edges of the graph. To do so, matrix $\mathcal{C}$ is proposed to be extended to include edge costs. Thus, to each entry of $\mathcal{C}_{a,i}^{p,q}$ the minimum sum of edge edit operations, computed through the Munkres algorithms, is added. This solutions assumes that adjacent edges of vertex v_a^p should be equivalent to adjacent edges of v_i^q. Given this new matrix $\mathcal{C}'$ the linear assignment problem is solved in the same way.

1.1.5.3 Association graph and Dominant sets

One of the techniques to solve the graph isomorphism problem is through the use of association graphs. This framework reduces the graph isomorphism problem to the maximum clique problem (Bomze, Budinich et al. 1999) which by the Motzkin-Straus (Motzkin and Straus 1965) theorem is proven to be equivalent to a particular quadratic program.

1.1.5.3.1 Association graph construction

Given a non-attributed graph $G^p = (\Sigma_v^p, \Sigma_e^p)$, we say that two nodes v_i^p and v_j^p are adjacent if $\left(v_i^p, v_j^p\right) \in \Sigma_e^p$. Related to this adjacency property of

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

vertices, we define the adjacency matrix of G^p as an $A = (a_{i,j}\ ) \in \mathbb{R}^{N\times N}$ symmetric matrix where:

$$a_{i,j} = \begin{cases} 1 & if\ \left(v_i^p, v_j^p\right) \in \Sigma_e^p \\ 0 & otherwise \end{cases} \tag{1.20}$$

Definition 1-3: Given two graphs G^p and G^q, we define its association graph as $\tilde{G} = (\tilde{V}, \tilde{E})$ where:

$$\tilde{V} = \Sigma_v^p \times \Sigma_v^q$$
$$\tilde{E} = \left\{ \left(v_a^p, v_i^q\right) \times \left(v_b^p, v_j^q\right) \middle| \begin{cases} \left(\left(v_a^p, v_b^p\right) \in \Sigma_e^p \wedge \left(v_i^q, v_j^q\right) \in \Sigma_e^q\right) \\ \vee \\ \left(\left(v_a^p, v_b^p\right) \notin \Sigma_e^p \wedge \left(v_i^q, v_j^q\right) \notin \Sigma_e^q\right) \end{cases} \right\} \tag{1.21}$$

In addition, we represent its associated adjacency matrix by $\tilde{A}$.

Definition 1-4: given a graph G, we call a clique a subset $\Sigma_v' \subseteq \Sigma_v$ of vertices where all vertices are mutually adjacent, that is $\forall\ v_i, v_j \in \Sigma_v' \Rightarrow \left(v_i, v_j\right) \in \Sigma_e$.

1.1.5.3.2 The Motzkin-Straus theorem

Let G be an un-attributed graph of cardinality N, and $\boldsymbol{x}$ a point on the standard simplex of N dimensions, that is:

$$\boldsymbol{x} = \{x_i \in \mathbb{R}^+, \|\boldsymbol{x}\|_1 = 1\} \tag{1.22}$$

We denote all points in the simplex of N dimensions as Δ_N.
We define the support of $\boldsymbol{x}$ as $p \in \{0,1\}_N$ where :

$$p = \sigma(\boldsymbol{x}) = \begin{cases} 1 & x_i > 0 \\ 0 & otherwise \end{cases} \tag{1.23}$$

In addition, we define the characteristic vector of $\boldsymbol{x}$ as the barycenter of the simplex face indicated by the support p.

$$x_i^c = \begin{cases} \dfrac{1}{\sum_{i=1}^N p_i} & x_i > 0 \\ 0 & otherwise \end{cases} \tag{1.24}$$

Note that $\boldsymbol{x^c}$ is the point with minimum norm of all the points over the simplex faces indicated by the support of $\boldsymbol{x}$.

The original paper of Motzkin-Straus relates the aforementioned maximum clique problem with the following quadratic form:

$$Q(\boldsymbol{x}) = \boldsymbol{x}A\boldsymbol{x^T} = \langle \boldsymbol{x}, A\boldsymbol{x} \rangle = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{i,j}x_i x_j \tag{1.25}$$

where A is the adjacency matrix of G and T indicates the transpose of a matrix.
A point $\boldsymbol{x}^*$ is a global maximizer of Q if $\forall \boldsymbol{x} \in \Delta_N\ Q(\boldsymbol{x}^*) \geq Q(\boldsymbol{x})$, and a local maximizer if $\forall \boldsymbol{x} \in \Delta_N | \ \|x - x^*\| < \epsilon,\ Q(\boldsymbol{x}^*) \geq Q(\boldsymbol{x})$. $\boldsymbol{x}^*$ is a strict local maximizer if the only point which holds for the $Q(\boldsymbol{x}^*) = Q(\boldsymbol{x})$ is $\boldsymbol{x} = \boldsymbol{x}^*$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

The Motzkin-Straus (Motzkin and Straus 1965) theorem proves a connection between the maximum clique problem and the objective function given in (1.25). It states that a subset of nodes of G is a maximum clique C if and only if its characteristic vector $\boldsymbol{x^c}$ is a global maximizer of $Q(x)$. The equivalent proof is made in (Pelillo and Jagota 1995; Gibbons, Hearn et al. 1997) for strict local maximizes and maximal cliques. These results had a strong impact in the maximum clique problem since it converts a strict combinatorial problem to the continuous domain in a solid theoretical manner. This allows using a large set of existing quadratic optimization methodologies to compute solutions to the maximal clique problem. The main drawback of this original formalization is that it contains spurious solutions. This spurious solutions are formalized in the following theorem (Pelillo and Jagota 1995):

Theorem 1: *Consider a graph G which contains two cliques C and D of equal cardinality $|C| = |D| = k$. Let $|C\backslash D| = |D\backslash C| = m < k$. Then, for every $\alpha_1, \alpha_2 \geq 0$ such that $\alpha_1 + \alpha_2 = 1$:*

1. *If G has exactly $m(m-1)$ edges crossing $C\backslash D$ and $D\backslash C$ then $Q(\alpha_1 x^C + \alpha_2 x^D) = Q(x^C)$.*
2. *If G has fewer edges than $m(m-1)$ crossing $C\backslash D$ and $D\backslash C$ then $Q(\alpha_1 x^C + \alpha_2 x^D) < Q(x^C)$.*

This drawback was solved in (Bomze 1997) by adding a regularization term in the diagonal of the adjacency matrix of the graph. Including this regularization term the quadratic form becomes:

$$Q(\boldsymbol{x}) = \langle \boldsymbol{x}, \left(A + \frac{1}{N} I_N\right) \boldsymbol{x} \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j} x_i x_j + \frac{1}{2} \sum_{i=1}^{N} x_i^2 \tag{1.26}$$

1.1.5.3.3 Computing the graph isomorphism

Considering the section 1.1.5.3.2, it is not hard to formulate the graph/sub-graph isomorphism problem in terms of the association graph and the maximum clique problem. Thus, given two graph G^p and G^q and its respective association graph $\tilde{G}$, with adjacency matrix $\tilde{A}$. The graph/sub-graph isomorphism problem can be stated as:

$$\max_{x \in \Delta_N} Q(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j} x_i x_j + \frac{1}{2} \sum_{i=1}^{N} x_i^2 \tag{1.27}$$

Note that quadratic program of (1.27) is closely related to the formalization of the graph matching problem given by (1.3). And although the search space is different solutions can be straightforwardly adapted since each x_i describes a vertex-to-vertex correspondence given by f in (1.3).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1.1.5.3.4 Dominant sets

The framework described above is restricted to compute the graph isomorphism of un-attributed graph using its clique representation in the association graph. This represents a strong restriction, since lately it is common to represent information using attributed graph what will produce weighted adjacency matrices in the association graph. In this sense, the above framework cannot be used. There are several extensions of the Motzkin-Straus theorem to other types of graph (Gibbons, Hearn et al. 1997; Bomze, Pelillo et al. 2000; Rota-Bulò and Pelillo 2008). In this work, we are interested in (Pavan and Pelillo 2003) where the Motzkin-Straus theorem is extended to edge weighted graphs and is related to the concept of dominant set. The concept of dominant set is defined as a group of nodes all connected with each other with high internal connection between them. In (Pavan and Pelillo 2003; Pavan and Pelillo 2007), they establish a correspondence between dominant sets and local solution of the following optimization problem:

$$\max_{x \in \Delta_N} x^T \tilde{A} x \tag{1.28}$$

where $\tilde{A}$ represents a weighted association matrix, and Δ_N represents the simplex of N dimensions.

The dominant set objective function is equivalent to the one defined for the un-weighted case, that is (1.27), but with a continuous definition of matrix A weights.

Our final aim is to use the association graph framework to compute a solution for the graph edit distance problem. As an advance of concept and to justify the decision of focusing on dominant sets, it is worth to say that we will codify the problem in an association graph where each cell of the adjacency matrix will contain the cost of matching the respective elements. In this way, isomorphism restrictions may not be fulfilled and all nodes, with some restrictions, are susceptible to be matched. This methodology will encode every possible bijection as a dominant set in the association graph.

2. MULTIPLE ISOMORPHISM BETWEEN A SET OF GRAPHS

This section generalizes the concept of pair-wise graph matching to the concept group-wise graph matching. In addition, we will overview and formally describe several related problems and summarize state of the art algorithms.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2.1 Multiple isomorphism and related problems

Let $\Gamma = \{G^1, G^2, \dots, G^P\}$ be a set of P attributed graphs of order $|\Sigma_v^1|$, $|\Sigma_v^2|$, ..., $|\Sigma_v^P|$. Assume for the moment that all attributed graphs have $|\Sigma_v^1| = |\Sigma_v^2| = \cdots = |\Sigma_v^P| = N$ nodes.

Definition 2-1: the set φ is a multiple isomorphism of Γ if it contains one and only one isomorphism between attributed graphs in Γ, $\varphi = \{f^{1,2}, \dots, f^{p,q}, \dots, f^{N,N-1}\}$ being $p \neq q$. □

Considering Definition 2-1, we compute the cost of a multiple isomorphism as the sum of the individual costs for all isomorphisms in φ:

$$C_{MI}(\Gamma, \varphi) = \sum_{p=1}^{P} \sum_{\substack{q=1 \\ p \neq q}}^{P} \sum_{a=1}^{N} \sum_{\substack{b=1 \\ b \neq a}}^{N} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} F^{p,q}[a,i] \cdot F^{p,q}[b,j] \cdot C_{ai,bj}^{p,q} \qquad (2.1)$$

□

Note that in φ two isomorphisms exist for any pair of graphs and so it contains $f^{p,q}$ and $f^{q,p}$. Taking into account that a multiple isomorphism could be computed using a sub-optimal pair-wise graph matching algorithm (Rosenfeld, Hummel et al. 1976; Christmas, Kittler et al. 1995; Gold and Rangarajan 1996; Riesen and Bunke 2009; Gao, Xiao et al. 2010), which do not ensure that the result of $f^{p,q}$ and $f^{q,p}$ are the same labeling, the cost of (2.1) must consider both bijections.

With respect to the issue of the number of extra null nodes that should be included in each graph, it is worth to say that this number is in general unknown and it is application dependant. In general, all graphs should be extended with null nodes to have $|\Sigma_v^1| = \cdots = |\Sigma_v^P| = \sum_{i=1}^{P} |\Sigma_v^i|$ nodes. This would allow all graphs not to be labeled to each other in case they are absolutely different. As a particular case if our aim is to compute a multiple isomorphism, that is a one-to-one correspondence between all nodes of any two graphs, it is only required to extend each graph with null nodes to make all input graphs be of the same size in this case $|\Sigma_v^{1..P}| = \max_{i=1..P} |\Sigma_v^i|$. On the other hand, if we just expect some degree of overlap it is usual, to include just some extra null nodes to allow non-common nodes to be assigned to null nodes in the final solution (e.g. 10 or 20 percent). Heuristics for the pair-wise case could be extended to the multiple isomorphism problem.

Definition 2-2: the optimal multiple isomorphism is the one that minimizes (2.1), formally,

$$\varphi^{opt} = arg \min_{\varphi \in MI} C_{MI}(\Gamma, \varphi) \qquad (2.2)$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

We say that the multiple isomorphism φ is *consistent*[2] if composing any subset of bijections from φ we can define separate partitions of vertices (Bonev, Escolano et al. 2007). This concept have already appeared in (Bunke, Münger et al. 1999; Jiang, Münger et al. 2001) and also in a more fuzzy fashion in some graph prototype generation algorithms (Sanfeliu, Serratosa et al. 2004) or generative models (White and Wilson 2008; White 2009). The main idea is to look for multiple isomorphism where each node of each graph is labeled to a single characteristic of the element that graphs represent. Thus, given a set of attributed graphs, each partition contains one and only one vertex per attributed graph and, in addition, every vertex must belong to only one partition. More formally, a consistent multiple isomorphism can be defined as:

Definition 2-3: let φ be a multiple isomorphism of Γ. φ^{cons} is a consistent multiple isomorphism of Γ if it fulfils that:

$$f^{k,q}\left(f^{p,k}\left(v_i^p\right)\right) = f^{p,q}\left(v_i^p\right) \quad 1 \le p,k,q \le P, p \ne q \ne k, 1 \le i \le N \tag{2.3}$$

Furthermore, we define the cost of a consistent multiple isomorphism as the cost of the related multiple isomorphism. As opposed to the meaning of consistency, we consider any multiple isomorphism which does not fulfill (2.3) as inconsistent multiple isomorphism.

Figure 2-1 shows a consistent multiple isomorphism between three attributed graphs, being $N = 2$. We can distinguish two partitions, P^1 and P^2. Figure 2-2 shows the same attributed graphs with an inconsistent multiple isomorphism. In this case, no partitions can be defined since v_1^3 and v_2^3 cannot belong to several partitions.

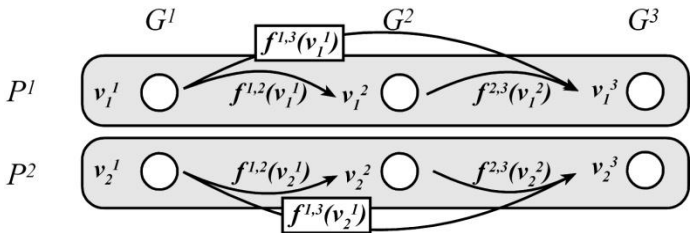


Figure 2-1: Example of consistent multiple isomorphism.

[2] The reader should not confuse our sense of consistent/inconsistent with the meaning of consistent estimator used in statistics.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

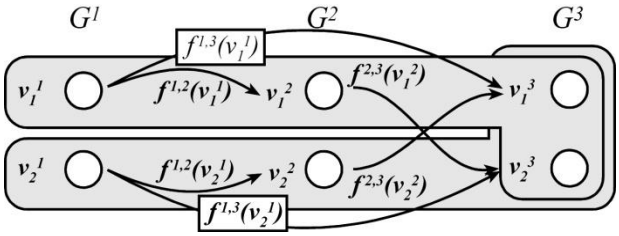


Figure 2-2: example of inconsistent multiple isomorphism.

To quantify the degree of consistency in a multiple isomorphism we define the consistency index as:

Definition 2-4 (Consistency Index) : let φ be a multiple isomorphism of Γ with P graphs of N nodes each graph. We define the Consistency Index as:

$$I_c(\varphi) = 1 - \frac{\sum_{p=1}^{N}\sum_{\substack{q=1\\q\neq p}}^{N}\sum_{\substack{k=1\\k\neq p\\k\neq q}}^{N}\sum_{i=1}^{N}\delta\left(f^{k,q}\left(f^{p,k}\left(v_a^p\right)\right), f^{p,q}\left(v_a^p\right)\right)}{P(P-1)(P-2)\cdot N} \tag{2.4}$$

where δ corresponds to the Kronecker delta function.

□

The optimal consistent multiple isomorphism is the consistent multiple isomorphism with the minimum cost. Note that the minimum cost may be obtained by non-optimal pair-wise isomorphisms since it is restricted to be consistent.

Definition 2-5: let φ^{cons} be a consistent multiple isomorphism of Γ. $\varphi^{opt,cons}$ is an optimal consistent multiple isomorphism of Γ if it fulfils that :

$$\varphi^{opt,cons} = arg\min_{\varphi^{cons}\in CMI} \mathcal{C}_{CMI}(\Gamma, \varphi^{cons}) \tag{2.5}$$

□

2.2 Common labeling

Given Γ, we define a common labeling as a bijective mapping between all nodes in the graphs of Γ to a virtual node set L. We initially construct this common labeling through a consistent multiple isomorphism φ^{cons}. Consistency requirement is mandatory. If this was not the case, an attributed graph node could be labeled to several nodes of the virtual node set which is not allowed in the described framework.

Definition 2-6: let φ^{cons} be a consistent multiple isomorphism of Γ and let $L = \{l_1, l_2, \dots, l_N\}$ be a vertex set, $L \in V$. The common labeling $\psi = \{h^1, h^2, \dots, h^N\}$ is defined to be a set of bijective mappings from the vertices of attributed graphs to L as follows:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\begin{gathered} h^1(v_i^p) = l_i \,, 1 \le i \le N \\ h^p(v_i^p) = h^{p-1}(v_j^{p-1}) \,, 1 \le i,j \le N, 2 \le p \le P \\ being\ f^{p-1,p}(v_j^{p-1}) = v_i^p \end{gathered} \tag{2.6}$$

Note that the common labeling defined in Definition 2-6 and the consistent multiple isomorphisms defined in Definition 2-3 represent the same information. In the consistent multiple isomorphism all pair-wise isomorphisms are explicitly described. However, the common labeling represents the same information using a compact notation.

Figure 2-3 represents a set of attributed graphs and a common labeling among them. Isomorphism between graphs G^1 and G^2 is represented by $f^{1,2}$. Moreover, it can be seen that h^1 is the identity (2.6) and h^2 depends on $f^{1,2}$, i.e. $h^2\left(f^{1,2}(v_i^1)\right) = h^1(v_i^1)$. Finally, we show that h^N is obtained through h^{N-1} and $f^{N-1,N}$.

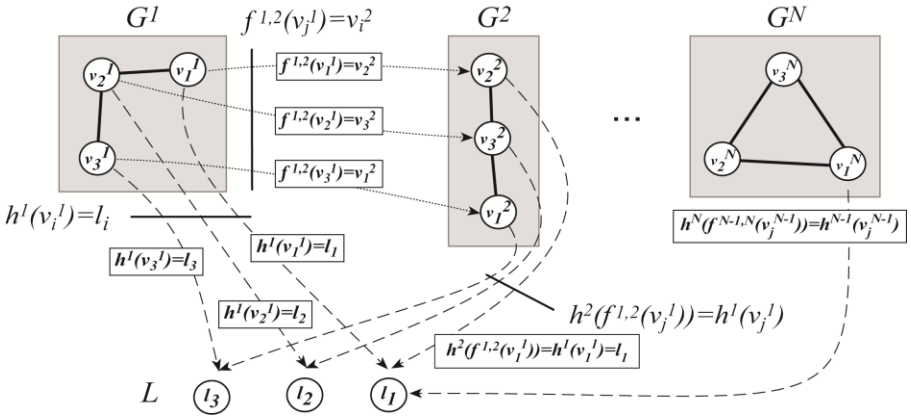


Figure 2-3: graphical representation of a common labeling.

We define the cost of a common labeling as the cost of the related multiple isomorphism (2.1).

$$C_{CL}(\Gamma, \psi) = C_{MI}(\Gamma, \varphi) \tag{2.7}$$

where φ is obtained from ψ through (2.6).

Definition 2-7: the optimal common labeling of a set of attributed graphs Γ is the one that obtains the minimum cost.

$$\psi^{opt} = arg\min_{\psi \in CL} C_{CL}(\Gamma, \psi) \tag{2.8}$$

Note that given Definition 2-2, Definition 2-5 and Definition 2-7, we obtain:

$$C_{MI}(\Gamma, \varphi^{opt}) \le C_{MI}(\Gamma, \varphi^{opt,cons}) = C_{CL}(\Gamma, \psi^{opt}) \tag{2.9}$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2.3 Algorithms

Since the exponential nature of the common labeling problem, no optimal algorithm exists neither for graphs with small cardinality nor for small Γ sets. However, several sub-optimal methods that deal with the common labeling problem, either in an implicit or explicit form, have been proposed in the literature. We have classified them into two categories.

Prototype oriented methods

Prototype oriented methods were the first ones to appear in the literature. Their main purpose is to generate a graph class prototype that represents a given set of graphs. In this type of methods, the common labeling is not explicitly computed. However, it is usually obtained simultaneously with the prototype. They mostly work using two types of prototype synthesis: *Incremental synthesis* and *Agglomerative synthesis*. In **Incremental synthesis methodology,** applied in (Lozano and Escolano 2003; Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004; Weskamp, Hullermeier et al. 2007), the prototype is updated while new graphs are sequentially introduced and labeled to the current prototype. Figure 2-4 gives an example of computing a prototype of four attributed graphs. The advantage of this method is that the learning and recognition processes can be interleaved. Nevertheless, the main drawback of the approach is that, depending on the processing order, different prototypes can be synthesized from the same set of graphs and consequently different common labelings will be obtained. Once the prototype is constructed the common labeling can be deduced from the sequential labelings computed against the current prototype. In **Agglomerative synthesis methodology**, applied in (Wong and You 1985; Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004; Lozano, Escolano et al. 2009)**,** the generated prototype does not depend on the order of the graphs although it is an iterative algorithm. At each step of the process, the pair of temporal prototypes with the minimum distance is merged to generate a new prototype. Figure 2-5 gives an example of computing a prototype of four attributed graphs using agglomerative synthesis methodology. Note that the graphs are not processed sequentially and a distance function guides the synthesis process. The main advantage of using this type of synthesis is that, since at each iteration the closest prototypes are merged, less error in the matching process is expected. As in the incremental methods, the common labeling can be obtained once the final prototype is known.

Prototype oriented methods have the main drawback that in cases where the prototype is unable to capture the structural and semantic information of the graphs involved at the moment, it is difficult to obtain a good common labeling or prototype for the rest of the graph set.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

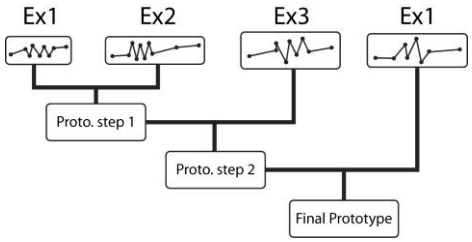


Figure 2-4: incremental Synthesis example.

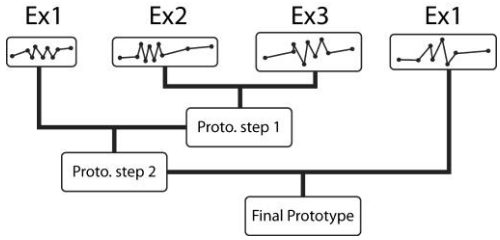


Figure 2-5: agglomerative Synthesis example.

Multiple-Isomorphism Oriented methods

Unlike prototype oriented methods, multiple-isomorphism oriented methods are specifically designed to compute a common labeling. That is, they explicitly compute a mutual labeling among all graphs from the given set. Two types of methods exist. Let's call them **consistent multiple isomorphism methods** and **direct methods**.

Consistent multiple isomorphism methods consist usually in two phases. In the first phase they compute all the pair-wise labelings between the graphs. Then, in the second phase, they deduce a common labeling applying a set of consistency rules or some kind of discretization or cleanup methodology over the bijections deduced in the first step. Figure 2-6 illustrates the process. First using algorithm K all pair-wise labelings are computed, then using a *cleanup* method the final consistent multiple isomorphism is computed.

With regard to these methodologies, we bring attention to the method presented in (Bonev, Escolano et al. 2007). This method will be explained in detail in the following section.

Direct methods compute directly the common labeling without computing explicitly the pair-wise labelings a priori. Up to the knowledge of the author just two methods exists to this aim. The method (Jiang, Münger et al. 2001) codify the common labeling solution in an array of labelings that are optimized using a genetic search algorithm. This method will be explained in detail in Section 2.3.2. The second method (Fober, Mernberger et al. 2009) which also use a genetic algorithm is very similar to the first one.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

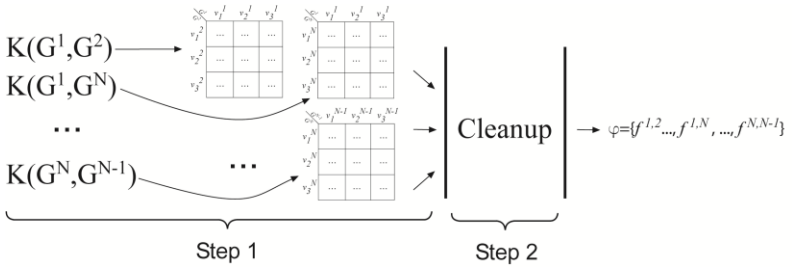


Figure 2-6: multiple isomorphism oriented methods scheme.

Finally, even it is not explicitly related to compute a common labeling, it is worth mentioning another methodology (Williams, Wilson et al. 1997) which uses the common labeling to define a single isomorphism between two graphs. In (Williams, Wilson et al. 1997), representations obtained from Infra-red, Optical, Cartographic and SAR images must be combined and labeled one into the another. Authors of (Williams, Wilson et al. 1997) note that using pair-wise labelings to find correspondences among the different presented datasets, the pair-wise labeling algorithm (Wilson, Evans et al. 1995) was not able to recover any correct label between SAR and Infra-red images. In (Williams, Wilson et al. 1997), they present a multiple graph matching algorithm based on Bayesian inference whereby SAR and Infra-red images are correctly labeled by using Optical and Cartographic images. The main drawback of the methodology is that it cannot be extended to label a set of *N* graphs. However, it clearly shows that, in certain applications where precise isomorphism is required, computing a common labeling instead of pair-wise labelings between graphs reduces matching errors and therefore ensures a better global solution.

The main advantage of using a multiple isomorphism oriented methodologies is that they decouple the common labeling problem from the prototype synthesis, solving one of the drawbacks of the prototype oriented methods. In addition, methods of the second type compute the common labeling considering all the knowledge of the set instead of just using local information provided by the pair-wise labelings. Thus, intuitively this type of methods should obtain better result than the previous ones.

2.3.1 Super Graph Partitions methodology

The Super Graph Partitions method presented in (Bonev, Escolano et al. 2007) computes the common labeling in a pair-wise fashion. The method aims to compute a set of disjoint partitions $P = \{p^1, p^2, \dots p^{max}\}$ between the nodes of each graph in the set Γ. A partition p^i contains one node of each graph and two nodes of the same graph cannot belong to the same partition.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

In this way, once the partitions are computed each partition p^i can be assigned to a label l_i of the virtual set in our definition of the common labeling.

To compute these disjoint partitions, the proposed method applies the Graduated Assignment algorithm (Gold and Rangarajan 1996) to compute all possible pair-wise labelings between the given set of attributed graphs. Thus, the continuous result $F^{p,q}$ of the Graduated Assignment is used to compute the partitions. Each cell of $F^{p,q}[a,i]$ contains the probability of labeling a node v_a^p of G^p to a node v_i^q of G^q. All this assignations are sorted considering its probability and considered in descending order. Assignations with the same probability are sorted using a heuristic such as the distance between attributes in the nodes. Each assignation that relates v_a^p of G^p to a node v_i^q of G^q, is assigned to a partition considering the following rules. The method starts with an empty set of partitions, that is $P = \phi$.

- Neither v_a^p nor v_i^q are assigned to any partition in P. In this case a new partition p^j is created and added to P. Both v_a^p and v_i^q are assigned to p^j.
- v_a^p is assigned to some partition p^j but v_i^q is not. Add v_i^q to the partition p^j if disjoint partition restrictions are satified. Otherwise, add v_i^q to a new partition p^k and add it to P.
- v_i^q is assigned but v_a^p is not. Equivalent case as above, proceed in the same way.
- v_a^p is assigned to partition p^j and v_i^q to partition p^k. If $p^j = p^k$, no action is needed. Otherwise, replace p^j and p^k in P by $p^j \cup p^k$ if super graph restrictions are satisfied.

Once all cells $F^{p,q}[a,i]$ are considered, the common labeling can be generated as commented above.

The method presented in (Bonev, Escolano et al. 2007) has a computational cost of $O(P^2(T \cdot N^4))$, where N represents the number of nodes, P the number of graphs in the set and T the number of iterations of the graduated assignment.

2.3.2 Genetic algorithm

The method proposed in (Jiang, Münger et al. 2001) is not directly addressed to compute the common labeling between a set of graph. However, since the computation of a common labeling is one of the first steps to compute a graph prototype, they implicitly propose a method to do so. As introduced in the above section, the method uses a genetic approach to compute the common labeling. To this aim, each chromosome corresponds to an array of integers that represents the labeling of each graph in $\Gamma = \{G^1, \dots, G^P\}$ to the corresponding median graph $\bar{G}$ (that can be

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

interpreted as the virtual node set L in our definition). Thus, the final chromosome has a size equal to the sum of the cardinalities of the graphs in Γ. Each position of the chromosome corresponds to one node of a graph in Γ and contains either 0 or a value in $\{1..N\}$. Zero indicates that the particular node is removed and a value between $1..N$ indicates that the node is substituted by this candidate of $\bar{G}$. We illustrate the method using the same example they propose in the original paper, see Figure 2-7. Let's consider a set of two graphs $\Gamma = \{G^1, G^2\}$ with $|G^1| = 3$, $|G^2| = 2$. In addition, suppose we aim to construct a median graph such that $|\bar{G}| = 2$. Considering this example the chromosome $Kr = (kr_1, kr_2, kr_3, kr_4, kr_5)$ is defined as an array of five positions. The first three positions, kr_1, kr_2, kr_3, correspond to the nodes of G^1 and the last two, kr_4, kr_5, to the nodes of G^2. The first, node of $\bar{G}$ is assigned to first node of G^2 and to the third node of G^1, nodes two of G^1 and G^2 are deleted from the prototype graph $\bar{G}$.

Given a chromosome, one can compute the fitness function value depending on the objective function to optimize. In the original article, the objective function is related to the median graph that would be constructed through the chromosome. However, in our case, since we do not aim to compute the median graph, the common labeling objective function in (2.7) could be used instead.

The proposed search strategy is based on roulette wheel sampling to select solutions for combination. The combination of the chromosomes is done by single point crossover with a consistency check to force solutions of the common labeling to be consistent bijections. Also mutation is use to randomly change each number of the array with some probability. The consistency check is also applied after the mutation operator.

In the proposed algorithm, some of the initial population of the algorithm is created using some heuristic to ease the convergence of the algorithm. The rest of the initial population is created randomly, but representing consistent bijections between the graphs and the generalized median graph. The original paper proposes two termination conditions. The first sets a maximum number of iterations, so if we reach that maximum we consider the algorithm has finished. The second is related to the convergence of the chromosomes to a single solution. In this way, if the fitness functional of all the population becomes closer than a given threshold it is considered that the algorithm has converged and so the algorithm terminates satisfactory. The computation cost of each iteration of the algorithms is low. However, it is known that the execution time of genetic algorithms highly depends several factors including maximum number of generations or proximity to the solution of the initial population.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

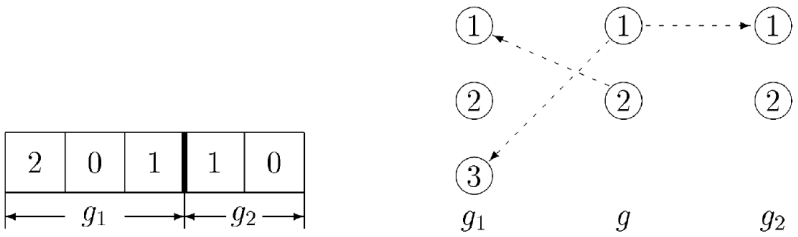


Figure 2-7: chromosome definition and example labeling.

3. CLASS PROTOTYPES AND GRAPH REPRESENTATIVES

We define a graph prototype (Wong, Constant et al. 1990; Bunke, Foggia et al. 2003; Lozano and Escolano 2006; Lozano, Escolano et al. 2009) as a structure that represents a set or a cluster of graphs. Usually, in a pattern recognition application, the system, in the training phase, learns a model from a set of training elements. In structural pattern recognition these elements are commonly represented by graphs. Each graph of the training set corresponds to a perturbed version of the element it represents. Like centroid-based clustering (Hartigan and Wong 1979) one can think to represent each cluster or set of elements with a prototype. Among the possible graph prototypes, we mainly differentiate between two types: **class prototypes** and **graph representatives**. We consider a graph representative a graph in the same domain of the graphs in the training set (Ferrer, Valveny et al. 2009). We consider a class prototype as a graph, in a different domain of the training (Wong, Constant et al. 1990; Bagdanov and Worring 2003; Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004).

There are conceptual similarities between graph prototypes or representatives and centroids in some P-dimensional Euclidean space. On the contrary, differences mainly rely on the computation of the centroid. For centroids in a P-dimensional space one can simply compute a point which minimizes the sum of squared distances to the individual points, let's say by averaging the individual components. However, to compute a graph centroid the issue is completely difference since distance between graphs cannot be computed in polynomial time, and so in practice just approximations can be achieved. As we saw before, these approximations are computed through correspondences between the graphs nodes. Thus, to compute a representative or a graph prototype, we require to compute the best correspondence possible to a virtual structure which independently indentifies the object parts, that is a common labeling. Having this in mind,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

we can state that if we want the prototype to capture the features of the object it represents, it is crucial to find a good common labeling. In this section, we focus our attention on several graph prototypes and two representatives, the generalized median and the set median.

A **Generalized Median Graph** (Jiang, Münger et al. 2001) is an Attributed Graph that minimizes the sum of distances between it and all graphs in the training set. The generalized median graph is usually not a member of the set. And, in general, more than one Median Graph may exist for a given set of graphs. The computation of a Median Graph is of exponential complexity. Nevertheless, several suboptimal methods to obtain approximate solutions for the Median Graph, in reasonable time, have been presented (Bunke, Münger et al. 1999; Ferrer, Valveny et al. 2009; Ferrer, Valveny et al. 2010). These methods apply some heuristic functions in order to reduce the complexity of the graph distance computation and the size of the search space.

Considering a set of Attributed Graphs $\Gamma = \{G^1, \dots, G^p\}$ in domain U a Median Graph $\bar{G} = (\bar{\Sigma}_v, \bar{\Sigma}_e, \bar{\gamma}_v, \bar{\gamma}_e)$ is another attributed graph such that:

$$\bar{G} = \arg\min_{G \in U} \sum_{i=1}^{P} d(G_i, G) \tag{3.1}$$

Since the synthesis of the median graph can differ, we rely on the following definition.

Definition 3-1: A Generalized Median Graph $\bar{G} = (\bar{\Sigma}_v, \bar{\Sigma}_e, \bar{\gamma}_v, \bar{\gamma}_e)$ from a set of Attributed Graphs Γ and an Optimal Common Labeling ψ can be constructed as another attributed graph where attributes on nodes and arcs are computed by:

$$\bar{\gamma}_v(\bar{v}_i) = \sum_{\substack{j=1 \\ \gamma_v(h^{j^{-1}}(\bar{v}_i)) \neq \emptyset}}^{P} \frac{\gamma_v\left(h^{j^{-1}}(\bar{v}_i)\right)}{\sum_{k=1}^{P}\left(1 - \delta(h^{k^{-1}}(\bar{v}_i), \emptyset)\right)} \tag{3.2}$$

where δ corresponds to the Kronecker delta function. In case they exist, attributes for edges are computed in an equivalent manner. The main idea of (3.2) is that the attribute values of the Median Graph is the mean of the values of all the nodes or arcs of the Attributed Graphs it represents. In case a node or arc does not exist its value is not considered to compute the mean.

The **Set Median Graph** (Jiang, Münger et al. 2001) is an alternative to Generalized Median Graph. The difference between the two models consists in the search space where the Median Graph is looked for. The search space for the Generalized Median Graph is U that is the whole universe of Attributed Graphs. In contrast, the search space for the Set Median Graph is simply the set of graphs that represents:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\bar{G} = \arg\min_{G \in \Gamma} \sum_{i=1}^{P} d(G_i, G) \tag{3.3}$$

The computation of Set Median Graph is exponential with respect to the cardinality of the graphs, due to the complexity of graph distance, but quadratic with respect to the number of graphs in the set. In some applications, Set Median Graphs are preferred to Generalized Median Graphs considering two main reasons. First, practical evaluations show that the capacity of Set Median Graphs to represent a set is almost similar to the capacity of the Generalized Median Graphs (Ferrer, Valveny et al. 2009; Ferrer, Valveny et al. 2010). And second, the synthesis (using the whole set of graphs or incrementally) is less computationally demanding.

A **Closure Graph** (He and Singh 2006) is class prototype where the structure of the attributes is different from the attributed graphs that represent. The structurally similar nodes of the set of attributed graphs that have different attribute values are represented in the Closure Graph with only one node but with more than one attribute. Closure Graphs need the attributes in the nodes or edges to be discrete; if this is not the case, an extra discretization phase has to be performed. Closure Graphs need a few more physical space than Median Graphs.

Formally, a Closure Graph $\bar{G} = (\bar{\Sigma}_{\mathrm{v}}, \bar{\Sigma}_{\mathrm{e}}, \bar{\gamma}_{\mathrm{v}}, \bar{\gamma}_{e})$ is a graph where the node and arc attributes are represented as an array of values in the domain of the nodes and arcs of the attribute graphs. The Closure Graph is synthesized from a set of Attributed Graphs Γ and a Common Labeling ψ as follows:

$$\begin{aligned} \hat{\gamma}_v(\hat{v}_a) &= \left\{ a_k \,\middle|\, a_k = \gamma_v\left(h^{i^{-1}}(\hat{v}_a)\right), 1 \le i \le |\Gamma| \right\} \\ \hat{\gamma}_v(\hat{e}_{a,b}) &= \left\{ b_k \middle|\, b_k = \gamma_e(e_{a',b'}), h^i(v_{a'}^i) = \hat{v}_a, h^i(v_{b'}^i) = \hat{v}_b, 1 \le i \le |\Gamma| \right\} \end{aligned} \tag{3.4}$$

The reasoning behind (3.4) is that nodes or arcs of the Closure Graph can take all values that nodes or arcs of the Attributed Graphs have taken. If a node or arc, in the attributed graphs, does not exist, a special null label is introduced in the node or arch of the Closure Graph to represent it.

A **First-Order Random Graph** (FORG) (Wong and You 1985) is a class prototype graph that contains first-order probabilities on nodes and arcs attributes. The first-order probabilities are modeled with a random variable associated with each vertex or arc which represents the attribute information of the corresponding graph nodes and arcs in the training set of Attributed Graphs. This random variable has a one-dimensional probability density function defined over the same attribute domain of the Attributed Graphs, including a null value that denotes the non-instantiation of a FORG graph node or arc.

First Order Random Graphs are the first probabilistic models that appeared in the literature to represent a set of Attributed Graphs. It assumes that the Attributed Graphs in a set or cluster had similar local parts. Nevertheless, in practical applications some graphs can be quite different

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

despite of belonging to the same class. For this reason, representing a set of graphs with only first order probabilities seems to be too restrictive.

Formally, a First Order Random Graph $\bar{G} = (\bar{\Sigma}_v, \bar{\Sigma}_e, P_v, P_e)$ is a graph where node and arc attribute domains are random variables with values in domain $\Delta_v \cup \emptyset$ and $\Delta_e \cup \emptyset$ ($\emptyset$ represents a special null attribute addressed to indicate that the node can be null). Probabilities at the nodes and arcs of the FORG are related to the model used to model the random variable. In the case of the nodes $p_{\bar{v}_a}(x) = P[\bar{v}_a = x]$ represents the probability that random variable $\bar{v}_a$ takes value x. Equivalently, for the case of edges, $p_{\bar{e}_{a,b}}(x) = P[\bar{e}_{a,b} = x]$ indicates the probability that random variable $\bar{e}_{a,b}$ takes value x. A simple approach to model discrete random variables could be to represent this probability using a histogram where each position takes the number of times that value x has appeared in the training set. That is,

$$p_{\hat{v}_a}(x) = \frac{\sum_{i=1}^{P} \delta\left(\gamma_v\left(h^{i^{-1}}(\hat{v}_a)\right), x\right)}{P}, \tag{3.5}$$

where δ represent the Kronecker delta function.

Random variables on the edges are modeled in an equivalent form but conditioned that the terminal nodes exist.

A **Function-Described Graph** (FDG) (Serratosa, Alquézar et al. 2002; Serratosa, Alquézar et al. 2003) is a model graph addressed to improve the representational power of FORGs. It contains first-order probabilities of attributes and second-order structural information to describe a set of Attributed Graphs. The first order information is equivalent to the FORGs trough probability density functions. The second-order structural information is qualitative information that describes the joint probability of instantiating each pair of vertices or arcs. This information is represented by binary relations called *Antagonisms*, *Occurrences* and *Existences* between nodes and arcs. FDGs increased the representational power at the cost of increasing also the required physical space.

Two nodes or arcs are *antagonistic* if they have never taken place together in any graph used to synthesize the FDG although these two nodes or arcs are included in the FDG as different elementary parts. There is an *occurrence relation* between two nodes or arcs of the FDG if always that one of the related nodes or arcs in the graph has appeared; also the other node or arc of the same graph has appeared. Finally, there is an *existence* relation between two nodes or arcs if all the graphs in the class described by the FDG have at least one of the two nodes or arcs.

A **Second-Order Random Graph** (SORG) (Sanfeliu, Serratosa et al. 2004) is a probabilistic model closely related to FDGs. The main difference lies in the fact that the second-order structural information is not defined as binary relations but with the specific information of the second-order joint probability. Thus, the physical space needed to represent SORGs is much

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

higher than FDGs but also its ability to represent the set of Attributed Graphs. A Second-Order Random Graphs are defined similarly to FORGs but there are first order and also second order probability density functions.

3.1 Prototype synthesis

To construct any of the previously defined class prototypes or graph representatives, a common labeling among the graphs in the training set Γ is required. There are several methods to construct a graph prototype. The two main procedures are closely related to the common labeling construction methodologies.

Using a prototype oriented methodology to construct the prototype, an initial prototype is constructed using a single graph to later refine it adding some new information iteratively until all the training elements are analyzed. The second form corresponds to the Multiple-Isomorphism Oriented methods which are based on computing, in an initial phase, a common labeling without relying on the prototype effectiveness to represent data. The prototype is straightforwardly constructed on a second phase. Refer to section 2 for further details.

4. GRAPH DATABASES

It is well known that the main bottleneck of graphs based pattern recognition applications is the computational complexity of comparing two graphs. As a consequence, in practical applications, like the K-NN classifier where all training data is required to be compared with the input graph, may be prohibitive. To alleviate these problems, some attempts have been made to organize a set of graphs into a database. We differentiate between two techniques: techniques based on graph indexes and techniques based on trees.

Several techniques based indexes exist (Shasha, Wang et al. 2002; Yan, Yu et al. 2004). We emphasize the method developed in (Shasha, Wang et al. 2002) called GraphGrep. GraphGrep is based on a table in which each row stands for a path inside the graph (up to a threshold length) and each column stands for a graph. Each entry in the table is the number of occurrences of the path in the graph. Queries are processed in two phases. The filtering phase generates a set of candidate graphs for which the count of each path is at least that of the query. The verification phase verifies each candidate graph by assessing the sub-graph isomorphism between it and the query graph. Only sub-isomorphic graphs are returned as correct coincidences. Two years later, in 2004, Yan *et. al.* (Yan, Yu et al. 2004)

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

proposed GIndex that uses frequent patterns as indexing features. These frequent patterns reduce the indexing space as well as improve the filtering rate. The main drawback of these models is that the construction of the indices requires an exhaustive enumeration of the paths or fragments which increases the memory and time requirements. Moreover, since paths or fragments carry little information about a graph, the lost of information at the filtering step seems to be unavoidable.

With respect to tree organization of databases we highlight two methods. The first by Berretti *et. al. (Berretti, Bimbo et al. 2001)* in 2001. Attributed graphs were clustered hierarchically according to their mutual distances and indexed by m-trees. Queries are processed in a top-down manner by routing the query along the metric tree. Each node of the metric tree represents a cluster and it has one of the graphs of the cluster as a representative. The graph matching problem, in the tree construction and at query time, was solved by an extension of the A* algorithm that uses a look-ahead strategy plus a stopping threshold. The second method proposed by He and Singh (He and Singh 2006) in 2006 is called Closure-tree. It uses a similar structure than the one presented in (Berretti, Bimbo et al. 2001) but the representative of the cluster was not one of the graphs but a graph prototype called Closure Graph (see section 3) that could be seen as the union of the attributed graphs that compose the cluster.

In the next sections, we explain in detail the concept of metric trees and how they are used to speed up graph queries in a dataset. We focus on this method because, in chapter 6, it will be used to test how the common labeling concept can aid on the construction of the metric tree in graph databases.

4.1.1 Metric Trees

A metric-tree[3] (m-tree) (Ciaccia, Patella et al. 1997) is a method to partition a database in a hierarchical set of clusters, collecting similar objects. Each cluster contains a routing object and a radius providing an upper bound for the maximum distance between the reference object and any other object in the cluster.

More formally, a metric-tree, is a tree of nodes, each containing a fixed maximum number m of entries, $\langle node \rangle := \{entry\}^m$. In turn, each entry is constituted by a routing element M; a reference to the root r^M of a sub-index containing the element in the so-called covering region of M; and a radius d^M providing an upper bound for the distance between M and any element in its covering region, $\langle entry \rangle := \{M, r^M, d^M\}$. During retrieval, triangular

[3] Metric-tree uses distance d as a similarity measure. This distance d should fulfill the properties of a metric.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

inequality is used to support efficient processing of queries. To this end, the distance between a query element Q and any element in the covering region of a routing element M can be upper-bounded using the radius d^M and the distance between Q and M.

M-trees for vectorial data can be constructed using several different schemes. Each schema proposes specific methods to insert new elements or to select different routing elements. Following a static scheme, such as that proposed for mvp-trees (Bozkaya and Ozsoyoglu 1999), routing elements are selected when the entire database is determined. In this case, the m-tree is constructed in a top-down manner, by repeatedly partitioning the database through the selection of routing elements which yield a balanced split. Following an alternative approach, in (Ciaccia, Patella et al. 1997), the tree is constructed dynamically by inserting new elements from the bottom layer and promoting routing elements when insertion overflow occurs.

4.1.2 Similarity Queries on Metric Trees

One of the operations that can be speeded up, when structuring data in form of an m-tree, are range queries. These queries are addressed to return all elements, in the dataset, which their distances to a query graph Q are lower than a given threshold. To perform range queries in Metric Trees, the tree is analyzed in a top down fashion. Specifically, if d_{max} is the range of the query and Q is the query graph, the following conditions are employed, at each node of the tree, to check whether all the elements in the covering region of M, sub^M, can be discarded, accepted or need more exploration. The conditions are based on the evaluation of the distance between the routing element and the query element $d(Q, M)$. Several cases appear:

- If $d(Q, M) \geq d_{max} + d^M$, we reject all elements deeper from the routing element.
- If $d(Q, M) \leq d_{max} - d^M$ all the elements in the covering region of M can be accepted. That is, we accept all element in sub^M.
- In the critical case where neither of the above cases hold, the covering region of M, sub^M, may contain both acceptable and no acceptable elements, and the search must be extended deeper on the m-tree by explicitly exploring sub^M.

4.1.3 Nearest Neighbor Queries on Metric Trees

The k-nearest neighbor algorithm (*K*-NN) is one of the most used and simple method to classify objects based on a set of training examples. Usually, given a set of objects, that compose the knowledge of the system, the distance between the query object to all the objects in the knowledge

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

database is computed. Finally, the object is classified as belonging to the most common class among its *K* nearest neighbors. See that the systems rely on the distance measure to evaluate the similarity between the queried object and the objects in the knowledge database. These distance computations are deferred until classification time. K, usually, corresponds to a small positive integer ($K = \{1,3,5\}$) for the specific case of $K = 1$, the object is simply assigned to the class of its nearest neighbor.

Considering the training set as a knowledge database, structured in a m-tree fashion, the aim of the Nearest Neighbor Queries is to retrieve the K elements in a database that have minimum distance between them and the queried element. It is assumed that, at least, there are K elements in the database.

The proposed method to perform Nearest-Neighbor queries (Ciaccia, Patella et al. 1997) uses a branch-and-bound algorithm, which utilizes two global structures: a priority queue PR that stores the possibly fruitful tree nodes to be explored and an array NN that stores the best K elements found until the moment. At query time, the K values of NN are initialised to a null element. PR is initialised with one element which is the root of the metric tree. Note that PR does not have a maximum number of elements.

Let Q be a query element and $d_{max}^{NN}(Q)$ the maximum distance from Q to any element in NN. The distance $d_{max}^{NN}(Q)$ is initialized to infinity. In each iteration of the search algorithm, the tree node in PR with lower distance to Q is selected, let this node be named N and its children be $N^1, \dots, N^t$. The distances between Q and all the children of N must be computed. If son N^i is a **routing node**, this node is inserted in PR if

$$d(N^i, Q) - d^{N^i} \leq d_{max}^{NN}(Q) \tag{4.1}$$

This insertion is done due to it is possible to find an element with lower distance than the ones already found. On the contrary, if

$$d(N^i, Q) - d^{N^i} > d_{max}^{NN}(Q) \tag{4.2}$$

it is not possible that any of their descendants have a distance lower than $d_{max}^{NN}(Q)$ and so, none of the nodes and leaves of the branch will be further explored.

If son N^i is a **leave**, that is, a database element, and

$$d(N^i, Q) \leq d_{max}^{NN}(Q) \tag{4.3}$$

array NN and $d_{max}^{NN}(Q)$ are updated to consider this element in the following way. There are two possibilities:

- If all NN has its K positions full with leaves, then the element with higher distance is discarded. Moreover, the distance $d_{max}^{NN}(Q)$ is updated to be the maximum distance from Q to any element in NN. This new value has to be lower than the previous

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

value since $d_{max}^{NN}(Q)$ acts as a dynamic maximum search distance of range queries.

- If NN does not have its K positions full, then, the new tree leave N^i (that is, an element of the database) is inserted in an empty position of NN. Related to the distance $d_{max}^{NN}(Q)$, there are also two cases. If NN continues to be non full, then, the distance keeps its initial value, which is infinity. But, if the new situation of NN is that all the elements are used, $d_{max}^{NN}(Q)$ takes the maximum value of the distances between the leaves in NN and Q.

With respect the routing element M, it can be defined as one of the elements of the sub-cluster or a new element that represents the elements of the sub-cluster. The main effect of using a prototype instead of a representative is the theoretical reduction of the overlap between sub-clusters, due to the radius of the covering region should be more tightly adjusted.

Figure 4-1 and Figure 4-2 show an example where the same query is performed using the two types of routing nodes. In the example, Q is compared to a cluster that represents elements G^1, G^2 and G^3. In Figure 4-1 the cluster is represented with one of the elements in the cluster, G^2. In Figure 4-2 the cluster is represented with a new prototype. In the given example the cluster represented in Figure 4-1 must be explored due to $d(G^2,Q) - d^{G^2} \leq d_{max}^{NN}(Q)$. However, in Figure 4-2 the cluster radius is better adjusted and we can ensure that it will not have any desired element due to $d(M^3,Q) - d^{M^3} > d_{max}^{NN}(Q)$. Note that, in metric trees, the smaller the radius of clusters are, the lower the number of comparisons that we must perform to find the solution.

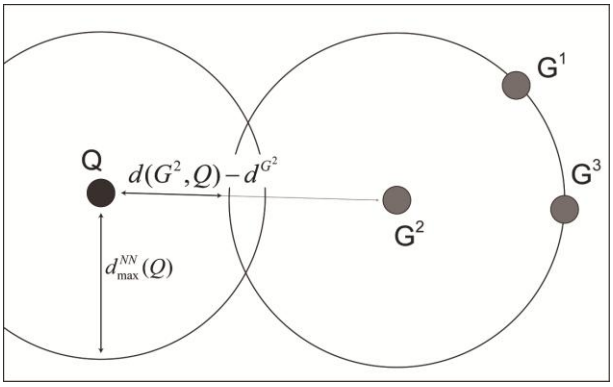


Figure 4-1: clusters represented by an element.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

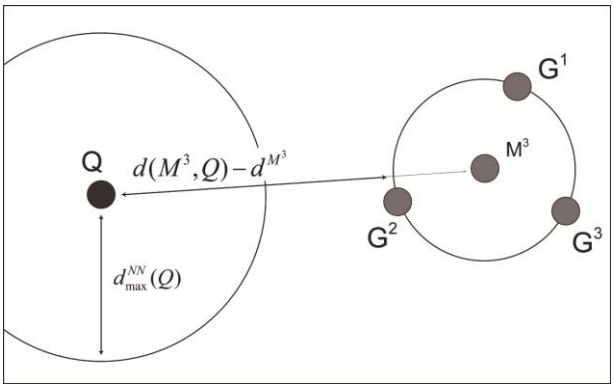


Figure 4-2: clusters represented by a computed prototype

4.1.4 Graph Databases based on metric distances

Considering the structure of the m-tree it is clear that the extension to a graph databases is not theoretically difficult. In this section, we introduce a general construction methodology from which one is able to construct a metric tree independently of the type of the routing element. That is, the structure of the m-tree and the clusters that are generated in the database are independent of the kind of routing node we use in the tree. This fact has only relative importance in vectorial databases but on graph databases may be fundamental since different graph prototypes can achieve different performance and so generate different clusters. Given a graph set Γ, it is crucial to obtain the same structure of the m-tree for all types of routing elements, since in the evaluation phase we want to compare the effectiveness of the common labeling to compute the prototype and not the prototype itself.

To synthesize the tree, we use a non-balanced tree constructed through a hierarchical clustering algorithm with complete linkage (Hastie, Tibshirani et al. 2009). Using this procedure, given a set of graphs, the distance matrix over the whole set is computed and then a dendogram is constructed. Using this dendogram and some horizontal cuts, a set of partitions, that clusters the database, is obtained. With these partitions the m-tree is generated. Finally, the information, M and d^M, on the routing elements in the m-tree is inserted.

In a graph M-Tree, M is corresponds to Graph Prototype (see section 3) and d^M is the maximum distance between the Prototype Graph and any of the graphs in the covering region sub^M. Figure 4-3 and Figure 4-4 show an example dendogram and its associated m-tree. The elements G^i are placed on the leaves of the dendogram and the routing elements M^j are placed on the junctions between the cuts and the horizontal lines of the dendogram.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

4.1.5 Evaluation measures for graph metric trees

To evaluate the metric trees and the queries performed over them three classical indices will be used: Overlap, Access ratio and F-measure.

Overlap: this index evaluates the quality of the tree itself, without the need of performing queries on it. We want the tree nodes to be the most discriminative possible, for this reason, the lower is the overlap between the covering regions of sibling nodes, the higher is the quality of the m-tree since nodes are more discriminative and therefore the time to compute the query reduces.

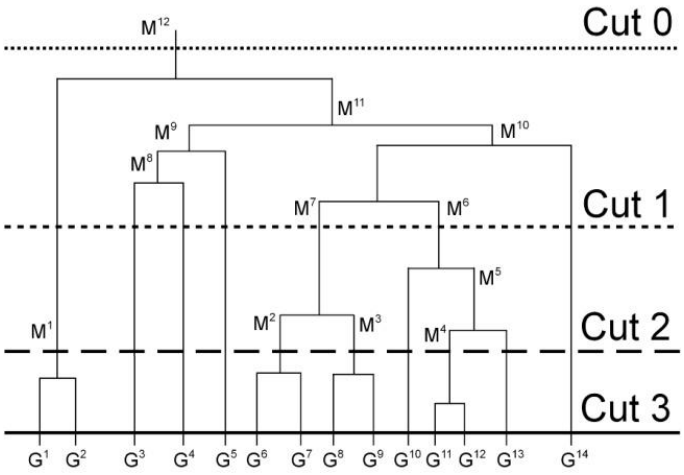


Figure 4-3: example of a dendogram.

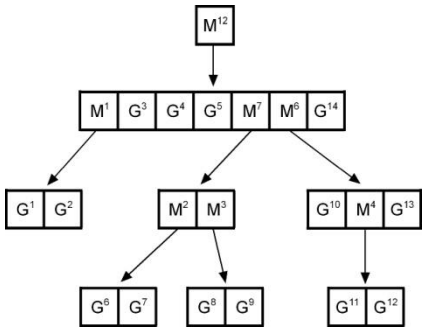


Figure 4-4: m-tree obtained with dendogram of Figure 4-3.

Given two sibling nodes M^i and M^j, the overlap of their covering regions is defined as follows,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$S(M^i, M^j) = \begin{cases} \dfrac{d^{M^i} + d^{M^i}}{d(M^i, M^j)} & if \dfrac{d^{M^i} + d^{M^i}}{d(M^i, M^j)} > 1 \\ 0 & otherwise \end{cases} \quad (4.4)$$

Where $d(M^i, M^j)$ is any distance between the prototype or Attributed Graphs. Note that M^i and M^j can represent a prototype or an Attributed Graph. Recall that the covering region of a node that is an Attributed Graph is 0.

Given a node of the m-tree that is not a leave, M, their own overlap is computed as the normalized overlap between their children. The radius of the sub-clusters that the children represent is obtained from the parameter d^M in their m-tree nodes.

$$S(M) = \frac{1}{\binom{C_M}{2}} \sum_{i=1}^{C_M} \sum_{j=i+1}^{C_M} S(M^i, M^j) \quad (4.5)$$

where C_M is the number of sons of the m-tree node M and $M^1, \dots, M^{C_M}$ are its node sons. The own overlap $S(M)$ of a leave is not defined since it is not needed. Finally, the general overlap of an m-tree is computed as,

$$Overlap(T) = \frac{1}{C_T} \sum_{i=1}^{C_T} S(M^i) \quad (4.6)$$

where C_T is the number of nodes (without considering the leaves) of the m-tree node T.

Access ratio: This index evaluates the capacity of the m-tree to properly route the queries. Given a query graph Q, this index is the number of accessed nodes and leaves of the m-tree or the number of graph matching operations performed, *A*. Finally, it is normalized by the number of graphs used to generate the m-tree, N. If the Access ratio is higher than 1, then it is faster not to use an m-tree since without the m-tree, the system would perform less comparisons.

$$AccessRatio(Q, d_{max}, T) = \frac{A}{N} \quad (4.7)$$

F-measure: The F-measure is a measure of a test's accuracy computed through the Precision and Recall. In the field of information retrieval, Precision is the fraction of retrieved elements that are relevant to the search. And Recall is the fraction of the elements that are relevant to the query that are successfully retrieved. We obtain these two metrics as follows. We compute a query with a graph Q and a range d_{max}. The m-tree returns the W graphs that the distance between them and Q is lower than d_{max}. Moreover, we compute the distance between the graph Q and all the graphs in the dataset (without using the m-tree) and obtain the S elements with the

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

minimum distance. Finally, we count the number of relevant elements, C, that appear in both sets. Therefore,

$$Precision(Q, d_{max}, T) = \frac{C}{W}, Recall(Q, d_{max}, T) = \frac{C}{S} \quad (4.8)$$

$$F - measure(Q, d_{max}, T) = \frac{2C}{S + W} \quad (4.9)$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 3

MODELING THE GRAPH EDIT DISTANCE THROUGH EDIT SURFACES

1. GRAPH EDIT DITANCE

Using definition of the Graph Edit Distance given in chapter 2, the tailoring of this distance to the application at hand essentially depends on C_{ns}, C_{nd}, C_{ni}, C_{es}, C_{ed} and C_{ei} functions. Several definitions of these functions exist in the literature. We focus first on the definition of functions C_{ns} and C_{es}. The most common approaches are the following. The first and simplest approach considers cost $C_{ns} \in \{0, K_{ns}\}$ where $C_{ns}(v_a^p, v_i^q) = K_{ns}$ if $d\left(\gamma_v(v_a^p), \gamma_v(v_i^q)\right) > Threshold$ otherwise $C_{ns} = 0$, d is defined as a distance function over the domain of the attributes. Specific examples of this cost can be found in fingerprint verification (Jain and Maltoni 2003) where $C_{ns} \in \{0,1\}$ or in (Bunke 1998; Bunke 1999). The second and most frequently used approach corresponds to the case where $C_{ns}\left(v_a^p, v_b^q, \theta_v\right) \in \mathbb{R}$. In this case, node substitution cost depends on the attributes of the nodes and possibly on some other parameters θ_v as shown in (Neuhaus and Bunke 2006), (Lladós, Martí et al. 2001) and (Caetano, McAuley et al. 2009), among others. Similar approaches can be used to define C_{es}. With regard to C_{nd}, C_{ni}, C_{ed} and C_{ei}, these functions usually simply assign a constant cost. However, in particular models they can also depend on node or edge attributes (Wong and You 1985; Serratosa, Alquézar et al. 2003; Sanfeliu, Serratosa et al. 2004).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Nodes and edges can be mapped, by functions γ_v and γ_e, to several types of data: nominal, ordinal or modulo. Depending on the data type a particular distance function is required (see section 3 of (Serratosa and Sanfeliu 2006)).

Several specific joint definitions for C_{ns}, C_{nd}, C_{ni}, C_{es}, C_{ed} and C_{ei} functions have been theoretically studied. We highlight (Bunke 1998) and (Bunke 1999) which are described in Table 1-1.

Reference	C_{ns}	C_{nd}	C_{ni}	C_{es}	C_{ed}/C_{ei}
(Bunke 1998), (Bunke 1999)	$\{0,\infty\}$	$\{1\}$	$\{1\}$	$\{0,\infty\}$	$\{0\}$
(Bunke 1999)	$d(v_a^p, v_i^q, \theta_v) \in \mathbb{R}$	$\{K_{nd}\}$	$\{K_{ni}\}$	$d(e_{ab}^p, v_{ij}^q, \theta_e) \in \mathbb{R}$	$\{0\}$

Table 1-1: graph edit cost defined in (Bunke 1998) and (Bunke 1999).

The specific cases studied in (Bunke 1999) and (Bunke 1998) yield to several interesting properties. The costs of first row of Table 1-1 relate the graph edit distance with the maximal common sub-graph. In this way, computing the graph edit distance with these specific costs leads to the computation of the maximal common sub-graph. The cost given in the second row has been studied in (Bunke 1999). Note that the cost of inserting and deleting an edge is always considered zero. In the definition of (Bunke 1999) authors assume that graphs are complete graphs and a non-existing edge is an edge with a *"null"* label. In this case, the cost of deleting and inserting and edge can be encoded in the edge substitution cost. With this definition authors describe several classes of costs that optimize at the same final labeling. In this chapter, we follow the same direction and give a deepest characterization of these classes of costs. To this aim, we slightly modify the graph edit distance definition of Table 1-1 (second row). The new definition, we propose, is given in Table 1-2.

C_{ns}	C_{nd}	C_{ni}	C_{es}	C_{ed}	C_{ei}
$d(v_a^p, v_i^q, \theta_v) \in \mathbb{R}$	K_n	K_n	$d(e_{ab}^p, v_{ij}^q, \theta_e) \in \mathbb{R}$	K_e	K_e

Table 1-2: particularization of Graph Edit Distance

Note that our definition is able to codify the same information. However, edge insertion and edge deletions are considered in a separate cost function. Besides, we impose the requirement that $C_{nd}(v) = C_{ni}(v) = K_n, \forall v$ and $C_{ed}(e) = C_{ei}(e) = K_e, \forall e$. This requirement is necessary for our development and, moreover, for the graph edit distance to fulfill the symmetric property of a distance.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2. CLASS OF COSTS AND EDITSURFACE

A Labeling Space is a 2-dimensional Euclidean space where the coordinates correspond to the graph edit insertion and deletion costs. Given a pair of graphs, we can select some regions in this space such that all points in the labeling space obtain the same optimal labeling (the labeling that obtains the graph edit distance). We call each region as **Class of Costs**. Moreover, given the labeling space and two graphs, we can define a function defined over all the labeling space where its value in each point is the distance value between both graphs. We call this function **Edit Surface**.

In this section, we first give some basic definitions and then we present two properties of the class of costs and two properties of the edit surface. From now to the rest of the chapter, we use the particular specification of the Graph Edit Distance given in Table 1-2. Therefore, the labeling space is a bi-dimensional space with the axis K_n and K_e.

2.1 Specific and complementary definitions

2.1.1 Definition 2-1: Edit Cost

Given two graphs, G^p and G^q, a bijection $f \in T$ between them and two constant values $(K_n, K_e) \in \mathbb{R}^{2^+}$, the graph edit cost is given by:

$$EditCost_{K_n,K_e}(G^p, G^q, f) = a_f K_n + b_f K_e + c_f \tag{2.1}$$

a_f refers to the number of inserted and deleted nodes and can be computed as:

$$a_f = \sum_{\{\forall\, v_a^p \in \Sigma_v^p - \widehat{\Sigma}_v^p\}} \sum_{\{\forall\, v_i^q \in \widehat{\Sigma}_v^q\}} H_v\left(v_a^p, v_i^q\right) + \sum_{\{\forall\, v_a^p \in \widehat{\Sigma}_v^p\}} \sum_{\{\forall\, v_i^q \in \Sigma_v^q - \widehat{\Sigma}_v^q\}} H_v\left(v_a^p, v_i^q\right) \tag{2.2}$$

b_f refers to the number of inserted and deleted edges and can be computed as:

$$b_f = \sum_{\{\forall\, e_{ab}^p \in \Sigma_e^p - \widehat{\Sigma}_e^p\}} \sum_{\{\forall\, e_{ij}^q \in \widehat{\Sigma}_e^q\}} H_e\left(e_{ab}^p, e_{ij}^q\right) + \sum_{\{\forall\, e_{ab}^p \in \widehat{\Sigma}_e^p\}} \sum_{\{\forall\, e_{ij}^q \in \Sigma_e^q - \widehat{\Sigma}_e^q\}} H_e\left(e_{ab}^p, e_{ij}^q\right) \tag{2.3}$$

c_f refers to the cost of substituting nodes and edges, this last cost can be computed as:

$$\begin{aligned} c_f = & \\ = & \sum_{\{\forall\, v_a^p \in \Sigma_v^p - \widehat{\Sigma}_v^p\}} \sum_{\{\forall\, v_i^q \in \Sigma_v^q - \widehat{\Sigma}_v^q\}} H_v\left(v_a^p, v_i^q\right) * d(v_a^p, v_i^q, \theta_v) + \\ + & \sum_{\{\forall\, e_{ab}^p \in \Sigma_e^p - \widehat{\Sigma}_e^p\}} \sum_{\{\forall\, e_{ij}^q \in \Sigma_e^q - \widehat{\Sigma}_e^q\}} H_e\left(e_{ab}^p, e_{ij}^q\right) * d(e_{ab}^p, e_{ij}^q, \theta_e) \end{aligned} \tag{2.4}$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$H_v(v_a^p, v_i^q)$ and $H_e(e_{ab}^p, e_{ij}^q)$ are computed as:

$$H_v(v_a^p, v_i^q) = \begin{cases} 1 & if\ f(v_a^p) = v_i^q \\ 0 & otherwise \end{cases}; H_e(e_{ab}^p, e_{ij}^q) = \begin{cases} 1 & if\ f_e(e_{ab}^p) = e_{ij}^q \\ 0 & otherwise \end{cases} \quad (2.5)$$

and identify the matchings given by bijection f.

Using this particular definition, the graph edit cost function can be represented in a 3-dimensional space where x-axis corresponds to K_n, y-axis corresponds to K_e and z-axis corresponds to $EditCost$. Note that $EditCost$ depends linearly on K_n and K_e.

2.1.2 Definition 2-2: Edit Distance

Given two graphs G^p and G^q and two constant values $(K_n, K_e) \in \mathbb{R}^{2^+}$, the Graph Edit Distance is defined as:

$$\begin{aligned} EditDistance_{K_n,K_e}(G^p, G^q) &= \min_{f \in T} EditCost_{K_n,K_e}(G^p, G^q, f) \\ &= \min_{f \in T} a_f K_n + b_f K_e + c_f \end{aligned} \quad (2.6)$$

In other words, the Graph Edit Distance is the minimum cost that can be obtained for particular values of K_n and K_e.

2.1.3 Definition 2-3: Class of Cost

Given two graphs, G^p and G^q, and a bijection $f \in T$ between them, a class of cost $\Omega^{p,q}(f)$ is the sub-set of values in $\mathbb{R}^{2^+}$ for which f is the bijection whereby the minimum graph edit cost is obtained,

$$\Omega^{p,q}(f) = \left\{ (K_n, K_e) \in \mathbb{R}^{2^+} \middle| f = \operatorname*{argmin}_{f' \in T} a_{f'} K_n + b_{f'} K_e + c_{f'} \right\} \quad (2.7)$$

We write $\Omega(f)$ instead of $\Omega^{p,q}(f)$ when no confusion is possible. We designate the set of all classes of cost given to graphs G^p and G^q by $\widehat{\Omega}^{p,q}$.

2.1.4 Definition 2-4: Edit Surface

Given two graphs, G^p and G^q, and a bi-dimensional space composed of values (K_n, K_e) in $\mathbb{R}^{2^+}$, we define the Edit Surface as,

$$\begin{aligned} &EditSurface_{G^p,G^q}: \mathbb{R}^{2^+} \to \mathbb{R}^+ \\ &EditSurface_{G^p,G^q}(K_n, K_e) = EditDistance_{K_n,K_e}(G^p, G^q) = \min_{f \in T} a_f K_n + b_f K_e + c_f \end{aligned} \quad (2.8)$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2.2 Properties of the Class of Costs

Property 2-1. Given two graphs, G^p and G^q, any class of cost $\Omega(f)$ is either empty or its values form a convex polygon in the bi-dimensional space composed of $(K_n, K_e) \in \mathbb{R}^{2^+}$.□

Given two graphs, G^p and G^q, and a labeling f. We see that for f to yield the Graph Edit Distance at a concrete point (K_n, K_e), its cost must be less or equal than the cost which can be obtained with any other labeling $f' \in T$. That is, the following system of inequalities must hold:

$$a_f K_n + b_f K_e + c_f \leq a_{f'} K_n + b_{f'} K_e + c_{f'}, \forall\, f' \in (T - f) \tag{2.9}$$

Each of the above inequalities (one for each f') divides $\mathbb{R}^{2^+}$into two parts by means of a linear equation. It is known that the intersection of any finite set of linear inequalities is a convex polygon (Grunbaum 2003). Consequently, each optimal labeling appears only in a single convex polygon.

Property 2-2. Given two graphs, G^p and G^q, and a class of costs $\Omega^{p,q}(f)$, any class of costs $\Omega^{p,q}(f')$ where $f' \in (T - f)$, $a_f = a_{f'}$, $b_f = b_{f'}$ and $c_f = c_f$ is optimal at the same set of points as $\Omega(f)$.□

This property is easily deduced through equation (2.9).

Note that Property 2-2 implies that the graph edit cost is not an injective function due to several labeling can give the same optimal cost.

Discussion of property 2-1 and 2-2

Using Property 2-1, we see that $\Omega(f)$, $\forall\, f$ optimal, tessellates $\mathbb{R}^{2^+}$ with convex polygons. Each polygon defines a class of costs $\Omega(f)$. A class forms a convex polygon with finite area if its values of K_n and K_e are finite. Otherwise, the area is infinite.

Figure 2-2 and Figure 2-3 show an example of Property 2-1. Figure 2-1 shows two graphs of the Letter dataset (Riesen and Bunke 2008). Examples correspond to graph 35 and 72 of class *A*. Figure 2-2 and Figure 2-3 show how two labelings are described by the intersection of a set of inequalities, each line corresponding to a concrete inequality of (2.9). Figure 2-2 shows a finite class of costs and Figure 2-3 shows an infinite class of cost.

Note that the above formulation allows dividing $\mathbb{R}^{2^+}$ into convex polygons, each of which corresponds to an optimal labeling. Note also that it is possible given (2.9) to produce an empty intersection, in this case the tested labeling f is never optimal at any (K_n, K_e).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Knowing that the labeling space is tessellated with labelings, it is interesting to see how these labelings tend to be distributed and their relation with the values and meaning of K_n and K_e, specially for the extreme values of $(K_n, K_e) \in \{(0,0), (K_n, \infty), (\infty, K_n), (\infty, \infty)\}$.

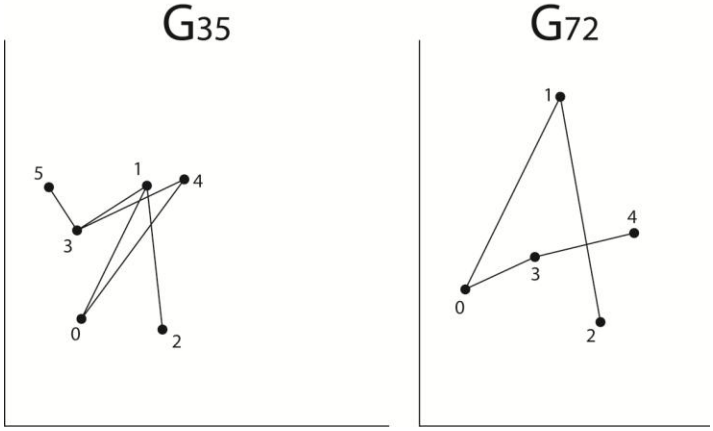


Figure 2-1: two graphs.

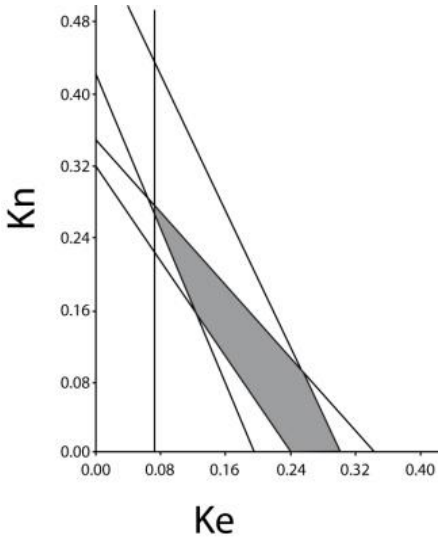


Figure 2-2: example of a finite area

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

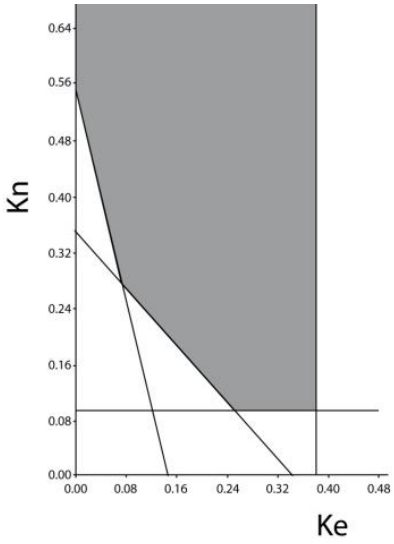


Figure 2-3: example of infinite area

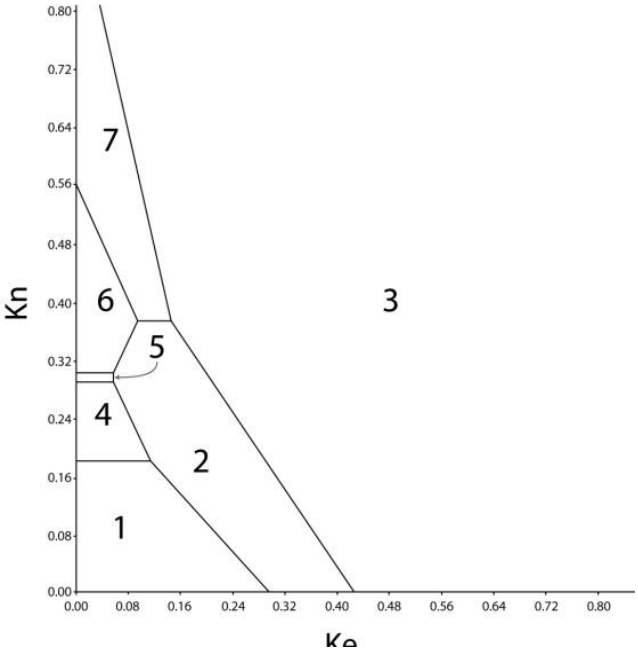


Figure 2-4: diagram of classes of cost.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

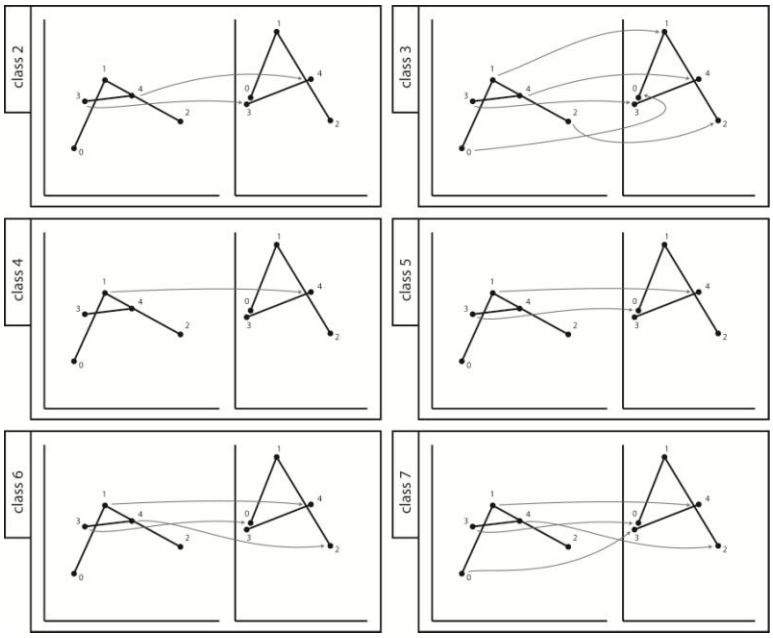


Figure 2-5: labelings related to of classes of cost.

Consider, as an example, graphs G^0 and G^{10} of Letter dataset (Riesen and Bunke 2008) class A. Figure 2-4 presents the classes of cost in the range of $K_n, K_e \in [0, 0.8]$ for these graphs computed using an optimal graph matching algorithm. The vertical axis corresponds to K_n values and K_e values are shown in the horizontal axis. In addition, Figure 2-5 shows the labelings that each class of cost produces. We first analyze the labelings computed at $(K_n, K_e) = (0,0)$. At this special point every node insertion has a cost of zero. Therefore, from the node point of view, the less costly assignation is to delete all nodes of the first graph and insert all nodes of the second graph. From the edge point of view, note that, if we assign all nodes to null the edges will be either substituted if the edge was not initially in the graph or deleted if the edge was on the graph; in both cases the edges cost will be zeros. Consequently, we can ensure that at point $(K_n, K_e) = (0,0)$ the resulting distance between both graphs will be zero, either because all nodes from both graphs will be assigned to null nodes of the other graph or because both graphs are isomorphic. Analyzing labelings attached to the vertical axis, that is $(K_n, K_e) \in (0..\infty, 0)$, it is clear that from high to low values of K_n labelings associated with each class (Figure 2-5) go from substituting all nodes ($\Omega_3^{0,10}$) to only performing insertions and deletions ($\Omega_1^{0,10}$). However, an interesting fact is that not all node substitutions are sub-contained in the class of costs with lower K_n. We see that this happens in some classes

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$\Omega_1^{0,10} \subset \Omega_4^{0,10} \subset \Omega_5^{0,10} \subset \Omega_6^{0,10} \subset \Omega_7^{0,10}$but not in others $\Omega_3^{0,10}$. Note that even if insertion and deletion of edges is not considered, edge substitution it is. In the special case where $K_e = 0$, notice that if edges do not have attributes (that is $C_{es}(\cdot) = 0$) the problem of computing the Graph Edit Distance turns from the quadratic assignment problem to the linear assignment problem. An interesting special value when moving over the K_n axis is the value of $(K_n, K_e) = (\infty, 0)$. The labeling computed using this value maximizes assignation from nodes of the first graph to nodes of the second graph considering the minimum number of null assignations required which is $|\mathrm{R}^{\mathrm{p}} - \mathrm{R}^{\mathrm{q}}|$. We now consider the labelings we obtain when moving over the K_e axis. That is, we consider values $(K_n, K_e) \in (0, 0..\infty)$. It is obvious that different values of K_e force the result to be more structurally correct. However, forcing in addition $K_n = 0$, it does not necessarily mean that the node attributes are not considered. In fact they are, due to node substitution cost is considered. In the example of Figure 2-4, see that as we move K_e towards $+\infty$, the classes change to force the labelings to be more structurally consistent. Note again how node substitutions are sometimes not sub-contained in adjacent classes, e.g. $\Omega_6^{0,10} \subset \Omega_7^{0,10} \not\subset \Omega_3^{0,10}$. In the extreme case $(K_n, K_e) = (0, \infty)$, we can affirm that the resulting optimal correspondence, if enough null nodes are provided and edges do not have attributes, corresponds to the maximal common sub-graph as demonstrated in (Bunke 1999). If we aim to obtain the maximal common sub-graph when attributes are present in edges, the edge substitution cost must restrict edges to have the same attribute and so the edge substitution cost must be defined as Table 1-1 row 1. The final extreme value to analyze corresponds to $(K_n, K_e) = (\infty, \infty)$. In most of the cases, while using these costs, the resulting labeling maximizes the node substitutions and edge substitutions at the same time. However, this double maximization can be troublesome in several cases. Considering this issue, we differentiate between two types of $\widehat{\Omega}$ (Definition 2-3) sets. The first corresponds to graphs where for value $(K_n, K_e) = (\infty, 0)$ the optimal labeling is equivalent to the optimal labeling for value $(K_n, K_e) = (0, \infty)$. That is, the structurally optimal and the semantically optimal labelings are equivalent. This is shown in Figure 2-4. We consider this situation to be the desired case when applying graph matching to pattern recognition. Two similar objects, compared under the optimal labeling, should maximize structural and syntactical relations at the same labeling. The second type of labeling spaces corresponds to functions in which the optimal semantic labeling differs from optimal structural labeling. An example is shown in Figure 2-6 and Figure 2-7 which show graphs G^{35} and G^{72} of the letter dataset (Riesen and Bunke 2008) of class A. See that optimal labelings when $(K_n, K_e) = (\infty, 0)$ and $(K_n, K_e) = (0, \infty)$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

differ. When this situation occurs in the application at hand, we must decide which labelings we prefer to optimize, structural or semantic.

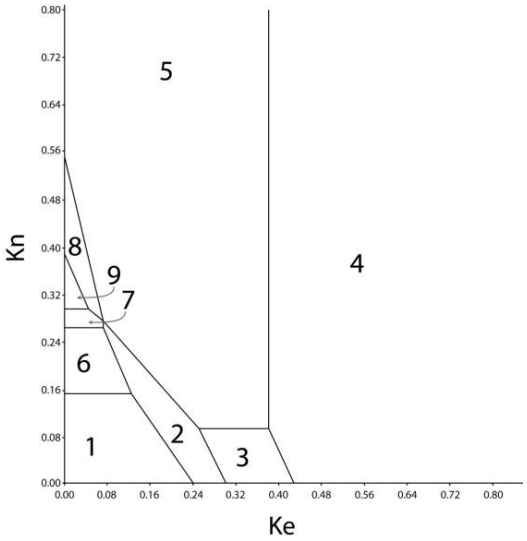


Figure 2-6: diagram of classes of cost.

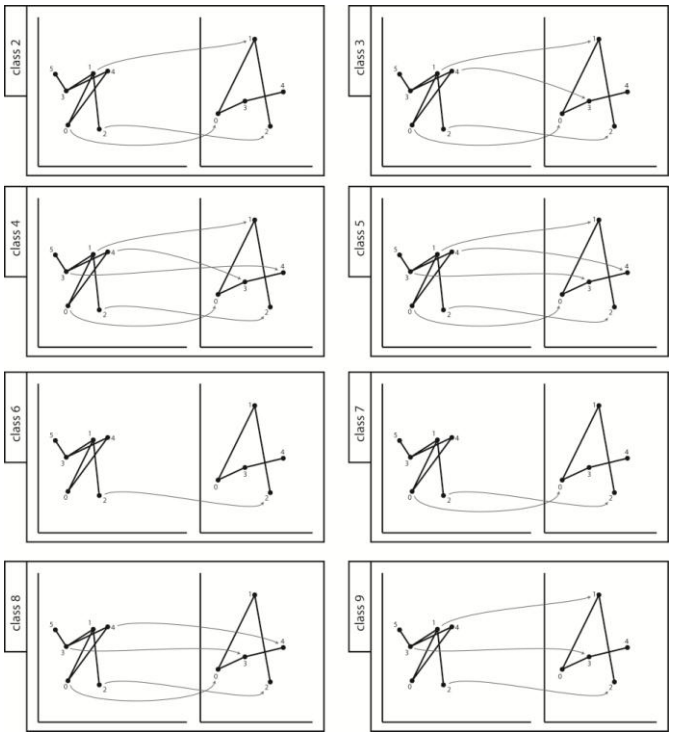


Figure 2-7: labelings related to of classes of cost.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2.3 Property of the Edit Surface

Property 2-3. Given two graphs, G^p and G^q, the function $EditSurface_{G^p,G^q}$ monotonically increases. In other words,

$$EditSurface_{G^p,G^q}(p_1) \leq EditSurface_{G^p,G^q}(p_2) \qquad (2.10)$$

where $p_1 = (K_n^1, K_e^1)$, $p_2 = (K_n^2, K_e^2)$, $K_n^1 \leq K_n^2$and $K_e^1 \leq K_e^2$.□

We know from Property 2-1 that the (K_n, K_e) bi-dimensional space can be divided into several classes of cost, $\Omega(f_{1..m})$. Each class of cost $\Omega(f_i)$ is represented by its plane equation (2.7). We know from Definition 2-1 that values a_{f_i}, b_{f_i} and c_{f_i} are positive. Thus, we can conclude that within each class of cost $\Omega(f_i)$, costs monotonically increase.

It is important to see that where two classes of cost intersect, costs do not decrease but remain equal or increase. Two labelings change their optimality when costs for both classes are equal, that is, when equation (2.9) for two different labelings,f and f', is equal. Using this min operator ($\leq$), cost cannot decrease and so when two classes intersect, costs keep increasing.

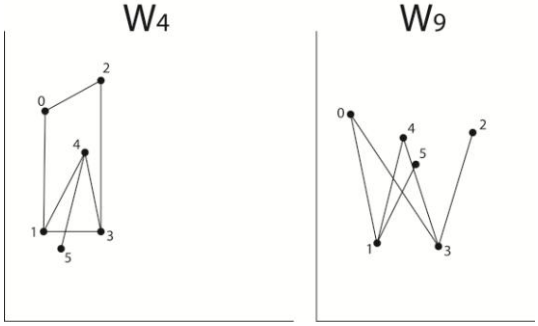


Figure 2-8: two graphs.

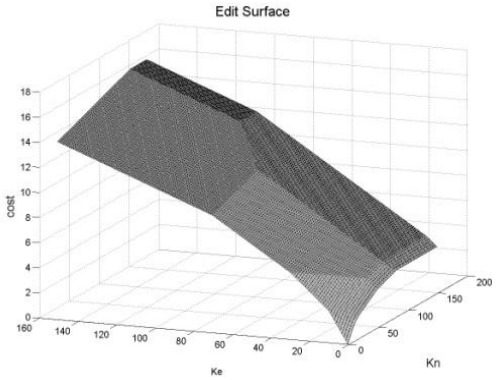


Figure 2-9: example of Edit Surface.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Property 2-4. Given two graphs G^p and G^q and three points, $p_1 = (K_n^1, K_e^1)$ and $p_2 = (K_n^2, K_e^2)$ and $p_3 = (K_n^3, K_e^3)$ over the edit surface and vertical plane $sin(\theta)K_n + \cos(\theta)\, K_e = C$ where $K_n^1 \leq K_n^2$, $K_e^1 \leq K_e^2$, $K_n^2 \leq K_n^3$ and $K_e^2 \leq K_e^3$. $\forall \theta \in [\pi/2\,, \pi]$ and $C \in \mathbb{R} \Rightarrow$

$$m_1 = \frac{\left|EditDistance_{p_2}(G^p, G^q) - EditDistance_{p_1}(G^p, G^q)\right|}{\|\mathrm{p}_2 - \mathrm{p}_1\|}$$
$$\geq$$
$$m_2 = \frac{\left|EditDistance_{p_3}(G^p, G^q) - EditDistance_{p_2}(G^p, G^q)\right|}{\|\mathrm{p}_3 - \mathrm{p}_2\|} \geq 0$$

That is, the slope of l decreases as we move towards infinity. See Figure 2-10.

We know from Property 2-1 that surface generated by $EditDistance(G^p, G^q, K_n, K_e)$ is a composition of planes given by (2.9). Each of those planes has the property that given any two points $p_1 = (K_n^1, K_e^1)$ and $p_2 = (K_n^2, K_e^2)$ where $K_n^1 \leq K_n^2$, $K_e^1 \leq K_e^2$ slope m_1 is positive or zero. This is consequence of being $a, b, c \geq 0$.

In this way, given two planes $\Pi^{f_1^{p,q}}$ and $\Pi^{f_2^{p,q}}$ generated using (2.6). We compute the intersection with the plane $sin(\theta)K_n + \cos(\theta)\, K_e + C = 0$ giving as a result two lines l_1 and l_2 (see Figure 2-11, Figure 2-12 and Figure 2-13). Using operator $l = min(l_1, l_2)$ we reduce these two lines to a two dimensional function. We distinguish two cases: (Figure 2-11) l_1 and l_2 cross at some point $K_n > 0$ and $K_e > 0$ and (Figure 2-12, Figure 2-13) do not cross. For the first case, m_2 must be lower than m_1 otherwise lines cannot cross, see Figure 2-11. For the second case we distinguish between l_1 and l_2 cross at some point $K_n \leq 0$ and $K_e \leq 0$ (Figure 2-12) or are parallel (Figure 2-13), in both cases the slope is kept constant.

For the case of n planes $\Pi^{f_{1\ldots n}^{p,q}}$ an exponential number of combinations appear when ordering n lines. Using the above demonstration, we can affirm that between two lines the one with small slope will become optimal at some point. Thus, given n lines, studying the surroundings of every intersection we can reduce the problem to the two lines case. See Figure 2-14.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

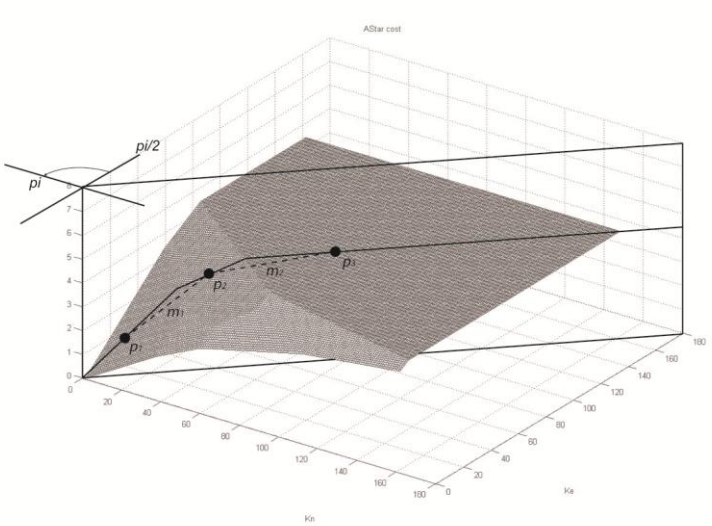


Figure 2-10: illustrative example of property 4.

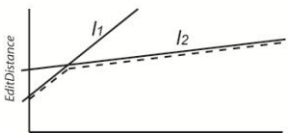


Figure 2-11: lines intersect.

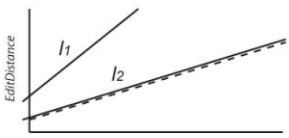


Figure 2-12: lines do not intersect.

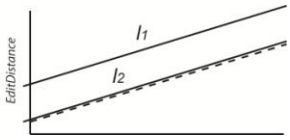


Figure 2-13: lines are parallel.

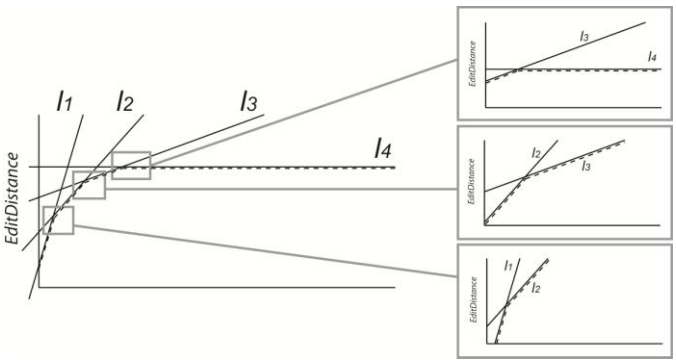


Figure 2-14: reduction of n lines case to the pair-wise case.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

3. APPLICATIONS

In this section, we illustrate how the given properties can assist some graph matching problems. We study three applications where the described properties can be used.

3.1 Interactive and Adaptive Graph Recognition

The aim of the application we present is to learn a model that represents a set of graphs such that the labeling obtained by the graph matching algorithm is as similar as possible to the labeling between graph nodes imposed by a human expert.

In most applications, the labeling between nodes is only partially considered. This is because it is considered in the first stages of the pattern recognition process, in which it is desired to find a similarity measure between graphs. But, when this similarity value is obtained (the final distance value between graphs given the labeling), the knowledge of the labeling is not further considered. Nevertheless, we consider that although the graph is properly classified or identified, the result of the comparison (the final distance value) has not sense if the labeling between their local parts is far from the labeling proposed by the human specialist.

In (Serratosa, Solé-Ribalta et al. 2011) it is defined an interactive and adaptive graph recognition model with the aim of increasing the quality of the labeling between the graph to be identified and the reference graphs of the database. To that aim, the graph recognition model is extended to consider the labeling between nodes proposed by a human specialist. This new knowledge is incorporated into the system and used to modify the weights of the model (such as K_n and K_e) that tune the similarity function between graphs.

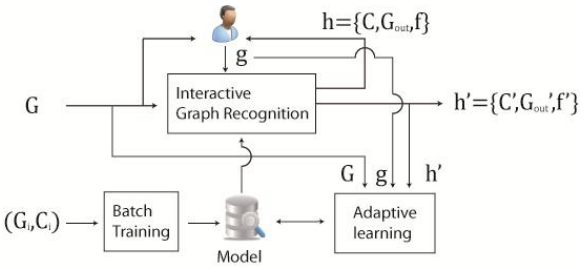


Figure 3-1: scheme of the Interactive and Adaptive Graph Recognition Model.

The batch training process of the application, shown in Figure 3-1, generates the first knowledge of the system that forms the model given a set

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

(graph, class) pairs and other parameters, such as K_n and K_e. The Interactive Recognition process generates a first hypothesis $h = \{C, G_{out}, f\}$ given an input graph G and using the model. This hypothesis is composed by a class C, a graph with the minimum distance G_{out} and a labeling f between both graphs. When the human proposes a new labeling g, the interactive process generates the final hypothesis h' using the model and also the human interaction g. Note that, h' can be completely different from h. Not only the graph and the labeling can be different but also the class. Finally, the Adaptive Learning module updates the parameters in the model. Specifically, two of these parameters are K_n and K_e. Computing the new values of these parameters is a process related to the aim of the chapter since the new values are obtained through the labeling space. Moreover, the proposed algorithm needs Property 2-1 to perform properly.

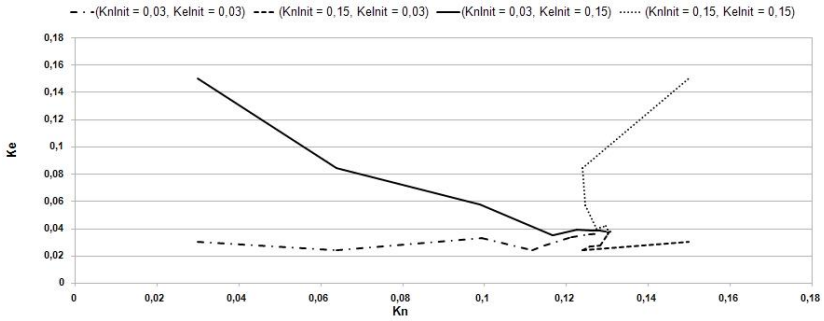


Figure 3-2: evolution of K_n and K_e in four different initialisations.

The inputs of the algorithm used to update K_n and K_e are the input graph G, the labeling imposed by the user g and the final output graph G'_{out}. The outputs of the algorithm are the new values of K_n and K_e which maximize the labelings proposed by the expert. The main idea of the algorithm is to build a histogram of the classes of costs that appeared while using the general model. That is, each time the algorithm receives a new input, the class of costs is obtained, $\Omega^{G,G'_{out}}(g)$, and added to the histogram. Thus, the peak of the histogram (there could be several peaks) represents the values of K_n and K_e that maximize the similarity of the human labelings with the labelings proposed by the system.

Figure 3-2 (reprinted from (Serratosa, Solé-Ribalta et al. 2011)) shows the evolution of values of K_n and K_e when new graphs are added into the system. The system has been initialized with four different values of K_n and K_e. The initial values are the most external points. We see that when the algorithm converges, the four experiments converge to the same final value of K_n and K_e.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

3.2 Analysis of the behavior of human similarity measures

Given a pair of images that represent objects (pictures, handwritten characters...), humans can decide if they are similar or not or even they can decide certain degree of similarity. This is because we have an inherent similarity function (difficult to be mathematically defined) that may be application dependent. When we aim to solve the problem of automatically describing this similarity through automatic structural pattern recognition, it is usual to convert these images to graphs and apply a distance measure between graphs.

Graph Edit Distance has some application dependent weights that can be manually tuned. Some research has been done to automatically obtain these weights such that the overall recognition ratio is maximized given a database (Neuhaus and Bunke 2006; Neuhaus and Bunke 2007) or the difference between the node bijection between both graphs imposed by the specialist and the node bijection obtained by the machine is reduced (Serratosa, Solé-Ribalta et al. 2011). If we have enough theoretical information to understand the behavior about the graph distance at hand, it is possible to go a step further. It is possible to investigate if the inherent distance measure between nodes or between arcs that the user has, given an application, approaches the one that the method defines. Property 2-2 states that there could be two different labelings between nodes that are optimal at the same class of costs. These two labelings can be seen as different interpretation of the representation.

Suppose we want to compare pictures and we have extracted a region adjacency graph from each image. A region adjacency graph is a graph in which nodes represent important regions of the images. The attribute of the regions may be the average color, the area, the circularity of the region and so on. There is an arc between two nodes if regions are adjacent. There could be some attributes on the arcs such as the length of the border between regions. Suppose we compute the cost class given the labeling imposed by a human specialist for all the graph comparisons.

Then, some different situations can happen:

- The average value of K_n of all the cost class is low given all the graph comparisons. In this case, the specialist believes the semantic information on the nodes is very important. Therefore, the specialist considers two images are similar if they have similar regions but independently of the position of these regions.
- The average value of K_e of all the cost class is low given all the graph comparisons. This case is the opposite of the last one. The user believes the most important aspect while comparing two

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

images is the relation between regions (i.e. their relative position) although these regions seem to be very different (i.e. different area or color)

- The area of the union of the cost classes of all the comparisons is big although the intersection is small. This means that the specialist has different perceptions of the importance of the relations (arcs) respect the semantic information (nodes) depending on the images. This situation can happen when the system is trained by different specialist.
- An opposite case appears when the area of the union and the area of the intersection of the cost classes of all the comparisons is small. This case appears when the node bijections between both graphs that the user proposes are never the optimal ones or are only optimal in a small domain of K_n and K_e. In the case that the general cost class is elongated through the K_e axis, the inherent distance between image regions of the user performs in a different way than the distance between graph nodes of the system. On the contrary, in the case that the general cost class is elongated through the K_n axis, the system captures in a different way the relations between these regions.

In this way, by analyzing the application at hand and how the Graph Edit Distance behave over the graph representation and the data itself, we are able to adapt, change or replace the distance measure we are using.

3.3 Improving of sub-optimal graph matching algorithms

Property 2-3 defines that the edit surface increases when K_n and K_e increases. This certainly occurs when using optimal algorithms to compute the labelings given the values of K_n and K_e. However, we cannot assume these properties hold true if we compute the labeling space using suboptimal algorithms. Two examples are shown in Figure 3-3 and Figure 3-4, which show $\widehat{\Omega}$ obtained by the Graduated Assignment (Gold and Rangarajan 1996) using the graphs in Figure 2-5 and Figure 2-7. We see that most of the regions computed by the Graduated Assignment are not convex. A good approach for improving sub-optimal algorithms that minimize Graph Edit Distance would be to modify them to hold for Property 2-1 while predicting the labeling given some certain point in the labeling space.

Moreover, the consideration of Property 2-3 and Property 2-4 in graph edit distance result could help sub-optimal algorithms to achieve better performance. Figure 3-5 and Figure 3-6 show the EditSuface in a range of K_n and K_e values. Figure 3-5 is computed using the graphs of Figure 2-5 and

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Figure 3-6 using the graphs of Figure 2-7. Both figures show two surfaces. The first, printed in solid gray scale, is computed using an optimal A* algorithm. The second, printed using transparency is computed using the Graduated Assignment (Gold and Rangarajan 1996) algorithm. We see how the Graduated Assignment does not provide a good approximation for high K_e and relatively low K_n values. This means that the graduated assignment is able to compute very good edge labelings but fail in computing node assignments. We see that neither Property 2-3 nor Property 2-4 hold true for surfaces computed using the Graduated Assignment algorithm. A simple way of improving graph distance computations could be to compute several costs for different K_e and K_n values and average the results to force Property 2-3 to hold. Property 2-4 could also be use to filter the shape of the edit surface, possibly obtaining with this new corrected surface better approximations of the graph edit distance.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

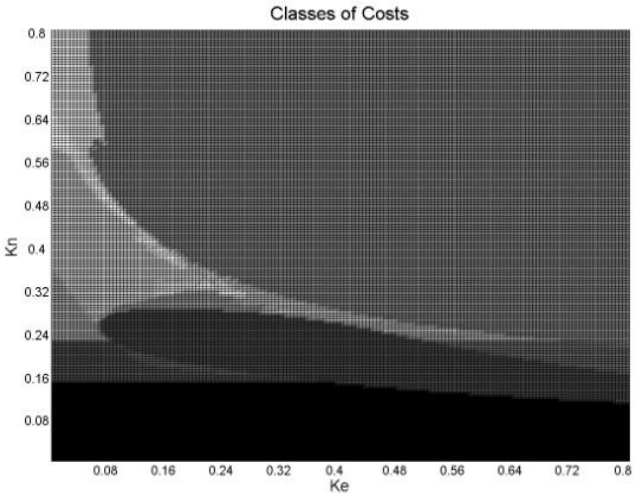


Figure 3-3: example of class of cost computation using a suboptimal algorithm. Graphs used are shown in Figure 2-5.

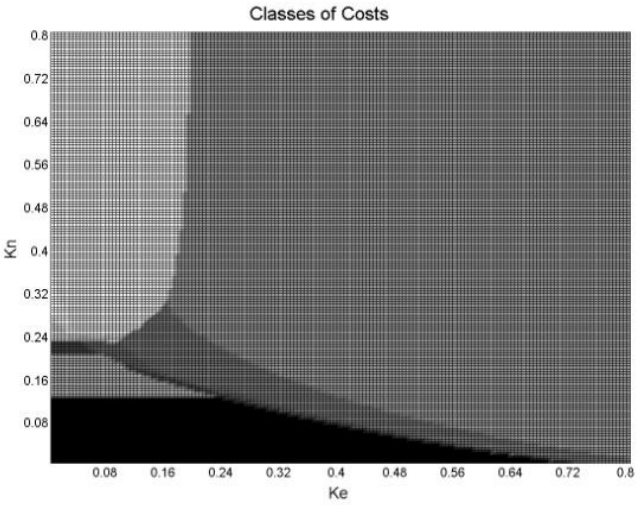


Figure 3-4: example of class of cost computation using a suboptimal algorithm. Graphs used are shown in Figure 2-7.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

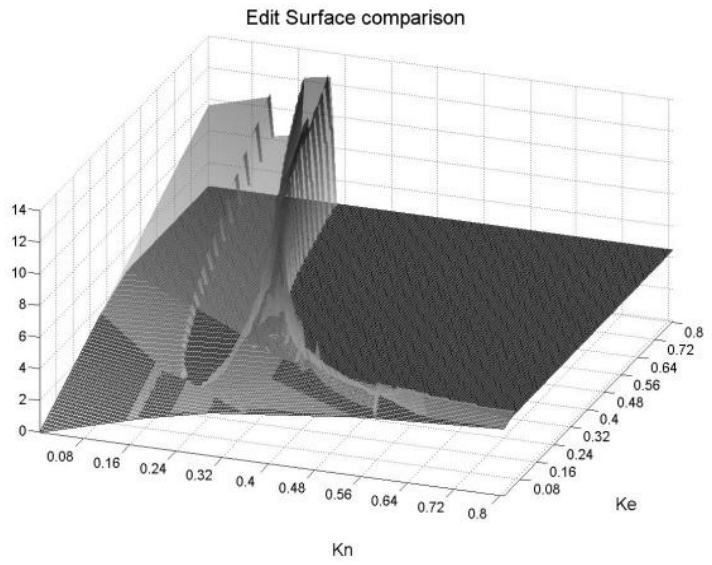


Figure 3-5: edit surfaces given by an optimal and a sub-optimal graph matching algorithm. Used graphs are shown in Figure 2-5.

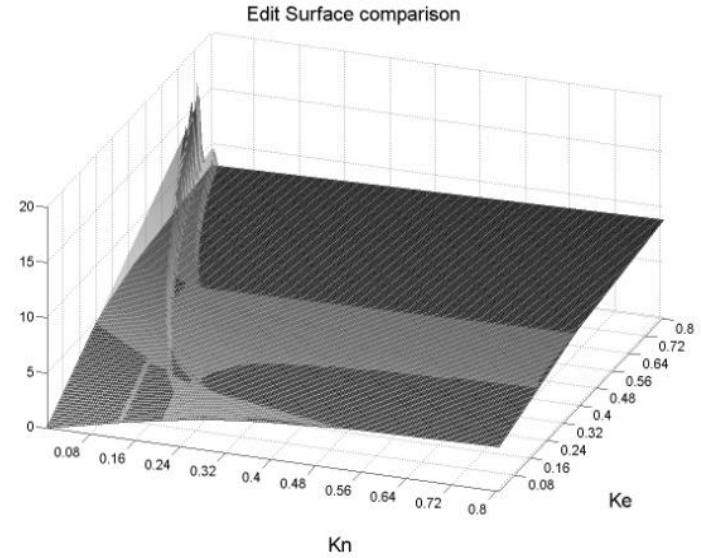


Figure 3-6: edit surfaces given by an optimal and a sub-optimal graph matching algorithm. Used graphs are shown in Figure 2-7.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 4

COMPUTATION OF GRAPH EDIT DISTANCE THROUGH DOMINANT SETS[4]

In this chapter, we describe a new algorithm to compute the graph edit distance based upon the notion of dominant set (Pavan and Pelillo 2003; Pavan and Pelillo 2007). Our idea generalizes a well-known method (Pelillo 1999) to reduce the problem of graph isomorphism to the maximum clique through the notion of association graph. In this case, in the association graph we do not look for cliques but for dominant sets, which are a generalization of maximal cliques in edge-weighted graphs.

1. RELATION OF GRAPH EDIT DISTANCE WITH THE DOMINTANT SETS

As described in chapter 2 several algorithms exist to compute the graph edit distance between two graphs. In the same way as the other algorithms, we focus our computation of the edit distance on the minimization of objective function in (1.8) with costs in (1.13) defined in chapter 2.

In order to minimize the graph edit distance through dominant sets, we define the association graph in the following form:

Definition 1-1: given G^P and G^q, we define the association graph $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{\gamma}_v, \tilde{\gamma}_E)$ as a tuple of four elements where $\tilde{V} = \Sigma_v^p \times \Sigma_v^q$ represents the node set, $\tilde{E} \subseteq \tilde{V} \times \tilde{V}$, $\tilde{\gamma}_v : \tilde{V} \to \mathbb{R}$ assigns a real value to each vertex and $\tilde{\gamma}_E : \tilde{E} \to \mathbb{R}$ assigns a real value to each edge.

[4] This work has been done with the collaboration of Nicola Rebagliati and Marcello Pelillo during my stay at *Università "Ca' Foscari" di Venezia*.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

As usual, each node of the association graph represents a matching between two nodes of the initial graphs. In addition, each edge of the association graph represents a matching between two edges of the initial graphs.

Thus, the main idea is to attribute edges of the association graph with the graph edit distance cost of labeling an edge of graph G^p to an edge of G^q, in a similar form as (1.6) in chapter 2. Thus, each clique (or dominant set of size N) will contain the graph edit distance cost of labeling both graphs under some bijection. To this aim, we define $\tilde{\gamma}_E$ as:

$$\tilde{\gamma}_E\left(\left(v_{a_1}^p, v_{b_1}^q\right),\left(v_{a_2}^p, v_{b_2}^q\right)\right) = \begin{cases} C_{v_{a_1}^p v_{b_1}^q, v_{a_2}^p, v_{b_2}^q} & v_{a_1}^p \neq v_{a_2}^p \wedge v_{b_1}^q \neq v_{b_2}^q \\ \frac{1}{2} & v_{a_1}^p = v_{a_2}^p \wedge v_{b_1}^q = v_{b_2}^q \\ 0 & otherwise \end{cases} \tag{1.1}$$

where,

$$\begin{aligned} C_{v_{a_1}^p v_{b_1}^q, v_{a_2}^p, v_{b_2}^q} &= (1-\alpha)\cdot \\ &\cdot\left(1-\frac{1}{\mathcal{C}_M}\left(c_1\left(\mathcal{C}_V\left(v_{a_1}^p, v_{b_1}^q\right)+\mathcal{C}_V\left(v_{a_2}^p, v_{b_2}^q\right)\right)\right.\right. \\ &\left.\left.+c_2\left(\mathcal{C}_E\left(\left(v_{a_1}^p, v_{a_2}^p\right),\left(v_{b_1}^q, v_{b_2}^q\right)\right)\right)\right)\right)+\alpha \\ c_1 &= \frac{N^2}{N-1}, c_2 = N^2, \alpha \in [0,1] \end{aligned} \tag{1.2}$$

$\mathcal{C}_M$ corresponds to the maximum value of $\mathcal{C}_V$ and $\mathcal{C}_E$ to adapt the cost to compatibility as indicated in (1.4) of chapter 2. N, as usual, corresponds to the number of nodes in the graphs (consider that null extensions of the graph may be required (see section 1.1.2.1 of chapter 2). Equation (1.2) basically inverts the cost of an edge labeling to a compatibility using last transformation function given in (1.7) of chapter 2. Parameter α has a similar effect to a regularization term. In this way, as α increases from zero to one $C_{v_{a_1}^p v_{b_1}^q, v_{a_2}^p, v_{b_2}^q}$ approaches 1. Constants c_1 and c_2, as we will see in the following, are related to the problem setting to compute the Graph Edit Distance. In addition the denominator of c_1 has the same effect described in (8.9) of chapter 2.

To analyze the relation between the graph edit distance problem and the dominant set problem we use a vectorial representation of the matching.

Definition 1-2: given a bijection f between graphs G^p and G^q, we represent f in the simplex as a barycentric point in some simplex x_f face:

$$x_{f_{a_1 * N + b_1}} = \begin{cases} \frac{1}{N} & if\ f\left(v_{a_1}^p\right) = v_{b_1}^q \\ 0 & otherwise \end{cases} \tag{1.3}$$

On the contrary we define the opposite transformation as:

Definition 1-3: given a point $\boldsymbol{x}$ in the simplex we define the bijection it represents as:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$f_x\left(v_{a_1}^p\right) = v_{b_1}^q \text{ if } x_{a_1*N+b_1} > 0 \tag{1.4}$$

Considering Definition 1-2, the dominant set objective function given in (1.28) of chapter 2, defining $A_{\alpha_{i,j}} = \tilde{\gamma}_E\left(\tilde{v}_i, \tilde{v}_j\right)$ and $I = \{i | x_{f_i} \neq 0\}$, we see that:

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \sum_{i=1}^{|\tilde{V}|} \sum_{j=1}^{|\tilde{V}|} x_{f_i} x_{f_j} A_{\alpha_{i,j}} =$$

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \frac{1}{N^2} \sum_{i \in I} \sum_{j \in I} A_{\alpha_{i,j}} = \frac{1}{N^2} \left[\sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} A_{\alpha_{i,j}} + \sum_{i \in I} A_{\alpha_{i,i}} \right]$$

Considering $i = a_1 * N + b_1$ and $j = a_2 * N + b_2$,

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \frac{1}{N^2} \left[\sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} ((1-\alpha) + \alpha) + \frac{\alpha - 1}{\mathcal{C}_M} \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} c_1 \left(\mathcal{C}_V\left(v_{a_1}^p, v_{b_1}^q\right) + \mathcal{C}_V\left(v_{a_2}^p, v_{b_2}^q\right) \right) \right.$$
$$\left. + c_2 \left(\mathcal{C}_E \left(\left(v_{a_1}^p, v_{a_2}^p\right), \left(v_{b_1}^q, v_{b_2}^q\right) \right) \right) + \sum_{i \in I} \frac{1}{2} \right]$$

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} \left(\frac{\mathcal{C}_V\left(v_{a_1}^p, v_{b_1}^q\right) + \mathcal{C}_V\left(v_{a_2}^p, v_{b_2}^q\right)}{N-1} \right.$$
$$\left. + \mathcal{C}_E \left(\left(v_{a_1}^p, v_{a_2}^p\right), \left(v_{b_1}^q, v_{b_2}^q\right) \right) \right)$$

Considering definition of F given in (1.4) of chapter 2 and Definition 1-2,

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \sum_{a_1=1}^{N} \sum_{b_1=1}^{N} \sum_{\substack{a_2=1 \\ a_2 \neq a_1}}^{N} \sum_{\substack{b_2=1 \\ b_2 \neq b_1}}^{N} F[a_1, b_1] F[a_1, b_2] \left(\frac{\mathcal{C}_V\left(v_{a_1}^p, v_{b_1}^q\right) + \mathcal{C}_V\left(v_{a_2}^p, v_{b_2}^q\right)}{N-1} \right.$$
$$\left. + \mathcal{C}_E \left(\left(v_{a_1}^p, v_{a_2}^p\right), \left(v_{b_1}^q, v_{b_2}^q\right) \right) \right) \tag{1.5}$$

Note that part of the objective function in (1.5) corresponds to the objective function of the Graph Edit Distance given by (1.8) and (1.13) of chapter 2. Hence, we can conclude that:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \mathcal{C}_{GED}(G^p, G^q, f) \tag{1.6}$$

We see that, if vector $\boldsymbol{x}$ corresponds to a barycentric point on a face of the simplex and $|\sigma(\boldsymbol{x})|_1 = N$, computing the dominant set functional is equivalent to computing the graph edit distance of the induced bijection $f_{\boldsymbol{x}}$. Thus, a direct consequence of (1.6) on the bijection that computes the graph edit distance f^* is that:

$$\langle \boldsymbol{x}^*, A_\alpha \boldsymbol{x}^* \rangle \geq \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \mathcal{C}_{GED}(G^p, G^q, f^*) \tag{1.7}$$

In addition, note that solutions of the dominant set functional $\boldsymbol{x}$ using payoff matrix A does not always correspond to a solution of the graph edit function since it may happen that $|\sigma(x)|_1 < N$ and so the solution does not correspond to a bijection between both graphs. However, increasing the parameter α we have a regularization (Pavan and Pelillo 2003) effect over matrix A_α. Consider the following equivalent problem:

$$\langle \boldsymbol{x}, \left((1-\alpha)A + \hat{A}\alpha\right)\boldsymbol{x} \rangle = (1-\alpha)\underbrace{\langle \boldsymbol{x}, A\boldsymbol{x} \rangle}_{k_1} + \alpha \underbrace{\langle \boldsymbol{x}, \hat{A}\boldsymbol{x} \rangle}_{k_2} \tag{1.8}$$

where A is defined considering:

$$\mathcal{C}_{v^p_{a_1} v^q_{b_1}, v^p_{a_2}, v^q_{b_2}} = \\ = \left(1 - \frac{1}{\mathcal{C}_M}\left(c_1\left(\mathcal{C}_V(v^p_{a_1}, v^q_{b_1}) + \mathcal{C}_V(v^p_{a_2}, v^q_{b_2})\right) + c_2\left(\mathcal{C}_E\left((v^p_{a_1}, v^p_{a_2}), (v^q_{b_1}, v^q_{b_2})\right)\right)\right)\right)$$

and $\hat{A}_{i,j} = 1$ if $A_{i,j} > 0 \wedge i \neq j$, $\hat{A}_{i,i} = 0$ otherwise. We see that k_1 corresponds to the original dominant set problem with payoff matrix (1.1). This function maximizes solutions with high internal coherence. The second term k_2, corresponds to the original un-attributed graph isomorphism problem as described in (Pelillo 1999). Thus, the second functional maximizes maximum/maximal cliques in the association graph defined by adjacency matrix $\hat{A}$. Note that, since we focus on the error tolerant graph matching problem, each possible bijection corresponds to a clique in the association graph. In the case of $\alpha = 0$, the problem corresponds purely to the dominant set problem. The main inconvenient of setting $\alpha = 0$ is that it is not possible to ensure that the solution of the dominant set corresponds to a bijection, it usually corresponds to some correspondence of a subset of nodes. Setting $\alpha = 1$ converts the problem to a purely graph clique problem. In this case, since every possible bijection in T will form a clique in the association graph, we have no information about the cost of node and edge local correspondences and therefore we do not compute any distance function. Nevertheless, we can ensure that the solution corresponds to the

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

barycenter of a face of the simplex. Values of α between 0 and 1 balance dominant set solutions and with a bijective solutions with barycentric solutions. Thus, moving α from 0 to 1 we tight the gap,

$$< \boldsymbol{x}^*, A_\alpha \boldsymbol{x}^* > \; - \left(1 - \frac{1}{2N}\right) + \frac{1-\alpha}{\mathcal{C}_M} \mathcal{C}_{GED}\left(G^p, G^q, f^*\right) \tag{1.9}$$

and we approach to equality shown on (1.5).

We formalize this intuition with the following theorem, which prove that there exist an α value lower than 1 where maximal/maximum solutions of the dominant set problem are in one-to-one correspondence to local/global solutions of the graph edit distance problem.

Theorem 1-1: Given two attributed graphs G^p and G^q and its respective regularized association graph G_α as defined in (1.1) and (1.2) there exist a value $\alpha^* \in [0,1)$ such that for every $\alpha > \alpha^*$ the bijection $f_{\boldsymbol{x}^*}$ induced by the support of the maximum dominant set $\boldsymbol{x}^*$ minimizes the graph edit distance between G^p and G^q.

Proof 1-1: we start denoting several values we will use in the proof. We denote the optimal dominant set value by:

$$x_\alpha^* = \arg\max_{x \in \Delta_n} \langle x, A_\alpha x \rangle, A_\alpha \in \mathbb{R}^{n \times n} \tag{1.10}$$

the optimal solution for the graph edit by:

$$f^d = \arg\min_{f \in T} d_{GED}(G^p, G^q, f) \tag{1.11}$$

and the dominant set functional value given any bijection by:

$$S_\alpha(f) = \max_{x \in \Delta_N} \langle x, A_\alpha^f x \rangle \tag{1.12}$$

where A_α^f represents the adjacency matrix of the association graph restricted to the support of x_f, that is to the values where $x_{f_i} > 0$.

We know from (1.5) that :

$$\langle \boldsymbol{x_f}, A_\alpha \boldsymbol{x_f} \rangle = \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \mathcal{C}_{GED}(G^p, G^q, f) \tag{1.13}$$

and

$$\langle \boldsymbol{x_{f^d}}, A_\alpha \boldsymbol{x_{f^d}} \rangle = \left(1 - \frac{1}{2N}\right) + \frac{\alpha - 1}{\mathcal{C}_M} \mathcal{C}_{GED}(G^p, G^q, f^d) \tag{1.14}$$

We focus on proving that $\forall f \in T \; S_\alpha(f) \leq S_\alpha\left(f^d\right)$, which is the same as:

$$S_\alpha\left(f^d\right) - S_\alpha(f) \geq 0 \tag{1.15}$$

To that aim let $R = A_\alpha^{f^d} - A_\alpha^f$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Lemma 1-1 (Positive points in the neighborhood of the barycenter of the simplex face) By continuity we know that if $\langle \frac{1}{N} e_N, R \frac{1}{N} e_N \rangle > 0$ there exist a value $\bar{\delta}$ such that:

$$\left\| v - \frac{1}{N} e_N \right\| \leq \bar{\delta} \Longrightarrow \langle v, Rv \rangle > 0 \tag{1.16}$$

□

Lemma 1-2 (Convergence of x_α with respect to α). There exist an $\bar{\alpha}$ such that $\forall\, \alpha > \bar{\alpha} \Longrightarrow \|x_\alpha^* - x_{\alpha=1}^*\| = \left\| x_\alpha^* - \frac{1}{N} e_N \right\| \leq \bar{\delta}$.

□

Proof 1-2: considering the decomposition of $F(\alpha) = \langle \boldsymbol{x}, A_\alpha \boldsymbol{x} \rangle$ given in (1.8) , we know by (Pelillo and Jagota 1995) and (Pavan and Pelillo 2007) that objective functions k_1 and k_2 are continuous and derivable in all domain of x and clearly they also are with respect to α. Thus, the linear combination of both objective functions is also continuous and derivable in all data domain. In addition, when $\alpha = 1$, the problem resembles the maximum clique problem; solutions of which, in (Pelillo and Jagota 1995) and (Motzkin and Straus 1965), are proven to correspond to strict local maximums of k_2. Besides, the Motzkin-Straus theorem shows that solutions maximizing k_2 correspond to vectors with the form $x_{\alpha=1}^* = N^{-1} e_N$ and so the linear combination of objective functions given in (1.8) will tend to converge to that solution while α increases. Concluding that, as α increases, solutions of (1.8) will tend to converge to the barycenter of the simplex. Thus,

$$\lim_{\alpha \to 1} \left\| x_\alpha^* - \frac{1}{N} e_N \right\| = 0$$

□

Extending (1.15) we have that:

$$\langle x_\alpha^{f^d}, A_\alpha^{f^d} x_\alpha^{f^d} \rangle - \langle x_\alpha^f, A_\alpha^f x_\alpha^f \rangle \geq 0 \tag{1.17}$$

where $x_\alpha^{f^d}$ and x_α^f are the best solutions that can be obtained with matrices $A_\alpha^{f^d}$ and A_α^f. Using R definition we have:

$$\begin{aligned} \langle x_\alpha^f, R x_\alpha^f \rangle &= \langle x_\alpha^f, A_\alpha^{f^d} x_\alpha^f \rangle - \langle x_\alpha^f, A_\alpha^f x_\alpha^f \rangle \\ \langle x_\alpha^f, A_\alpha^f x_\alpha^f \rangle &= \langle x_\alpha^f, A_\alpha^{f^d} x_\alpha^f \rangle - \langle x_\alpha^f, R x_\alpha^f \rangle \end{aligned} \tag{1.18}$$

substituting in (1.17)

$$\langle x_\alpha^{f^d}, A_\alpha^{f^d} x_\alpha^{f^d} \rangle - \langle x_\alpha^f, A_\alpha^{f^d} x_\alpha^f \rangle + \langle x_\alpha^f, R x_\alpha^f \rangle > 0 \tag{1.19}$$

considering optimality we know that:

$$\langle x_\alpha^{f^d}, A_\alpha^{f^d} x_\alpha^{f^d} \rangle - \langle x_\alpha^f, A_\alpha^{f^d} x_\alpha^f \rangle \geq 0 \tag{1.20}$$

holds for every α.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Considering $\langle \frac{1}{N} e_N, A_\alpha^{f^d} \frac{1}{N} e_N \rangle$ compute the graph edit compatibility (note that (1.2) converts cost to compatibility) we know that,

$$\langle \frac{1}{N} e_N, R \frac{1}{N} e_N \rangle = \langle \frac{1}{N} e_N, A_\alpha^{f^d} \frac{1}{N} e_N \rangle - \langle \frac{1}{N} e_N, A_\alpha^{f} \frac{1}{N} e_N \rangle \geq 0$$

and so by Lema1-1 and Lemma 1-2 we know that there exists an $\bar{\alpha}$ such that:

$$\langle x_\alpha^f, R x_\alpha^f \rangle \geq 0 \tag{1.21}$$

Concluding that for every $> \bar{\alpha}$, point $x_\alpha^{f^d}$ associated to the optimal graph edit distance bijection f^d will correspond to a maximal dominant set.

2. COMPUTING THE GRAPH EDIT DISTANCE THROUGH THE DOMINANT SET FRAMEWORK.

In order to compute the graph edit distance between to graphs we reduce the problem to the dominant set problem, which optimizes the functional given in (1.28) of chapter 2 with the payoff matrix defined in (1.1) and (1.2). In this way, the problem reduces to the optimization of a non-convex quadratic function. To compute the solutions to this last problem, there are several solutions available among them we highlight the Replicator Dynamics (Pelillo 1999), Exponential Replicator Dynamics (Pelillo 1999) and the Infection and Immunization Dynamics (Bulo' and Bomze 2011; Bulo', Pelillo et al. 2011). Among them, we recommend the use of Infection and Immunization Dynamics because of its good balance between speed and performance. Thus, using any of those algorithms to compute the dominant set we propose the following algorithm to compute the graph edit distance using the dominant set framework:

Algorithm *dominant set graph edit distance*(G^p, G^q)
$\alpha = setAlpha()$
$itr = 0; minCost = \infty; minF = \emptyset$
AG=constructAssociationGraph(G^p, G^q, α) *(Definition 1-1 and (1.1))*
for $itr <$ *maxItr*
  $initX = -log\left(rand(N^2, 1)\right)$
  $initX_i = \frac{1}{\sum_{j=1}^{N} initx_j} initX_i$
  $x_f =$ *InImDymAlg*$(initX, AG)$
  f=extractBijection(x_f) *(1.4)*
  $c = \mathcal{C}_{GED}(G^p, G^q, f)$
  if $c < minCost \wedge isbijection(f)$
    $minCost = c; minF = f$
  end
  $itr = itr + 1$
end
Returns *minF*

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Algorithm 2-1: pseudo-code of the dominant set graph edit distance algorithm.

As we see in Algorithm 2-1, there are several third party algorithms involved. The *InImDymAlg* computes a suboptimal solution of the dominant set problem with algorithm in (Bulo' and Bomze 2011). The function *setAlpha* returns an alpha such that the algorithm approaches the graph edit distance, this function is detailed in the next section. $rand(\mathrm{N}^2, 1)$ generates a column vector with random values. And, finally $isbijection(\mathrm{f})$ returns *true* if f is a bijection otherwise returns false.

3. LOCATING THE CORRECT α

As we see, in Section 2 and Section 1, the algorithm relies on a correct value of the α parameter. This value is not easy to compute and, in fact, we use a heuristic methodology to compute the value that fulfills our requirements. As we saw in Section 1, as we increase the α value its effect on the solutions is that approaches more to the clique objective function than dominant set objective function. Since each clique corresponds to a bijection, the value α should be large enough to force the algorithm to return a bijection. On the contrary, large values of alpha will tend to not consider the values of the association matrix, which will make difficult for the optimization algorithm to find the maximum solution. Since we aim to find the lowest value that returns a bijection, we will, for each problem at hand, iteratively transverse the possible values of alpha from low to high and take the lower value which give a high probability (in the experiment we used the 90%) to return a bijection between any pair of graph. We usually, do this process in an training phase. Once this training phase is done, we set a static α to compute the desired bijections.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 5

COMPUTATION OF COMMON LABELING

This chapter introduces six new methods to compute suboptimal solution to the common labeling problem. For each of them, we describe the objective function, how it is approximation and the methodology used to find the solution.

Considering that several algorithms are described through the chapter we summarize them in Summary 1. Summary shows the name of the algorithm, the computational cost (T refers to the number of iteration of the algorithm, N to the number of nodes of the graphs and P the number of graphs in the set) and if the algorithm computes the common labeling through a consistent multiple isomorphism or directly the common labeling.

Algorithm	CC	CMI	CL
P-Dim Graduated Assignment	$O(TN^{2P})$	∘	
Agglomerative Graduated Assignment	$O(N^P)$	∘	
Least squares method	$O(P^3N^6)$	∘	
Average Alignment	$O(TP^2N^4)$		∘
Common Labeling Graduated Assignment	$O(TP^2N^4)$		∘
Common Labeling dominant sets	$O(TP^2N^3)$ using (Bulo' and Bomze 2011)	∘	

Summary 1: Algorithms summary.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1. THE COMMON LABELING THROUGH A CMI

1.1 CMI using P-Dimensional assignation matrix

The two algorithm we propose in the next three sections, given a node of each graph of the set $|\Gamma|$, i.e. $v_i^1, v_j^2, \dots, v_k^P$, compute the probability that this set of nodes represent the same local part of the object. That is, the probability that these nodes are labeled to the same node $l_p \in L$. The more similar the features of each node are (and/or their adjacent vertices), the greater is the probability. We represent this probability by $\mathrm{P}(v_i^1, v_j^2, \dots, v_k^P)$, where v_i^p is the node i of the graph G^p. This probability is defined as the joint probability of all the isomorphisms between the graphs of the set and the related nodes,

$$\begin{aligned} P\left(v_i^1, v_j^2, \dots, v_k^P\right) = \\ = P\left(f^{1,2}\left(v_i^1\right) = v_j^2, \dots, f^{1,N}\left(v_i^1\right) = v_k^N, \dots\right) \end{aligned} \tag{1.1}$$

We graphically represent this joint probability as a hypercube. Each coordinate of the hypercube is related with a graph and the number of cells per coordinate is the number of nodes per graph. Consequently, we represent each cell of the hypercube as $P^{1,2,\dots,P}[i, j, \dots, k]$ which represents the joint correspondence $v_i^1, v_j^2, \dots, v_k^P$. Figure 1-1 shows a hypercube of three dimensions that represents the joint probability of a set of 3 graphs. The value of the highlighted cell in the cube represents the joint probability of nodes v_2^1, v_1^2 and v_2^3, which corresponds to the joint probability depicted by the bijection in Figure 1-2. Note that in the case that the set is composed by only two graphs, the joint probability is represented by a 2-dimensional matrix, which is equivalent to the representation that most of the probabilistic graph-matching algorithms do (see section 1.1.2 of chapter 1). For this reason, we consider the following methodology a generalization of the probabilistic graph-matching methodologies.

Figure 1-3 shows the proposed methodology. Differences with respect to the consistent multiple isomorphism methods described in section 2.3 of chapter 2 appear in both steps. In the first step, the set of assignation matrices is substituted by an assignation hypercube to alleviate the problem of taking the individual assignation matrices independently. As a consequence, the first step computes a probability hypercube. The second step computes the consistent MI. This step is exchanged by a discretization process.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

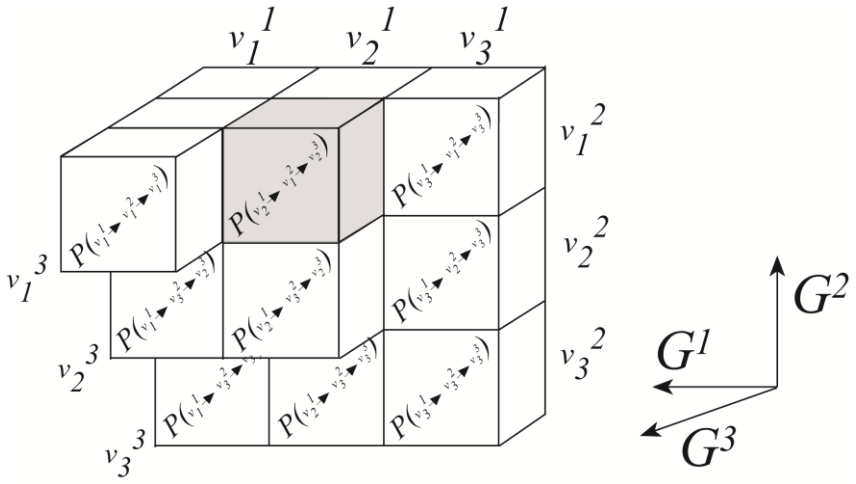


Figure 1-1: graphical representation of the joint probability.

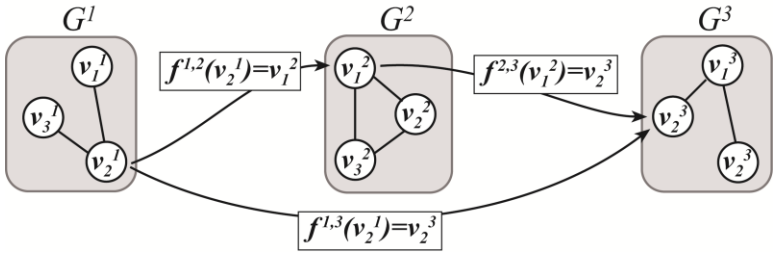


Figure 1-2: representation of probability of Figure 1-1.

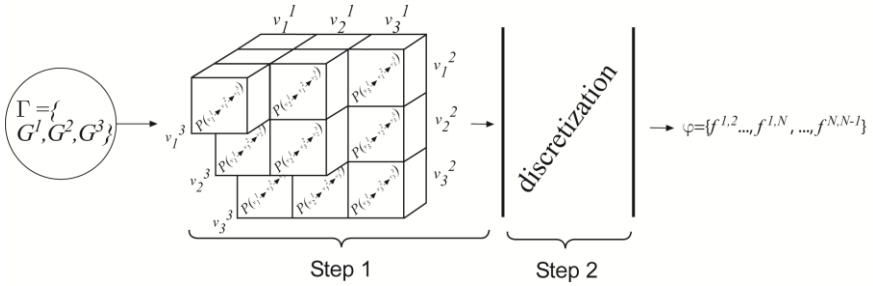


Figure 1-3: scheme to compute a CMI based on an assignation hypercube. Example using $N = P = 3$.

We present two possible approaches to compute the association hypercube. The first approach, called **P-Dimensional Graduated Assignment**, computes the joint probability $\mathrm{P}\left(v_i^1, v_j^2, \ldots, v_k^P\right)$ considering jointly the costs of all the related assignments (1.1). As we will see, the algorithm which computes (1.1) is high computationally demanding, due to the exponential size of the hypercube. We developed a second approach

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

which reduces the computational cost by assuming independence between isomorphisms $f^{p,q}$ in (1.1). Certainly, this second approach is an approximation of the first one, and consequently the cost of a common labeling computed with the first approach is usually lower than the cost of a common labeling computed with the second approach. We call this second approach *Agglomerative Graduated Assignment* and in this case, the joint probability is computed as follows:

$$\begin{aligned} &P\left(v_i^1, v_j^2, v_m^3, \dots, v_k^P\right) = \\ &\quad = P\left(f^{1,2}(v_i^1) = v_j^2\right) \cdot P(f^{1,3}(v_i^1) = v_m^3) \cdot \dots \cdot P\left(f^{P-1,P}\left(v_q^{P-1}\right) = v_k^P\right) \end{aligned} \tag{1.2}$$

In (1.2) each assignation between nodes is considered to be independent and so the joint probability can be obtained as a product of the individual ones. Using matrix representation of the pre-computed pair-wise probabilities, the joint probability of the set of nodes $S = \left\{v_i^1, v_j^2, \dots, v_k^P\right\}$ can be expressed as:

$$\begin{aligned} &P(S) = P\left(v_i^1, v_j^2, \dots, v_k^P\right) = \prod_{p,q=1}^{P} F^{p,q}[a,b] \ , v_a^p, v_b^q \in S \\ &P\left(v_i^1, v_j^2, \dots, v_k^P\right) = F^{1,2}[i,j] \cdot F^{1,3}[i,p] \cdot \dots \cdot F^{P-1,P}[p,k] \end{aligned} \tag{1.3}$$

1.1.1 P-Dimensional Graduated assignment

One of the most efficient algorithms used to compute a sub-optimal solutions for the error tolerant graph isomorphism problem between two graphs G^p and G^q is known as *Graduated Assignment (Gold and Rangarajan 1996)*. The aim of this iterative algorithm is to solve the quadratic assignment problem (Garey and Johnson 1979) by maximizing the objective function defined in (1.2) of chapter 2. Since the optimization procedure used by the graduated assignment is proven to be fast and quite effective, we follow the same idea. Consider, without lose of generality the case of three graphs. The objective function becomes:

$$\begin{aligned} &C\left(\Gamma, F^{p,q,k}\right) = \\ &\quad = \sum_{a}^{N}\sum_{i}^{N}\sum_{n}^{N}\sum_{b}^{N}\sum_{j}^{N}\sum_{m}^{N} F^{p,q,k}[a,i,n] \cdot F^{p,q,k}[b,j,m] \cdot C_{ain,bjm}^{p,q,k} \end{aligned} \tag{1.4}$$

where :

$F^{p,q,k} \in [0,1]^{|\Sigma_v^p| \times |\Sigma_v^p| \times |\Sigma_v^p|}$ is a doubly stochastic matrix which represents a function $f^{p,q,k}: \Sigma_v^p \times \Sigma_v^p \times \Sigma_v^p \to [0,1]$ defined as $f^{p,q,k}(v_i^1, v_j^2, \dots, v_k^P) = P\left(v_i^1, v_j^2, \dots, v_k^P\right)$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

We approximate the function in the point $\left(F_f^{p,q,k}\right)^t$, at time t, using the Taylor series expansion, which following the same notation as the original paper becomes:

$$C\left(\Gamma, F^{p,q,k}\right) \approx C'\left(\Gamma, \left(F^{p,q,k}\right)^t\right) =$$
$$= K_1 + \sum_a^N \sum_i^N \sum_n^N Q_{a,i,n}^{p,q,k} \left(F^{p,q,k}[a,i,n] - \left(F^{p,q,k}[a,i,n]\right)^t\right) \quad (1.5)$$
$$Q_{a,i,n}^{p,q,k} = \sum_b^N \sum_j^N \sum_m^N \left(F^{p,q,k}[b,j,m]\right)^t \cdot C_{ain,bjm}^{p,q,k}$$

Therefore,

$$\max_{F^{p,q,k}} C'(\Gamma, (F^{p,q,k})^t) = \max_{F^{p,q,k}} \sum_a^N \sum_i^N \sum_n^N Q_{a,i,n}^{p,q,k} \cdot F^{p,q,k}[a,i,n] \quad (1.6)$$

which can be stated as a linear assignment problem over the gradient on the current point $\left(F^{p,q,k}\right)^t$. Thus, the update step becomes:

$$\left(F^{p,q,k}[a,i,n]\right)^{t+1} = e^{\beta \cdot \sum_{b=1}^N \sum_{j=1}^N \sum_{k=1}^N \left(F^{p,q,k}[b,j,k]\right)^t \cdot C_{ain,bjm}^{p,q,k}} \quad (1.7)$$

The final algorithm proceeds in three steps, (1st) given an initial point (at time t) compute next assignation (at time $t+1$), (2nd) apply "continuous" discretization using β parameter and (3rd) project the results over the triply stochastic matrix space. Repeat step 1, 2 and 3 until convergence.

Equivalently, as the original paper we compute the projection to the double stochastic space using the Stinkhorn (Sinkhorn 1964) algorithm extended to P dimensions to normalize the multiple assignment matrix P. The compatibility $C_{ain,bjm}^{p,q,k}$ can be computed similarly, in a joint form, to (1.6) of chapter 2. Specifically,

$$C_{ain,bjm}^{p,q,k} = C_{ai,bj}^{p,q} + C_{an,bm}^{p,k} + C_{in,jm}^{q,k} \quad (1.8)$$

Since the algorithm may not reach a discrete matrix, we recommend a discretization of matrix $F^{p,q,k}$. A naive approach to this discretization could be to iteratively choose the maximum assignations until discretize matrix F' fulfils:

$$\sum_{a=1}^N F^{p,q,k}[a,i,n] = 1, \forall i,n = \{1..N\}$$
$$\sum_{i=1}^N F^{p,q,k}[a,i,n] = 1, \forall a,n = \{1..N\} \quad (1.9)$$
$$\sum_{n=1}^N F^{p,q,k}[a,i,n] = 1, \forall a,i = \{1..N\}$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

The cost of the algorithm is linear on the number of iterations and the cost of each iteration is $O(N^{2P})$.

1.1.2 Agglomerative Graduated Assignment method

Like in the P-Graduated Assignment, we describe the method considering that Γ is composed by 3 graphs. The algorithm is based on the assumption that each pair-wise isomorphism is independent with respect to the other pair-wise isomorphisms, see (1.3). The algorithm is composed by three main steps. The first computes all the probability matrices $F^{p,q}$ (lines 38 to 40 of Algorithm 1-1). The second computes the joint probability $F^{1,2,\ldots,P}$ applying (1.3) (lines 41 to 47). The third does the discretization process to obtain the final CMI (line 48),

```
37  Algorithm Agglomerative Graduated Assignment(G = {G^p, G^q, G^k})
38  F^{p,q} = doGraduatedAssignmentMethod(G^p, G^q)
39  F^{q,k} = doGraduatedAssignmentMethod(G^q, G^k)
40  F^{p,k} = doGraduatedAssignmentMethod(G^p, G^k)
41  for a in all nodes in G^p
42      for i in all nodes in G^q
43          for n in all nodes in G^k
44              F^{p,q,k}[a,i,n] = F^{p,q}[a,i] · F^{q,k}[i,n] · F^{p,k}[a,n]
45          end
46      end
47  end
48  φ = discretize(F^{p,q,k})
49  Returns φ
```

Algorithm 1-1: agglomerative Graduated Assignment algorithm for the case of |G|=3.

To extend the algorithm to greater cardinalities of Γ, it is necessary to compute step one and step two considering all the graphs. That is, in step one we compute all pair-wise labelings and in step two we compute the P-dimensional hypercube.

With respect to the computational cost, the first step computes a quadratic number of isomorphisms respect to the number of graphs, so the cost of the first step is $O(P^2 \cdot T \cdot N^4)$, where T is the number of iterations in the Graduated Assignment. The second step computes the hypercube of N^P cells, so the cost is $O(N^P)$.

Several similarities can be noticed with the algorithm in (Lozano, Escolano et al. 2009) where all pair-wise computations are performed independently and some consistency rules are applied a posteriori. The main advantage of algorithm in Algorithm 1-1 is that the joint node probability is explicitly computed and consequently the discretization (or consistency rules) works on a global fashion.

Even though the global computational cost has been drastically reduced, in comparison with the P-Dimensional Graduated Assignment algorithm, it

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

is still exponential with respect to the number of nodes and graphs. For this reason, in the following sections, we present several algorithms that compute directly a CL without computing the CMI. The main advantage of computing directly the common labeling is that we do not need to address the problem of computing consistent individual isomorphism because all common labeling assignations fulfill this requirement.

2. COMPUTING DIRECTLY THE COMMON LABELING

2.1 Probabilistic framework

In the same way as other probabilistic algorithms where assignation matrix F is defined as:

$$F^{p,q}[a,i] = P\left(f^{p,q}\left(v_a^p\right) = v_i^q\right) \tag{2.1}$$

we define the probability of matching node v_i^p to a virtual node l_j as a continuous assignation matrix H^p:

$$H^p[i,j] = P\left(h^p\left(v_i^p\right) = l_j\right) \tag{2.2}$$

We consider that the probability of matching a vertex v_i^p of graph G^p to a vertex l_j of the virtual node set L is the union of probabilities of all the paths that goes through the nodes ($v_{1..N}^q$) of a third graph G^q. In other words,

$$\begin{aligned} P\left(h^p\left(v_i^p\right) = l_j\right) = P\big(&\left[f^{p,q}\left(v_i^p\right) = v_1^q \wedge h^q\left(v_1^q\right) = l_i\right] \\ &\vee \left[f^{p,q}\left(v_i^p\right) = v_2^q \wedge h^q\left(v_2^q\right) = l_i\right] \vee \dots \\ &\vee \left[f^{p,q}\left(v_i^p\right) = v_N^q \wedge h^q\left(v_N^q\right) = l_i\right]\big) \end{aligned} \tag{2.3}$$

It seems logical that the events of matching v_i^p to l_j through different nodes cannot occur simultaneously. For this reason (similarly to (Williams, Wilson et al. 1997)), we assume that these events are mutually exclusive. Consequently, using the addition rule of probability and the mutual exclusivity assumption, (2.3) becomes:

$$\begin{aligned} P\left(h^p\left(v_i^p\right) = l_j\right) = P\big(&\left[f^{p,q}\left(v_i^p\right) = v_1^q \wedge h^q\left(v_1^q\right) = l_i\right]\big) \\ &+ P\left(f^{p,q}\left(v_i^p\right) = v_2^q \wedge h^q\left(v_2^q\right) = l_i\right) + \cdots \\ &+ P\left(f^{p,q}\left(v_i^p\right) = v_N^q \wedge h^q\left(v_N^q\right) = l_i\right) \end{aligned} \tag{2.4}$$

In order to compute (2.4) we consider, at this point, that $f^{p,q}$ is independent of h^q. Therefore:

$$\begin{aligned} P\left(h^p\left(v_i^p\right) = l_j\right) = \\ = P\left(f^{p,q}\left(v_i^p\right) = v_1^q\right) \cdot P\left(h^q\left(v_1^q\right) = l_i\right) + \\ + P\left(f^{p,q}\left(v_i^p\right) = v_2^q\right) \cdot P\left(h^q\left(v_2^q\right) = l_i\right) + \\ + \cdots + \end{aligned} \tag{2.5}$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$+P\left(f^{p,q}\left(v_i^p\right)=v_N^q\right)\cdot P\left(h^q\left(v_N^q\right)=l_i\right)$$

Now, combining (2.5) with (2.1) and (2.2) we obtain,

$$H^p[i,j]=\sum_{k=1}^{N}F^{p,q}[i,k]\cdot H^q[k,j]\Longrightarrow \\ \Longrightarrow H^p=F^{p,q}\cdot H^q \tag{2.6}$$

Besides, in a similar way, considering the probabilities in matrices H^p and H^q, we could obtain:

$$F^{p,q}=H^p\cdot(H^q)^T \tag{2.7}$$

Note that, in case matrices H are orthogonal we could obtain (2.7) from (2.6). However, since in our case we cannot assume orthogonality, matrices in equations (2.6) and (2.7) may not be equivalent. In the methods, we propose, we never consider these matrices to be the same, we just use equations (2.6) and (2.7) to compute the probabilities.

In the following, we will represent with symbol Φ the set of matrices $\Phi=\{F^{p,q},\forall p,q=\{1..P\}\}$ related to the multiple isomorphism φ. In the same way, we represent with symbol Ψ the set of matrices $\Psi=\{H^p,\forall p=\{1..P\}\}$.

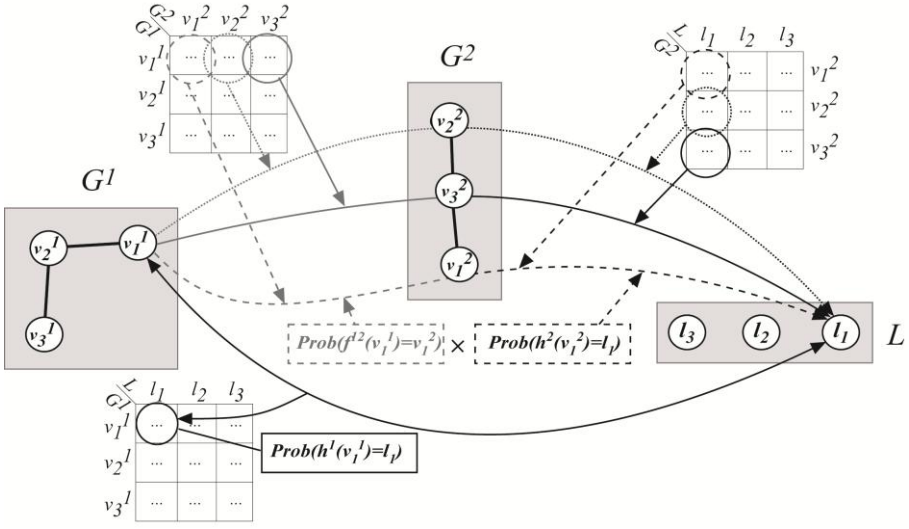


Figure 2-1: computation of the probability of matching v_1^1 to l_1

Figure 2-1 shows two attributed graphs and the virtual node set L. In this example, the probabilities of matching G^1 to L are computed through the probabilities of matching G^1 to G^2 and G^2 to L. Consider as an example the expression represented by a solid line which is computed as the addition of the three probabilities (represented by three dotted lines) of matching v_1^1 to l_1 through G^2. These probabilities are :

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\mathrm{P}\left(f^{1,2}(v_1^1) = v_i^2\right) \cdot \mathrm{P}\left(h^2\left(v_i^2\right) = l_1\right), i = 1,2,3 \tag{2.8}$$

2.2 Average alignment common labeling

Assume we have a consistent multiple isomorphism φ where the bijections are $\varphi = \{f^{1,2}, \dots, f^{2,1}, \dots, f^{N-1,N}\}$. Moreover, we have the induced CL $\psi = \{h^1, h^2, \dots, h^P\}$ (see (2.6) of chapter 2). Using some properties of the composition of functions, we see that are able to compute h^p using any other bijection in ψ and the corresponding one in φ. We can derive that equality:

$$h^p = f^{p,1} \circ h^1 = f^{p,2} \circ h^2 = \cdots = f^{p,P} \circ h^P \tag{2.9}$$

holds. Assuming so, we can derive several properties.

The former report that, if we fix any bijection in ψ e.g. $h^1(v_i^1) = l_i$, we are able to compute the set ψ using only the set φ and this reference point, i.e.

$$h^1 = id_M, h^2 = f^{2,1} \circ id_M, \dots., h^P = f^{P,1} \circ id_M \tag{2.10}$$

From (2.9) we conclude that ψ can be computed using several equivalent equations. Some of them are shown below:

$$\begin{aligned} h^p &= f^{p,p-1} \circ h^{p-1} = f^{p,p-1} \circ f^{p-1,p-2} \circ h^{p-2} \\ &= f^{p,p-1} \circ f^{p-1,p-2} \circ \dots \circ h^1 \end{aligned} \tag{2.11}$$

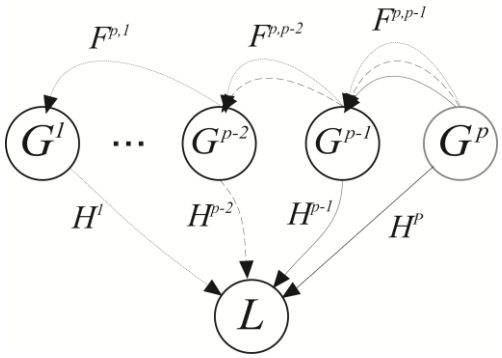


Figure 2-2: three possible equivalent ways of computing Hp. Each way is shown by a line pattern.

Figure 2-2 illustrates (2.11).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

The latter property concludes that given ψ and φ the related permutation matrices F and H it is possible to obtain each matrix of the set ψ as an average of all possible equalities in (2.9). That is,

$$H^p = \frac{1}{P-1}[F^{p,1} \cdot H^1 + \cdots + F^{p,k} \cdot H^k + \cdots + F^{p,P} \cdot H^P] \qquad (2.12)$$

where H^p and $F^{p,q}$ represents a permutation matrix induced with bijections h^p and $f^{p,q}$.

All equalities shown above are useful when φ form a consistent multiple isomorphism, independently how bijections $f^{p,q}$ are computed. However, in real data (due to distortion on the object representation, outliers or distortion induced by sub-optimality of the matching algorithms), it is usual that $h^p = f^{p,q} \circ h^q \neq f^{p,k} \circ h^k$. Thus, in real applications (2.9) may not hold, giving some different values of bijection h^p. Thus, to compute a single solution for the common labeling ψ, we rely on equation (2.12). The intuition is simple, each bijection H^p can be computed as the maximum estimation of all the other bijections $H^{1..P}$, given a reference set of permutations $f^{1..P}$ of the graphs $G^{1..P}$ to the current graph G^p. In this way, each probability $H^p[a,i]$ corresponds to the mean of all other assignation to the given l_i node. We generalize this intuition to the computation of the common labeling ψ. In this way, we relax the domain of matrix F and H to the continuous domain as shown in Section 2.1. Consequently, we approximate each bijection H^p as:

$$\bar{H}^p = \frac{1}{P-1}[F^{p,1} \cdot H^1 + \cdots + F^{p,k} \cdot H^k + \cdots + F^{p,P} \cdot H^P] \qquad (2.13)$$

Note that in case we assume Gaussian noise on the distortion of matrices $F^{p,q}$, the sample mean $\bar{H}^p$ would be a good approximation of the population mean H^p. Nevertheless, approximation in (2.13) resembles the chicken-egg problem where H and F depend one on the other and therefore cannot be computed directly. We aim to solve it using consecutive approximations. To this aim in sections 2.2.1 and 2.2.2 we describe two algorithms that bases its intuition on the approximation given in (2.13).

2.2.1 Least squares method.

The first method we present is based on an approximation that considers (2.13) as a system of linear equations. Even the method is placed in section 2, the method does not compute a common labeling directly, but it does it using previously computed pair-wise bijection. We place this method here because it is based on the probabilistic framework described in 2.1. Since we know that a large set of algorithms to compute a bijection between two graphs exists, in this algorithm we consider that all the pair-wise bijections that compose a multiple isomorphism are known.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

It is clear that considering Φ is known and fixed, we could interpret (2.13) as a system of linear equations, that in matrix form will look like:

$$k_1 \cdot \begin{bmatrix} H^1[1,1] & H^1[1,2] \\ H^1[2,1] & H^1[2,2] \end{bmatrix} = \begin{bmatrix} F^{1,2}[1,1] & F^{1,2}[1,2] \\ F^{1,2}[2,1] & F^{1,2}[2,2] \end{bmatrix} \begin{bmatrix} H^2[1,1] & H^2[1,2] \\ H^2[2,1] & H^2[2,2] \end{bmatrix} + F^{1,3} \cdot H^3$$
$$\cdots$$
$$k_1 \cdot \begin{bmatrix} H^3[1,1] & H^3[1,2] \\ H^3[2,1] & H^3[2,2] \end{bmatrix} = \begin{bmatrix} F^{3,1}[1,1] & F^{3,1}[1,2] \\ F^{3,1}[2,1] & F^{3,1}[2,2] \end{bmatrix} \begin{bmatrix} H^1[1,1] & H^1[1,2] \\ H^1[2,1] & H^1[2,2] \end{bmatrix} + F^{3,2} \cdot H^2 \quad (2.14)$$

$$k_1 = P - 1$$

The system corresponds to a linear system with $N^2 \cdot P$ equations and $N^2 \cdot P$ unknowns:

$$\begin{aligned} k_1 \cdot H^1[1,1] &= F^{1,2}[1,1] \cdot H^2[1,1] + \cdots + F^{1,3}[1,1] \cdot H^3[1,1] + F^{1,3}[1,2] \cdot H^3[2,1] \\ H^1[1,2] &= F^{1,2}[1,1] \cdot H^2[1,2] + F^{1,2}[1,2] \cdot H^2[2,2] + \cdots \\ &\cdots \\ k_1 \cdot H^3[2,2] &= F^{3,1}[2,1] \cdot H^1[1,2] + \cdots + F^{3,2}[2,1] \cdot H^2[1,2] + F^{3,2}[2,2] \cdot H^2[2,2] \end{aligned} \quad (2.15)$$

However, due to the system does not have any independent term, it just has one trivial solution $H^p = 0_{N \times N}$. This trivial solution has a comprehensible meaning due to the virtual set *L* has neither attributes nor structure. One solution to convert the problem to an over-determined system is to impose some H^p. Without loss of generality we consider $H^1 = I_{N \times N}$. Which produces the following considerations: $h^1\left(v_i^1\right) = h_i, \forall i = \{1..P\}$ as in (2.6) of chapter 2. With this restriction, the system of linear equations given in (2.15) becomes an over-determined system with $N^2 \cdot P$ equations and $N^2(P-1)$ variables. For the case of three graphs, the final over-determined system becomes:

$$\underbrace{\begin{bmatrix} I_{N \times N} & F^{1,2} & F^{1,3} \\ F^{2,1} & -I_{N \times N} & F^{2,3} \\ F^{1,3} & F^{2,3} & -I_{N \times N} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 0 \\ H^2 \\ H^3 \end{bmatrix}}_{H^{Lin}} = \underbrace{\begin{bmatrix} I \\ -F^{2,1} \\ -F^{3,1} \end{bmatrix}}_{B} \quad (2.16)$$

Since the system may not have a solution, we approximate it via the least squares approach (Williams 1990), in this way we minimize the square error:

$$H^{Lin^*} = arg \min_{H^{2..P}} \left\| AH^{Lin} - B \right\|_2 \quad (2.17)$$

The cost of solving (2.17) is $O(N^6 \cdot P^3)$ (to perform the QR factorization using the Gram-Schmidt) plus $O((N^2 \cdot P)^2 + 2(N^2 \cdot P)^2)$ (to solve the system using back substitution) (Björck 1974) section 5.7. In addition, we have to add the cost of computing all the multiple isomorphism Φ.

2.2.2 Average alignment algorithm

The algorithm we propose uses any error-tolerant graph-matching algorithm, that we identify with *K*, that iteratively updates a probability matrix that represents the probability of the isomorphism

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

between nodes. Each step of the algorithm is represented by:

$$(F^{p,q})^{t+1} = K^{step}(G^p, G^q, (F^{p,q})^t) \tag{2.18}$$

where G^p,G^q are a pair of graphs to be compared, $(F^{p,q})^t$ and $(F^{p,q})^{t+1}$ are the probability matrices at steps or iterations t and $t+1$ and K^{step} is a function that executes only one iteration of algorithm K. It is assumed that $(F^{p,q})^{t+1}$ is a better approximation of the labeling $F^{p,q}$ than $(F^{p,q})^t$. That is, the cost is reduced,

$$\mathcal{C}(G^p, G^q, (F^{p,q})^{t+1}) \leq \mathcal{C}(G^p, G^q, (F^{p,q})^t) \tag{2.19}$$

50 ***Algorithm*** *Average alignment Common-Labeling* (Γ)
51 $H^i = initializeCL();$
52 ***repeat until*** $(H^p)^t, \forall\, p \in \{1..N\}$ *converges* ***or*** $t > MaxIterations$
53 ***for all*** $G^p \in \Gamma$
54 $(H^p)^{t+1} = 0_{N\times N};$
55 ***for all*** $G^q \in \{\Gamma - G^p\}$
56 $(F^{p,q})^{t+1} = K^{step}(G^p, G^q, (H^p)^t \cdot ((H^q)^t)^T)$
57 $H' = enhanceEntropy((F^{p,q})^{t+1}, (F^{p,q})^{t+1} \cdot (H^q)^t)$
58 $(H^p)^{t+1} = (H^p)^{t+1} + \frac{1}{P-1}H'$
59 *end*
60 [5] $(H^p)^{t+1} = normalize((H^p)^{t+1})$
61 *end*
62 $t = t + 1$
63 *end*
64 $\Psi = \Lambda(P^h)$
65 *Returns* Ψ

Algorithm 2-1: Probabilistic Common-Labeling Assignment.

The algorithm has two main steps. In the first one, it updates all $F^{p,q}$ (line 56) computing an iteration of algorithm K. The input of K^{step} is a previous $F^{p,q}$ approximation computed as described in (2.7). We can ensure that at every iteration, we approach to a local minimum due to (2.19). However, even if function K^{step} minimizes the cost of the multiple isomorphism it does not guarantee the global consistency (the solution is a consistent multiple isomorphism). This is the objective of the second step whereby computing (2.13) (line 58) $\bar{H}^p$ fulfils this requirement.

The initialization of H at time 0 (line 51) requires that at least one H^p is different from a uniform distribution, i.e. $H^p \neq \frac{1}{N}J_{N\times N}$ where $J_{N\times N}$ represents a matrix of ones of size $N \times N$. However, this initialization is not so critical in the tested datasets we have used, and the algorithm tends to converge to equivalent results given several initialization of this single non-uniform H.

[5] depending on the algorithm K we use this step may be removed.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

The pseudo-code references two additional functions. The first one: $\cdot = enhanceEntropy(\cdot)$, is related to a well-documented property that a Markov chain may have, the ergodicity. A Markov chain having this property tends to converge to a system where any state can be accessed by any other state with uniform distribution. From the computational point of view, this is equivalent to say that the product of two stochastic matrices may tend to converge to a stationary distribution[6]. We have observed the same behavior in product $(F^{p,q})^{t+1} \cdot (H^q)^t$. Considering that result of function K^{step}, contains a certain degree of discretization to a permutation matrix, we are interested in maintaining this degree of discretization in the resulting product of $(F^{p,q})^{t+1} \cdot (H^q)$. In this case, we use the entropy as indicator of the degree of discretization. Thus, $enhanceEntropy$ applies a method similar to the Softmax (Bridle 1990) algorithm by which approaches the entropy of $H_1 = (F^{p,q})^{t+1} \cdot (H^q)^t$ to the entropy of $H_2 = (F^{p,q})^{t+1}$. Using this method, the problem is approximated to the problem of finding value H_1' which minimizes the distance Δ[7] $= \|H_1' \circ log(H_1')\|_1 - \|H_2 \circ log(H_2)\|_1$ with the restrictions that $\Lambda(H_1') \cong \Lambda(H_1)$, where Λ represents a linear assignment solver like the Hungarian method. The entropy of matrix H_1 is enhanced by approaching it to a discretized version using parameter β:

$$H_1' = \frac{e^{\beta \cdot H_1}}{\|e^{\beta \cdot H_1}\|_1} \tag{2.20}$$

It is clear that the Softmax is often used to compute in a continuous way the maximum of a vector. Even though we can't ensure that $\Lambda(H_1') = \Lambda(H_1)$, we expect that using (2.20), in some sense, matrix H_1' converges to a permutation matrix related to the initial H_1 matrix. In fact, this assumption is in practice true for low values of β. This approach to solve the problem seems to give good results in practice. However, some other function targeted to the same objective could be used.

A naive approach to compute the best β value could be a binary search algorithm to find $\beta \in [1..\beta_{max}]$ which minimizes Δ . Its computational cost would be $O(log\ N)$.

The second function referenced in Algorithm 2-1, i.e. $\cdot = normalize(\cdot)$ guarantees that the matrix is doubly stochastic by applying the method in (Sinkhorn 1964). At the end of the algorithm, the CL is obtained through a discretization process Λ (Kuhn 1955; Munkres 1957) of H.

The cost of the algorithm is lineal with respect to the number of iterations. Each iteration computes $(P \cdot (P-1))$ calls to the function K^{step}.

[6] with some restrictions, and special systems.
[7] Operation ° indicates Hadamard product.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Considering the cost of K^{step} $O(N^4)$ if K is the *Graduated Assignment* algorithm (Gold and Rangarajan 1996). The global cost of the algorithm is $O(T \cdot P \cdot (P-1) \cdot N^4)$. From now to the rest of the document we will consider that K refers to the *Graduated Assignment* algorithm.

2.3 Common Labeling Graduated Assignment algorithm

The idea of this algorithm is to directly compute the common labeling by iteratively augmenting the compatibility of the current common labeling by continuously updating the set of probabilities H. To this end, we define C_{CL} given H using C_{CL} ((2.7) of chapter 2), C_{MI} ((2.1) of chapter 2) and the relation between F and H (2.7). In this way, the objective function of the common labeling becomes:

$$C_{CL} = \frac{1}{K_1} \sum_{\substack{p,q=1 \\ p \neq q}}^{P} \sum_{a=1}^{N} \sum_{i=1}^{N} \sum_{\substack{b=1 \\ b \neq a}}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\sum_{w_1=1}^{N} H^p[a,w_1] \cdot H^q[i,w_1] \cdot \left[\sum_{\substack{w_2=1 \\ w_2 \neq w_1}}^{N} H^p[b,w_2] \cdot H^q[j,w_2] \right] \right) \cdot C_{ai,bj}^{p,q} \quad (2.21)$$

$$K_1 = \frac{1}{P(P-1)N(N-1)}$$

Since our objective is to compute a common labeling, our new energy function depends on the probabilities H instead of F. In addition, the common labeling has to represent consistent and bijective isomorphisms between the involved graphs and the virtual node set. Consequently, we impose $a \neq b$, $i \neq j$ and $w_1 \neq w_2$. Figure 2-3, Figure 2-4 and Figure 2-5 show non-valid bijections forbidden by these restrictions.

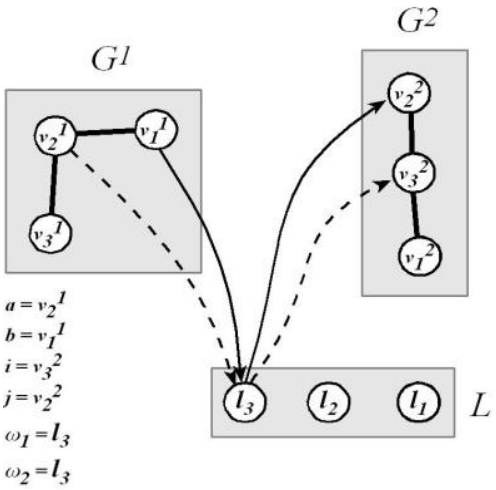


Figure 2-3: non-valid labeling: $w_1 = w_2$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

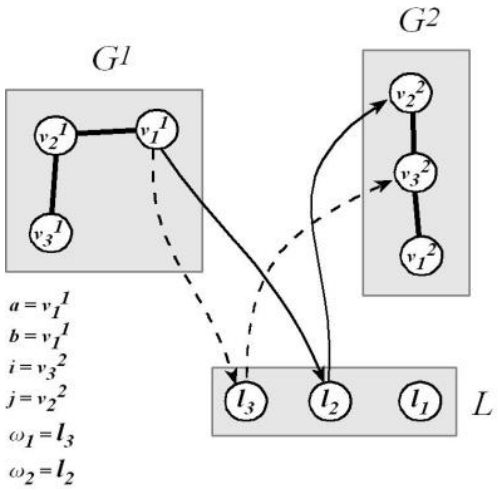


Figure 2-4: non-valid labeling: $a = b$.

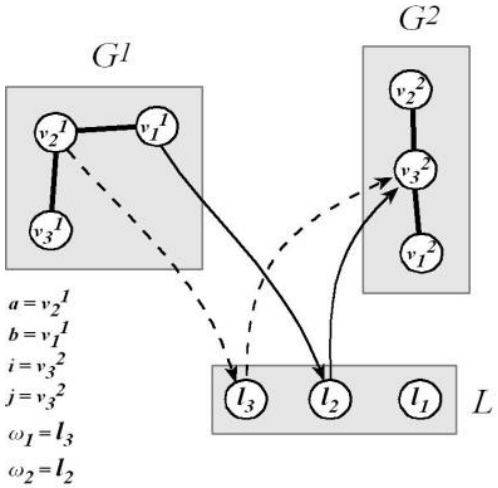


Figure 2-5: non-valid labeling: $i = j$.

Our optimization procedure is inspired by the Graduated Assignment (Gold and Rangarajan 1996) and maximizes C_{CL} (2.21).

2.3.1 Derivation of the algorithm

As commented above, the methodology we present applies a similar procedure than the Graduated Assignment algorithm (Gold and Rangarajan 1996) to compute an approximate solution for the common labeling problem. We start our development by computing the Taylor series expansion of the compatibility function (2.21) at the initial condition $(H^p)^t$,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

in our case, we focus on compatibility instead of energy. The approximation turns to be:

$$\begin{aligned} C_{CL} \approx C'_{CL} = \\ = \frac{1}{K_1}\sum_{p=1}^{P}\sum_{\substack{q=1\\q\neq p}}^{P}\sum_{a=1}^{N}\sum_{i=1}^{N}\sum_{\substack{b=1\\b\neq a}}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\left(\sum_{w_1=1}^{N}(H^p[a,w_1])^t\cdot(H^q[i,w_1])^t\right. \\ \left.\cdot\left[\sum_{\substack{w_2=1\\w_2\neq w_1}}^{N}(H^p[b,w_2])^t\cdot(H^q[j,w_2])^t\right]\right)\cdot C_{ai,bj}^{p,q} + \\ + \frac{1}{K_1}\sum_{p=1}^{P}\sum_{a=1}^{N}\sum_{w_1=1}^{N}Q_{a,w_1}^{p}(H^p[a,w_1] - (H^p[a,w_1])^t) \end{aligned} \tag{2.22}$$

where,

$$\begin{aligned} Q_{a,w_1}^{p} = \left.\frac{\partial C_{CL}}{\partial H^p}\right|_{H^p=(H^p)^t} \\ = \sum_{\substack{q=1\\q\neq p}}^{P}\sum_{i=1}^{N}\sum_{\substack{b=1\\b\neq a}}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}(H^p[i,w_1])^t \\ \cdot\left[\sum_{\substack{w_2=1\\w_2\neq w_1}}^{N}(H^p[b,w_2])^t\cdot(H^q[j,w_2])^t\right]\cdot C_{ai,bj}^{p,q} \end{aligned} \tag{2.23}$$

And in a similar manner to the Graduated Assignment algorithm (Gold and Rangarajan 1996) we obtain:

$$\begin{aligned} arg\max_{\Psi\in CL} C_{CL}(|\Gamma|,\Psi) \approx arg\max_{\Psi\in CL} C_{CL}(|\Gamma|,\Psi) = \\ = arg\max_{\Psi\in CL}\sum_{p=1}^{P}\sum_{a=1}^{N}\sum_{w_1=1}^{N}Q_{a,w_1}^{p}\cdot H^p[a,w_1] \end{aligned} \tag{2.24}$$

Thus, we can deduce that maximizing C_{CL} is equivalent to maximizing (2.24) which could be interpreted as maximizing the assignation with respect to the gradient of the function at point $(H^p)^t$. It is worth commenting that for correctness in the derivatives, restrictions $a \neq b, i \neq j$ and $w_1 \neq w_2$, in (2.21), are important to derivative (2.22). In case no restrictions were applied, second order terms would appear.

2.3.2 The algorithm

Algorithm 2-2 maximizes (2.22) and it returns a common labeling ψ given a set of graphs Γ. We impose H^1 to be the constant assignation throughout the iterative process. This requirement is necessary because the virtual node set does not contain any type of attributes or structure. Forcing the nodes of G^1 to concrete nodes of the virtual node set L, we impose all the other graphs to label each other according to this prior labeling. Without loss

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

of generality we impose H^1 to be the identity matrix. However, we could impose any other single H^q to any other permutation matrix having equivalent results, nevertheless to be coherent with (2.6) of chapter 2 we prefer to use this static assignation. Despite this restriction, the other probability matrices can be initialized to any stochastic matrix. We propose to initialize the matrices using a uniform probability. That is, $H^p = \frac{1}{N} J_{N\times N}$, $\forall\, p = \{2..P\}$.

Algorithm 2-3 computes function $Compute\_Q$ and obtains all Q^p_{a,w_1}. The computational cost is in general $O(P^2 \cdot N^6)$. Consequently, the total cost of Algorithm 2-2 is $O(T \cdot P^2 \cdot N^6)$, where T refers to the number of iterations of loops in lines 69 and 71 of Algorithm 2-2. We see that the main drawback of the algorithm is still the high computational cost of function $Compute\_Q$. With the aim of reducing this computational cost, in the next section we present an algorithm that obtains an approximation of Q^p_{a,w_1}.

66 ***Algorithm*** *Graduated Assignment common labeling (Γ)*
67 $H^i = initializeCL();$
68 $\beta = \beta_0$
69 *repeat until* $\beta > \beta_f$
70 $t = 0$
71 ***repeat until*** H *converges* ***or*** $t > I_0$)
72 $Q = Compute\_Q(H, \Gamma)$
73 $H^p[a, w_1] = e^{\beta \cdot Q^p_{a,w_1}}, 2 \le p \le P, 1 \le a, w_1 \le N$
74 ***repeat until*** H *converges* ***or*** $t_2 > I_1$
75 $H^p[a, w_1] = {H^p[a, w_1]}/{\sum_{x=1}^{N} H^p[x, w_1]}, 2 \le p \le P, 1 \le a, w_1 \le N$
76 $H^p[a, w_1] = {H^p[a, w_1]}/{\sum_{x=1}^{N} H^p[a, x]}, 2 \le p \le P, 1 \le a, w_1 \le N$
77 ***end***
78 $t = t + 1$
79 ***end***
80 $\beta = \beta \cdot \beta_r$
81 *end*
82 *Returns* Ψ

Algorithm 2-2: Graduated Assignment Common Labeling algorithm.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012


```
83  Algorithm Compute_Q( H, Γ)
84     for all G^p ∈ Γ
85        for a in all nodes in G^p
86           for w_1 in all nodes in L
87              Q^p_{a,w_1} = 0
88              for all G^q ∈ {Γ − G^p}
89                for i in all nodes in G^q
90                   for b in all nodes in G^p, b ≠ a
91                      for j in all nodes in G^q, i ≠ j
92                         v_2 = 0
93                         for w_2 in all nodes in {L − w_1}
94                            v_2 = v_2 + H^p[b, w_2] · H^q[j, w_2]
95                         end
96                         Q^p_{a,w_1} = Q^p_{a,w_1} + H^q[i, w_1] · v_2 · C^{p,q}_{ai,bj}
97                      end
98                   end
99                end
100           end
101        end
102     end
103  Returns Q
```

Algorithm 2-3: calculus of Q.

	Description	Value
β_0	Start value for Graduated Assignment control parameter	0.05
β_r	Increment rate for Graduated Assignment control parameter	1.075
β_f	Maximum value for Graduated Assignment control parameter	150
I_0	Number of iterations with the same control parameter value	4
I_1	Maximum number of iterations for Sinkhorn method	30

Table 2-1: Algorithm 2-2 parameter, summary and reference values.

Table 2-1 shows the description of the parameters of the algorithm presented in Algorithm 2-2 and a reference value. The performance and the run time of the algorithm depend highly on some of these parameters' values. All parameter values but β_0 and β_f are set to the same values of the original article (Gold and Rangarajan 1996) suggests. The values of β_0 and β_f are taken lower and higher respectively to obtain a good solution despite the initial values of H^p.

I_0 and I_1 are quite independent of the application and they can be considered more or less generic. However, the values of β_0 and β_f are application dependant and if they are not correctly tuned, the algorithm could return a not well approximated common labeling. To illustrate the problem, consider that we want to obtain the common labeling of three different sets Γ^1, Γ^2, Γ^3. We could suppose that cardinality of these sets is

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$|\Gamma^1| = 3$, $|\Gamma^2| = 10$, $|\Gamma^3| = 20$. For this concrete example, we consider that all of the attributed graphs in Γ^1, Γ^2 and Γ^3 are the same one. In the current example we used the first element of class V of the Letter Dataset (Riesen and Bunke 2008). The order of the graph is low, it has only 3 nodes. For each set, we compute the values of $Q^2_{a,i}, 1 \leq a, i, \leq 3$. Table 2-2, Table 2-3 and Table 2-4 show the values obtained at the first step of the algorithm.

Q^2 using Γ^1		
1.038	0.464	0.460
0.501	1.296	0.386
0.498	0.387	1.306

Table 2-2: Q^2 using 3 attributed graphs.

Q^2 using Γ^2		
3.33	2.75	2.75
3.05	3.84	2.93
3.05	2.94	3.86

Table 2-3: Q^2 using 10 attributed graphs.

Q^2 using Γ^3		
6.60	6.02	6.02
6.69	7.48	6.57
6.71	6.59	7.51

Table 2-4: Q^2 using 20 attributed graphs.

Note that the values of $Q^2_{a,i}$ are at least dependent on the number of attributed graphs in the set. Figure 2-6 shows $e^{\beta_0 \cdot x}$, $0 \leq x \leq 8$. The three grey columns represent the range of values of Table 2-2, Table 2-3 and Table 2-4 respectively. In the case of Γ^3, $e^{\beta_0 \cdot x}$ has high values, consequently the exponential function is too discriminative at the first step β_0 of the algorithm. If we desire to automatically calibrate β_0 and β_f, we propose to normalise matrices Q^p before applying the exponential function, in this way $e^{\beta_0, x}$ return values independent of the application. Another solution which was experimentally proven, and seems to work quite well, is to take $\beta_0 = 0.05/P$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

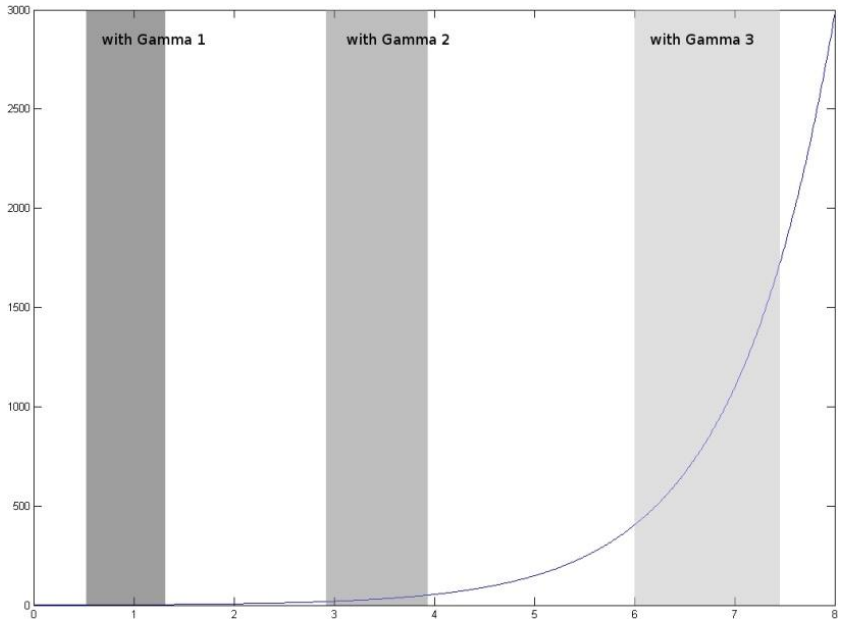


Figure 2-6: function $e^{\beta_0 \cdot x}$. Grey columns represent the range of values in Table 2-2, Table 2-3 and Table 2-4 respectively

2.3.3 Reduction of the computational cost of $\boldsymbol{Compute\_Q}$

Algorithm 2-4 shows an algorithm that computes an approximation of (2.23), the approximation is computed by relaxing constraint $w_1 \neq w_2$ in (2.23). It can be seen that, with graphs of reasonable number of nodes, the noise introduced by the non-valid isomorphism when $w_1 = w_2$ is not significant. Given a graph set Γ with graphs of order N, the percentage of non-valid isomorphisms when $w_1 = w_2$ is given by N/N^2. Note that, for instance, with $N = 20$ the percentage of non-valid isomorphisms taken into account is just 5%. Using this relaxation, Q can be sub-optimally computed in $O(P^2 \cdot N^4)$ instead of $O(P^2 \cdot N^6)$. Consequently, the final cost for Algorithm 2-2 becomes $O(T \cdot P^2 \cdot N^4)$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012


```
Algorithm Approx_Q( H, Γ)
    for all G^p ∈ Γ
        Q^p = 0_{N×N}
        for all G^q ∈ {Γ − G^p}
            M^{p,q} = H^p · (H^q)^T
            for a in all nodes in G^p
                for i in all nodes in G^q
                    v_1 = 0
                    for b in all nodes in G^p, b ≠ a
                        for j in all nodes in G^q, i ≠ j
                            v_1 = v_1 + M^{p,q}[b,j] · C^{p,q}_{ai,bj}
                        end
                    end
                    for w_1 in all nodes in L
                        Q^p_{a,w_1} = Q^p_{a,w_1} + v_1 · H^q[i,w_1]
                    end
                end
            end
        end
    end
Returns Q
```

Algorithm 2-4: calculus of $Approx\ Q$.

2.3.4 Statistical evaluation of the convergence

The convergence of the Graduated Assignment (Gold and Rangarajan 1996) has been studied (Rangarajan, Yuille et al. 1999). Since our algorithm follows the same optimization procedure we assume that the convergence properties are inherited. In this section, with the aim of a better understanding the proposed algorithm, we experimentally analyze its convergence. We study two features of the algorithm. The first aims to analyze how the algorithm tends to decrease the energy at each step. The second aims to analyze how it tends to converge to a stable solution. The dataset used in this study is the same as the one used in the experimental validation (see section 4.3 of chapter 7). The values we show are the average of 1050, 1080 and 1584 executions of the algorithm in Algorithm 2-2 (with Algorithm 2-4) using the synthetic, Letter and GREC dataset, respectively. In total, we performed 3714 experiments.

Figure 2-7 and Figure 2-8 show two examples of the value of energy $E^{CL} = -C_{CL}$ (2.21) at each step and throughout the execution of the algorithm. For each example, the evolution of the energy is different, although they do have similar shapes. We have parameterized these shapes at 5 intervals; the intervals are defined as {*1%*, *24%*, *50%*, *24%* and *1%*} of the E^{CL} final value. Each interval represents a part of the shape with equal characteristics. Interval 1: initial non-sloping part. Interval 2: curvature. Interval 3: central high-sloping part. Interval 4: curvature. Interval 5: final non-sloping part. Figure 2-9 shows the mean angle of the intervals of the 3714 experiments.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

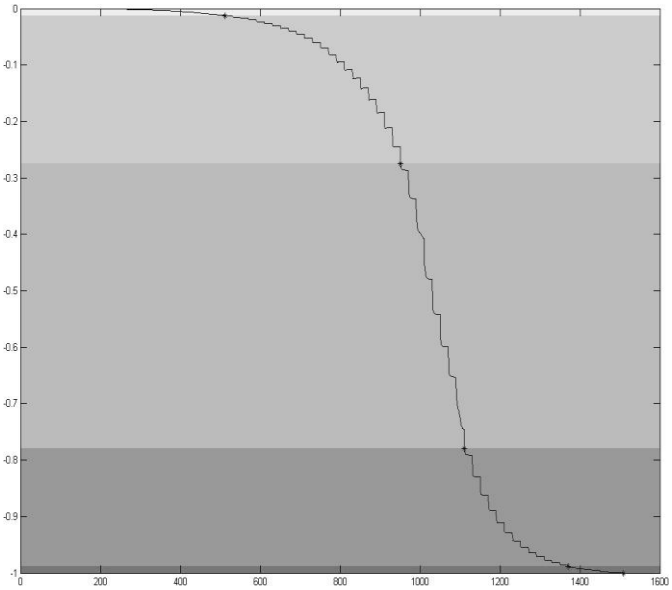


Figure 2-7: interval description.

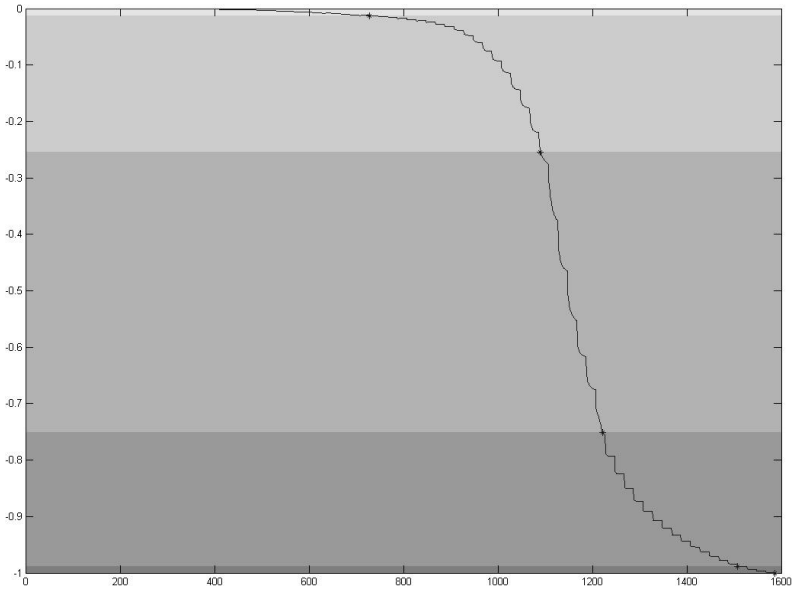


Figure 2-8: interval description.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

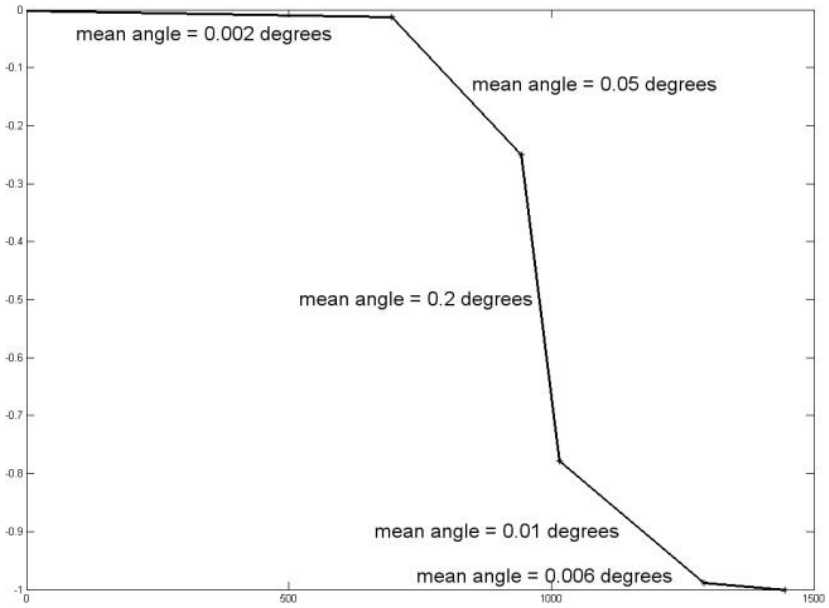


Figure 2-9: mean shape discretized in the 5 intervals.

To evaluate how the algorithm reduces energy at each step we present Figure 2-11 and Figure 2-10. Figure 2-11 shows the probability of having a concrete percent of E^{CL} reduction at a concrete interval. Figure 2-10 shows the probability of having a concrete percent of E^{CL} increment in every particular interval. These histograms have been computed using the sum of values at each interval which reduces (Figure 2-11) or increases (Figure 2-10) the energy. Although we see in the figures that the probability of observing increments of E^{CL} is not low when the size of increment is small, the mean value of decrements is much smaller than the mean value of increments. We can see that, with high probability, the increments are likely to appear when slope is low, in other words at the first and last interval. However, these increments are approximately 60 times lower than the decrements. Value of intervals two, three and four show the increments are also much smaller than the decrements.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

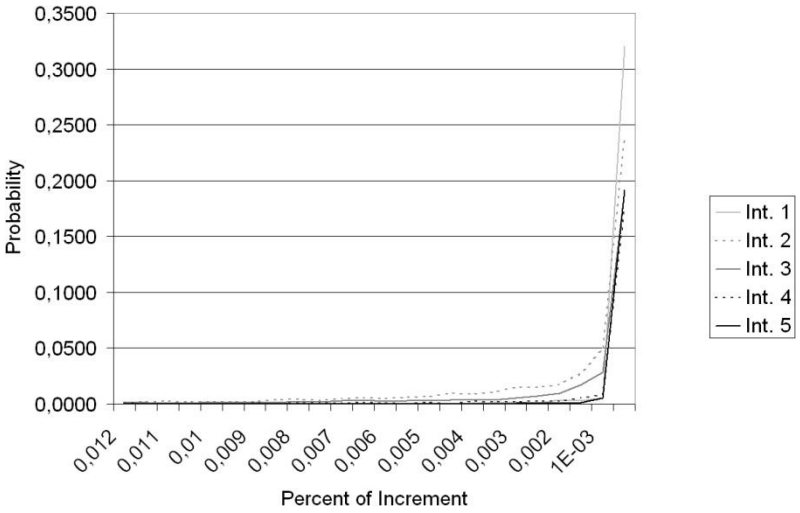


Figure 2-10: increment of energy.

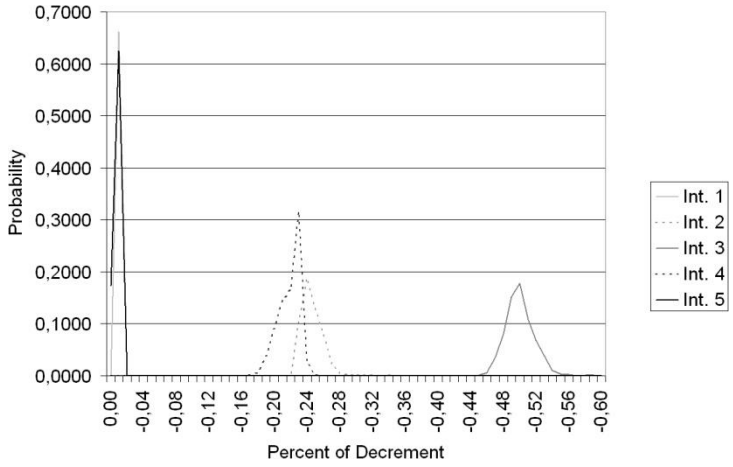


Figure 2-11: decrement of energy.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

3. COMPUTING THE COMMON LABELING THROUGH DOMINANT SETS[8]

The last method, we propose, to compute a common labeling is based on the association graph framework detailed in chapter 2. The main idea is based on encoding each possible common labeling solution as a clique in the so called association graph. Equivalently to the algorithm used to compute the graph edit distance for a pair of graphs, since we are focused in the error tolerant graph matching, and in particular in the graph edit distance, we do not rely on computing the maximum/maximal clique of the association graph but on computing the dominant set.

To fulfill structural and transitive properties of the common labeling, we use the same idea detailed in Section 2.1. That summarizing, labels each graph to a virtual node set L. Several definitions for the association graph can be considered. However, we propose to define the association graph as follows.

Definition 3-1: given a set of graphs $|\Gamma| = \{G^1, \dots, G^P\}$, we define the association graph $AG = (\tilde{V}, \tilde{E}, \tilde{\gamma}_v, \tilde{\gamma}_E)$ as a tuple of four elements where $\tilde{V} = \bigcup_{p=1}^{P} V^p \times \bigcup_{p=1}^{P} V^p \times L$ represents the node set, $\tilde{E} \subseteq \tilde{V} \times \tilde{V}$ represents the edge set, $\tilde{\gamma}_v : \tilde{V} \to \mathbb{R}$ assigns a real value to each vertex and $\tilde{\gamma}_E : \tilde{E} \to \mathbb{R}$ assigns a real value to each edge. □

In our case, each node of the association graph relates two nodes, let's say v_a^p and v_i^q to a single element in the virtual node set L. Consequently, each edge of the association graph relates an edge $(v_a^p, v_b^{p'})$ to another edge $(v_i^q, v_j^{q'})$ labeled to the same pair of nodes of the virtual set. Note that a priori p could be different than p' and q different than q'.

Definition 3-2: given a solution of the dominant set objective function:

$$\max ds(\boldsymbol{x}) = \sum_{i=1}^{|\tilde{V}|} \sum_{j=1}^{|\tilde{V}|} x_i \cdot x_j \cdot \tilde{\gamma}_E(i,j) \qquad subject\ to\ \boldsymbol{x} \in \Delta^{|\tilde{V}|} \tag{3.1}$$

where $\Delta^{|\tilde{V}|} = \{x_i \in \mathbb{R} \mid x_i \geq 0, \|x\|_1 = 1\}$. We denote the dominant set induced by the solution vector $\boldsymbol{x}$ as $AG_{\boldsymbol{x}} = (\tilde{V}_{\boldsymbol{x}}, \tilde{E}_{\boldsymbol{x}}, \tilde{\gamma}_{v_{\boldsymbol{x}}}, \tilde{\gamma}_{E_{\boldsymbol{x}}})$ where $\tilde{v}_i \equiv \left(v_a^p, v_i^q, l_k\right) \in \tilde{V}_{\boldsymbol{x}}$ if $x_i > 0$, and, edges of $E_{\boldsymbol{x}}$ belong to $\tilde{E}_{\boldsymbol{x}}$ if both terminal nodes belong to $\tilde{V}_{\boldsymbol{x}}$.

Definition 3-3: We define the consistent multiple isomorphism given $AG_{\boldsymbol{x}}$ as:

[8] This work has been done with the collaboration of Nicola Rebagliati and Marcello Pelillo during my stay at *Università "Ca' Foscari" di Venezia*.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$F^{p,q}[a,i] = \begin{cases} 1 & if\ \exists\ k = \{1..N\} \mid \left(v_a^p, v_i^q, l_k\right) \in \tilde{V}_{\boldsymbol{x}} \\ 0 & otherwise \end{cases} \tag{3.2}$$

Definition 3-4: We define the matching matrix H^p given $AG_{\boldsymbol{x}}$:

$$H^p[a,k] = \begin{cases} 1 & if\ \exists\ i = \{1..N\} \mid \left(v_a^p, v_i^q, l_k\right) \in \tilde{V}_{\boldsymbol{x}} \\ 0 & otherwise \end{cases} \tag{3.3}$$

3.1 Definition of the association matrix

Considering the initial definitions of this section we see that functions $\tilde{\gamma}_v$ and $\tilde{\gamma}_E$ are of crucial importance in the definition of the problem we face. These functions will give costs to the edges of the association graph, and consequently the maximum clique (or dominant set in our case) will be computed relying on this information. Considering this reasoning, we compute the weights of the association graph in the following way:

$$\tilde{\gamma}_E\left((a_1,b_1,l_1),(a_2,b_2,l_2)\right) = \begin{cases} 0^{9} & \left(\begin{pmatrix}(a_1,a_2 \in G_i \wedge a_1 \neq a_2)\vee\\(a_1,b_2 \in G_i \wedge a_1 \neq b_2)\vee\\(b_1,a_2 \in G_i \wedge b_1 \neq a_2)\vee\\(b_1,b_2 \in G_i \wedge b_1 \neq b_2)\end{pmatrix} \wedge (l_1 = l_2)\right) \vee \\ & \vee \left(\begin{pmatrix}(a_1,a_2 \in G_i \wedge a_1 = a_2)\vee\\(a_1,b_2 \in G_i \wedge a_1 = b_2)\vee\\(b_1,a_2 \in G_i \wedge b_1 = a_2)\vee\\(b_1,b_2 \in G_i \wedge b_1 = b_2)\end{pmatrix} \wedge (l_1 \neq l_2)\right) \vee \\ & \vee (a_1 = b_1 \wedge a_2 = b_2 \wedge l_1 = l_2) \vee (a_1,b_1 \in G_i) \vee \left(a_2,b_2 \in G_j\right) \\ C_{a_1,b_1,a_2,b_2} & otherwise\ if\ a_1,a_2 \in G_i \wedge b_1,b_2 \in G_j, i \neq j \\ 1 & otherwise \end{cases} \tag{3.4}$$

the first row assigns 0 value to the edge of the association graph where the labeling violates the restrictions of the common labeling shown in Figure 2-3, Figure 2-4 and Figure 2-5, self labeling and also to loop edges. Since this will indicate there is no edge between both nodes, these two nodes cannot form part of the same clique (or dominant set) and cannot form part of the same solution. The second value represents the cost of matching an edge (a_1, a_2) of graph G_i to edge (b_1, b_2) of graph G_j, this may have different definitions depending on the objective function we aim to optimize, we use the graph edit distance cost given in chapter 2. Eventually, last cost $\tilde{\gamma}_E\left((a_1,b_1,l_1),(a_2,b_2,l_2)\right) = 1$ is assigned when the labeling is consistent but nodes belong to different graphs and consequently the cost of matching the respective edge cannot be computed. See that, due to restrictions on (3.4)

[9] A low value c might have the same results as zero and increase the efficiency of the optimization algorithm

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

, transitivity between nodes in the consistent multiple isomorphism that we compute, are implicitly fulfilled in the clique solutions since two nodes that belong to the same graph cannot be labeled to the same virtual element l_i. Consider as an example non-consistent labelings in Figure 2-2 in chapter 2. Three of the nodes in the association graph that should appear on the clique for that multiple isomorphism to be a solution are: (v_1^1, v_1^2, l_1), (v_1^1, v_1^3, l_1) and (v_1^2, v_2^3, l_1). See that labeling $v_2^3, v_1^3 \in G_3, v_2^3 \neq v_1^3 \wedge l_1 = l_1$ and so nodes two and three would have zero cost and cannot form part of the same solution. Two solutions to solve the problem would be to set second element of second node to v_2^3 or second element of third node to element v_1^3.

3.2 Minimization of the Multiple Graph Edit Distance

In order to minimize the common labeling using the graph edit distance, we compute the cost of labeling two edges in the association graph as:

$$C_{a_1b_1,a_2b_2} = (1-\alpha)\cdot \left(1-\frac{1}{2\mathcal{C}_M}\left(c_1\left(C_V(a_1,b_1)+C_V(a_2,b_2)\right)+c_2\left(\mathcal{C}_E\left((a_1,a_2),(b_1,b_2)\right)\right)\right)\right) + \alpha \qquad (3.5)$$
$$c_1 = \frac{\left(NP(P-1)\right)^2}{N-1}, c_2 = \left(NP(P-1)\right)^2$$

c_1 and c_2, represent two constants values addressed to allow the system to compute the graph edit distance and α, equivalently to section 1 of chapter 4 is a parameter addressed to move the solution to the barycenter of a simplex face. Parameters c_1 and c_2 will be deduced at the end of the section. To reduce the problem to the objective function of the common labeling, we represent each possible consistent multiple isomorphism in the simplex as a vector where each position corresponds to a node in the association graph.

Definition 3-5: given a common labeling φ, we define $x_\varphi = \tilde{V} \to \mathbb{R}^+$ be a barycenter vector as:

$$x_\varphi(u,v,l) = \begin{cases} \frac{1}{NP(P-1)} & if \quad \begin{matrix} H^{G_i}(v) = l \\ H^{G_j}(u) = l \\ v \in G_i, u \in G_j, i \neq j \end{matrix} \\ 0 & otherwise \end{cases} \qquad (3.6)$$

Considering Definition 3-5, we analyze the functional that dominant sets optimize. So, given a dominant set solution x_φ, and defining A_α as a matrix of $\tilde{V} \times \tilde{V}$ positions where $A_{\alpha_{i,j}} = \tilde{\gamma}_E(\tilde{v}_i, \tilde{v}_j)$. The functional becomes:

$$\langle \boldsymbol{x}_\varphi, A\boldsymbol{x}_\varphi \rangle = \sum_{i=1}^{|\tilde{V}|}\sum_{j=1}^{|\tilde{V}|} \boldsymbol{x}_{\varphi_i}\boldsymbol{x}_{\varphi_j}A_{\alpha_{i,j}} \qquad (3.7)$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

since positions that do not belong to the common labeling will add zero to the functional we exclude to those values. Since we know that the consistent multiple isomorphism will contain $NP(P-1)$ non-zero values, we know that $\left|\sigma(\boldsymbol{x}_\varphi)\right|_{\mathbf{1}} = NP(P-1)$. In addition we define $I = \{i | \boldsymbol{x}_{\varphi_i} \neq 0\}$. Thus,

$$\sum_{i=1}^{|\boldsymbol{x}_\varphi|}\sum_{j=1}^{|\boldsymbol{x}_\varphi|} \boldsymbol{x}_{\varphi_i}\boldsymbol{x}_{\varphi_j} A_{\alpha_{i,j}} = \frac{1}{\left(NP(P-1)\right)^2}\sum_{i\in I}\sum_{\substack{j\in I \\ j\neq i}} A_{\alpha_{i,j}} \tag{3.8}$$

substituting in (3.8) definition on (3.4) and (3.5), and considering mappings of (3.2) and (3.6) we get:

$$\begin{aligned}
\langle \boldsymbol{x}_\varphi, A_\alpha \boldsymbol{x}_\varphi \rangle = \\
&= \frac{1}{\left(NP(P-1)\right)^2}\sum_{p=1}^{P}\sum_{\substack{q=1 \\ p\neq q}}^{P}\sum_{a_1=1}^{N}\sum_{b_1=1}^{N}\sum_{\substack{a_2=1 \\ a_2\neq a_1}}^{N}\sum_{\substack{b_2=1 \\ b_2\neq b_1}}^{N} F^{p,q}[a_1,b_1]F^{p,q}[a_2,b_2]C_{a_1b_1,a_2b_2} + \\
&+ \frac{2}{\left(NP(P-1)\right)^2}\sum_{p_1=1}^{P}\sum_{p_2=p_1}^{P}\sum_{\substack{q_1=1 \\ q_1\neq p_1}}^{P}\sum_{\substack{q_2=1 \\ q_2\neq q_1 \\ q_2\neq p_2}}^{P}\sum_{\substack{a_1,a_2,b_1,b_2=1 \\ a_2\neq a_1, b_2\neq b_1}}^{N} F^{p_1,q_1}[a_1,b_2]F^{p_2,q_2}[a_1,b_2] + \\
&+ \frac{1}{\left(NP(P-1)\right)^2}\sum_{p_1=1}^{P}\sum_{\substack{p_2=1 \\ p_2\neq p_1}}^{P}\sum_{q_1=p2}\sum_{\substack{q_2=1 \\ q_2\neq q_1 \\ q_2\neq p_2}}^{P}\sum_{\substack{a_1,a_2,b_1,b_2=1 \\ a_2\neq a_1, b_2\neq b_1}}^{N} F^{p_1,q_1}[a_1,b_2]F^{p_2,q_2}[a_1,b_2] + \\
&+ \frac{1}{\left(NP(P-1)\right)^2}\sum_{p_1=1}^{P}\sum_{\substack{p_2=1 \\ p_2\neq p_1}}^{P}\sum_{\substack{q_1=1 \\ q_1\neq p_1 \\ q_1\neq p_2}}^{P}\sum_{\substack{q_2=1 \\ q_2\neq p_2 \\ q_2\neq q_1}}^{P}\sum_{\substack{a_1,a_2,b_1,b_2=1 \\ a_2\neq a_1, b_2\neq b_1}}^{N} F^{p_1,q_1}[a_1,b_2]F^{p_2,q_2}[a_1,b_2]
\end{aligned}$$

$$\begin{aligned}
\langle \boldsymbol{x}_\varphi, A_\alpha \boldsymbol{x}_\varphi \rangle = \\
&= \frac{1}{\left(NP(P-1)\right)^2}\sum_{p=1}^{P}\sum_{\substack{q=1 \\ p\neq q}}^{P}\sum_{a_1=1}^{N}\sum_{b_1=1}^{N}\sum_{\substack{a_2=1 \\ a_2\neq a_1}}^{N}\sum_{\substack{b_2=1 \\ b_2\neq b_1}}^{N} F^{p,q}[a_1,b_1]F^{p,q}[a_2,b_2]C_{a_1b_1,a_2b_2} + \\
&+2\frac{P(P-1)(P-2)N^2}{\left(NP(P-1)\right)^2} + \frac{P(P-1)^2N^2}{\left(NP(P-1)\right)^2} + \frac{P(P-1)(P-2)^2N^2}{\left(NP(P-1)\right)^2}
\end{aligned} \tag{3.9}$$

The first term of (3.9) corresponds to positions of the second row of (3.4) and the rest to positions of the third row of (3.4).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\begin{aligned}\langle \boldsymbol{x}_\varphi, A_\alpha \boldsymbol{x}_\varphi \rangle = \\ = \frac{(P(P-1)(1-\alpha+\alpha)+P(P-1)(2(P-2)+(P-1)+(P-2)^2))N^2}{\left(NP(P-1)\right)^2} + \\ + \frac{(\alpha-1)}{2\mathcal{C}_M} \sum_{p=1}^{P} \sum_{\substack{q=1 \\ p\neq q}}^{P} \sum_{a_1=1}^{N} \sum_{b_1=1}^{N} \sum_{\substack{a_2=1 \\ a_2\neq a_1}}^{N} \sum_{\substack{b_2=1 \\ b_2\neq b_1}}^{N} F^{p,q}[a_1,b_1]F^{p,q}[a_2,b_2] \cdot \\ \cdot \left[\frac{1}{N-1}\left(\mathcal{C}_V(a_1,b_1)+\mathcal{C}_V(a_2,b_2)\right)+\left(\mathcal{C}_E\left((a_1,a_2),(b_1,b_2)\right)\right)\right] = \\ = 1 + \frac{(\alpha-1)}{2C_M}\mathcal{C}_{MI}(\Gamma,\phi)\end{aligned} \tag{3.10}$$

We see that, equivalently to the case of minimizing the graph edit distance, using the dominant set framework detailed in chapter 4, the computation of $\langle \boldsymbol{x}_\varphi, A_\alpha \boldsymbol{x}_\varphi \rangle$ is proportional to the functional of the consistent multiple isomorphism defined in chapter 2. Consequently, we know that:

$$\langle \boldsymbol{x}^*, A_\alpha \boldsymbol{x}^* \rangle \geq 1 + \frac{(\alpha-1)}{2C_M}\mathcal{C}_{MI}(\Gamma,\boldsymbol{\varphi}^*) \tag{3.11}$$

However, once again, solutions given by solution vector $\boldsymbol{x}$ may not correspond to bijections and usually, just a subset of matchings is returned. However, using theorem described in section 1 on chapter 4 and since every possible consistent multiple isomorphism is encoded as clique in the association graph, we can ensure that there is an $\bar{\alpha}$ such that the functional in Definition 3-2 returns a consistent multiple isomorphism. In addition, using the same theorem we can ensure that, for all $\alpha > \bar{\alpha}$ there exists a one-to-one correspondence between local solutions of the graph edit distance and maximal cliques in the association graph.

3.3 Algorithm to compute a consistent multiple isomorphism using the dominant set framework

The algorithm we propose to compute a suboptimal solution for the common labeling problem is on the same lines as the one proposed to compute the graph edit distance between two graphs. Thus, the algorithm sets a static α and computes several suboptimal solutions using several random initialization points either near the barycenter of the simplex or uniformly distributed on the simplex. We consider the output of the algorithm the best solution found. For self contentment of the chapter we include the pseudo-code of the algorithm:

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Algorithm *dominant set consistent multiple isomorphism* (Γ)
$\alpha = setAlpha()$
$itr = 0; minCost = \infty; min\varphi = \emptyset$
AG=constructAssociationGraph (Γ, α) *((3.4) and (3.5))*
for itr <maxItr
    $initX = -log\left(rand(N^3P^2, 1)\right)$
    $initX_i = \frac{1}{\sum_{j=1}^{N^3P^2} initx_j} initX_i$
    x=InImDymAlg($initX, AG$ *)*
    φ^{cons} *= extractBijections(x) ((3.2) or (3.3))*
    $c = \mathcal{C}_{CMI}(\Gamma, \varphi^{cons})$
    if $c < minCost \wedge isbijection(\varphi^{cons})$
        $minCost = c;\ min\varphi = \varphi^{cons}$
    end
    $itr = itr + 1$
end
Returns $min\varphi$

Algorithm 3-1: pseudo-code of the dominant set graph edit distance algorithm.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 6

EVALUATION AND APPLICATIONS OF THE COMMON LABELING

1. DATASETS DESCRIPTION

In this section, we will review the datasets used in the evaluation of the algorithms presented in chapters 4 and 5.

1.1 Letter dataset

The first dataset, called Letter, was created at the University of Bern (Riesen and Bunke 2008). The dataset considers the fifteen capital letters of the Roman alphabet that are composed of straight lines, i.e. A, E, F, H, I, K, L, M, N, T, V, W, X, Y, X. The dataset contains three subsets with different distortion levels: low, med, high. In all tests only the high distortion subset was considered. For each subset and a particular letter, 150 examples are given.

The letters are converted into graphs by assigning straight lines to edges and terminal points of the lines to nodes. Nodes are attributed with a two-dimensional attribute that represents the Euclidean position (x, y) of the terminal point in the plane. Figure 1-1 shows one example of letters A, E, H and M for the high distortion subset. Table 1-1 shows the basic statistics for the Letter dataset. The main characteristic of this dataset is the high distortion among elements of the same class.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

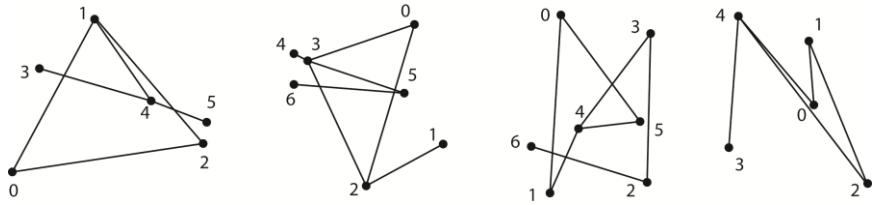


Figure 1-1**:** example of Letter dataset, letters A, E, H and M.

Property	Value
Number of classes	15
Elements per class	150
Maximum number of nodes	9
Minimum number of nodes	2
Mean number of nodes	4.6
Maximum number of edges	18
Minimum number of edges	0
Mean number of edges	9

Table 1-1**:** characteristics of the Letter dataset.

1.2 GREC dataset

The GREC dataset, created at the Universitat Autònoma de Barcelona (Riesen and Bunke 2008), is composed of 22 classes and 50 noisy graphs per class. Graphs represent symbols from architectural and electronic drawings. Each symbol is modeled using straight lines and, equivalently to the Letter dataset, images are converted into graphs by assigning a node to each junction or terminal point and an edge to each line. Nodes are attributed with the Euclidean position in the two-dimensional space corresponding to the location of the terminal point. Edges do not contain attributes. Several example images of this dataset are shown in Figure 1-2. In addition, graph representations of classes 11, 12 and 13 are shown in Figure 1-3. Table 1-2 summarizes the dataset characteristics.

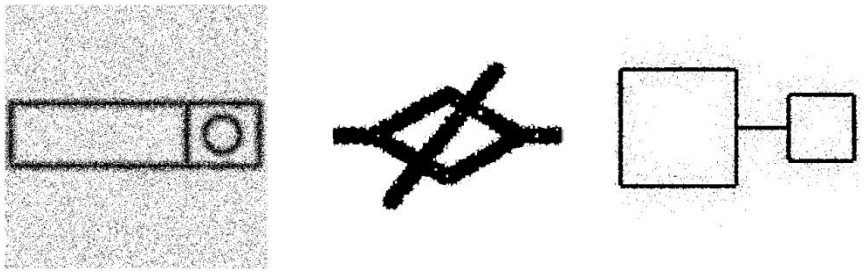


Figure 1-2: example images from where GREC dataset is extracted.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

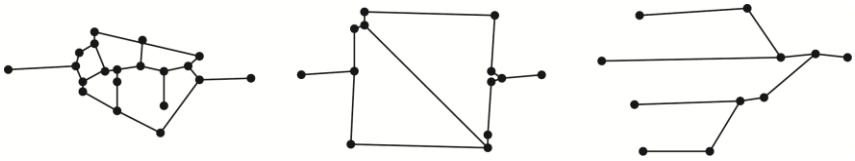


Figure 1-3: examples of GREC dataset. Examples of classes 11, 12 and 13.

Property	Value
Number of classes	22
Elements per class	50
Maximum number of nodes	24
Minimum number of nodes	4
Mean number of nodes	11
Maximum number of edges	58
Minimum number of edges	4
Mean number of edges	23

Table 1-2: characteristics of the GREC dataset.

1.3 Synthetic dataset

The third dataset, we used, is created synthetically, this allows us to experiment with different and controlled noise levels and fixed number of nodes per graph. Each class was created as follows. We randomly generate a base graph of N nodes with two-dimensional random attributes in the range $\Delta_v=[0..100, 0..100]$. Edges are defined by the Delaunay triangulation. Then, with this base graph, we created P other graphs by: 1) generating Gaussian noise at every node with standard deviation $\sigma \in [0,1]$, 2) removing $\nu = 100 \cdot \sigma$ percent of nodes randomly, 3) inserting $\nu = 100 \cdot \sigma$ percent of nodes (with random attributes) and 4) changing the state of $\nu = 100 \cdot \sigma$ percent of edges. Figure 1-4 shows, in diagram form, how to create each synthetic class. In addition, Figure 1-5 and Figure 1-6 show two synthetic classes. The first column of each example represents the base graph and the other columns represent three synthetically created graphs computed using this first base graph. Figure 1-5 was created using $N = 10$, $P = 3$, $\sigma = 0.5$ and $\nu = 50\%$, Figure 1-6 using $N = 10$, $P = 3$, $\sigma = 0.8$ and $\nu = 80\%$.

1.4 COIL

The COIL-100 database (Riesen and Bunke 2008) consists of images of 100 different objects (100 classes). Images of the objects are taken at

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

intervals of 5 degrees; therefore, there are 72 images per object. The Harris corner (Harris and Stephens 1988) detection algorithm is used to extract corner features from the images. Based on these corner points, a Delaunay triangulation is applied. The result of the triangulation is then converted into a graph by representing lines by undirected edges and ending points of lines by nodes. Nodes are labeled with a two-dimensional attribute identifying its position. Edges do not have attributes. In the performed tests, we will consider images taken at angles 0, 10, 20,... represent the test set and angles taken at angles 5, 15, 25,... represent the reference set. Some examples of the images of this dataset are shown in Figure 1-7. In addition, Table 1-3 summarizes the dataset characteristics.

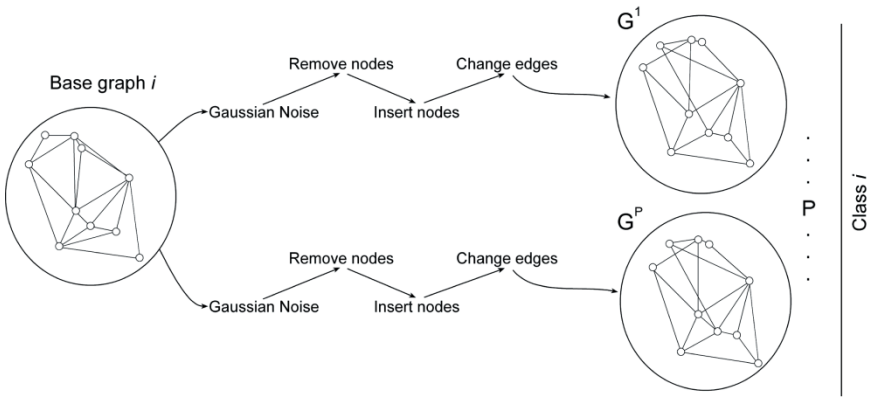


Figure 1-4: synthetic dataset construction procedure.

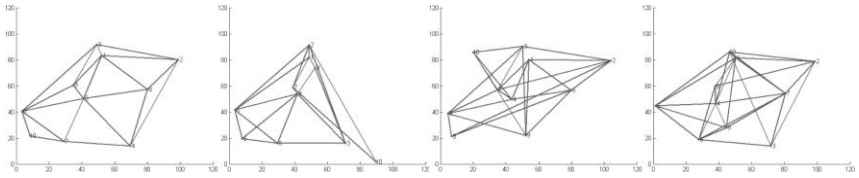


Figure 1-5: examples from Synthetic dataset created using $N = 10, P = 3, r = 0.5$ and $v = 50\%$.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

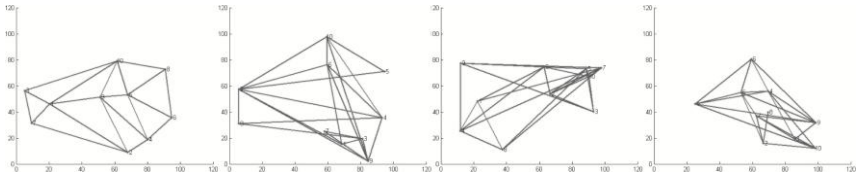


Figure 1-6: examples from Synthetic dataset created using $N = 10, P = 3, r = 0.8$ and $v = 80\%$.

Figure 1-7: COIL images of 100 different objects.

Property	Value
Number of classes	100
Elements per class	72
Maximum number of nodes	15
Minimum number of nodes	3
Mean number of nodes	12
Maximum number of edges	76
Minimum number of edges	6
Mean number of edges	54

Table 1-3: characteristics of the COIL dataset.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1.5 Fingerprint

Image fingerprints are converted into graphs by filtering the images, extracting regions and obtaining the ridges (Neuhaus and Bunke 2005). Ending points and bifurcation points of the ridges are represented by nodes. Undirected edges are inserted to link nodes that are directly connected through a ridge in the skeleton. Each node is labeled with a two-dimensional attribute giving its position. Edges do not have attributes. The original fingerprint database is based on the NIST-4 reference database of fingerprints (Watson and Wilson 1992). It consists of 2,800 fingerprint images totally out of the 4 classes arch (A), left (L), right (R), and whorl (W) from the Galton-Henry classification system. Each of the four classes has been equally divided into the test set and the reference set. An example of each class is given in Figure 1-8. In addition, Table 1-4 summarizes the dataset characteristics. In this dataset, elements are not homogeneously distributed per classes. Thus, Table 1-4 presents the number of elements per class.

Figure 1-8: Fingerprint examples, one per class.

Property	Value
Number of classes	4
Elements per class	$\|A\| = 667, \|L\| = 698, \|R\| = 721, \|W\| = 551$
Maximum number of nodes	26
Minimum number of nodes	2
Mean number of nodes	6
Maximum number of edges	44
Minimum number of edges	1
Mean number of edges	7

Table 1-4: characteristics of the fingerprint dataset.

1.6 Shapes 99

This dataset (Sebastian, Klein et al. 2004) contains shapes of objects and animals. The dataset contains 99 shapes uniformly distributed in 9 classes consisting of 11 shapes each. Shapes of both datasets are represented using shock graphs (Siddiqi, Shokoufandeh et al. 1999). Shock graphs are

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

constructed as section 2.5 of (Macrini 2003) shows. Each node of the shock graph is attributed with information regarding the *node type* and the *length* of the segment it represents. Edges connect nodes considering the shock graph hierarchy. The dataset if fully represented in Figure 1-9. Statistics on the number of nodes and edges are shown in Table 1-5.

1.7 Shapes 216

This dataset (Sebastian, Klein et al. 2004) contains shapes of objects and animals. The dataset contains 216 shapes uniformly distributed in 18 classes consisting of 12 shapes each. Shapes of both datasets are represented using shock graphs (Siddiqi, Shokoufandeh et al. 1999). Shock graphs are constructed as section 2.5 in (Macrini 2003) shows. Each node of the shock graph is attributed with information regarding the *node type* and the *length* of the segment it represents. Edges connect nodes considering the shock graph hierarchy. The dataset if fully represented in Figure 1-10. Statistics on the number of nodes and edges are shown in Table 1-6.

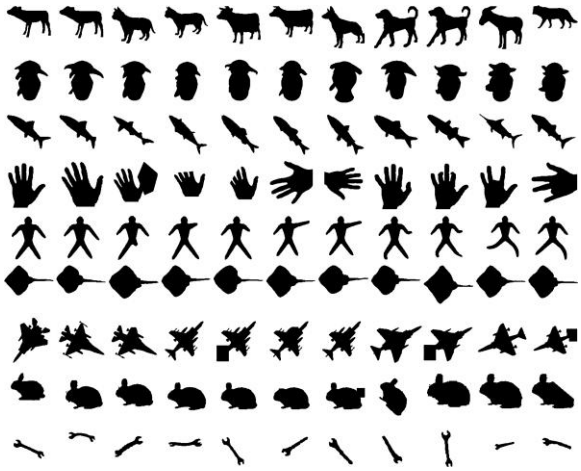


Figure 1-9: Shapes 99 dataset.

Property	Value
Number of classes	9
Elements per class	11
Maximum number of nodes	22
Minimum number of nodes	2
Mean number of nodes	11
Maximum number of edges	46
Minimum number of edges	2
Mean number of edges	22

Table 1-5: characteristics of the Shapes 99 dataset.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1.8 Feature Space Dataset

This dataset[10] is addressed to evaluate methods which aim to register a set of images. For each class the dataset contains a set of sequentially transformed images, including rotations, zoom and shear. We used three images sequences extracted from the *Image Matching: Planar Scenes* section. These image datasets are New York, Van Gogh and Asterix. The first dataset contains 35 images of the New York city. Figure 1-11 shows 4 sequential images of that dataset. The second dataset contains 17 images from a Van Gogh painting. Figure 1-12 shows a set of 4 sequential images of that dataset. The last dataset contains 17 images of drawings. Figure 1-13 shows a set of 4 sequential images of this last dataset.

Figure 1-10: Shapes 216 dataset.

[10] http://www.featurespace.org/

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Property	Value
Number of classes	18
Elements per class	12
Maximum number of nodes	39
Minimum number of nodes	2
Mean number of nodes	9
Maximum number of edges	90
Minimum number of edges	2
Mean number of edges	18

Table 1-6: characteristics of the Shapes 216 dataset.

Figure 1-11: sequence of images from New York dataset.

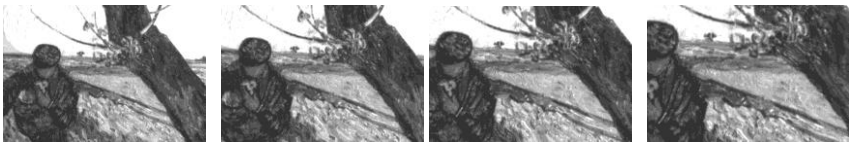


Figure 1-12: sequence of images from Van Gogh dataset.

Figure 1-13: sequence of images from Asterix dataset.

Images are modeled using a set of salient points. Salient points from each image have been extracted using the Harris operator (Harris and Stephens 1988). We thresholded each image to contain around 50 salient points.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

2. EVALUATION OF THE EDIT DISTANCE WITH DOMINANT SETS

To evaluate the method presented in chapter 4, addressed to compute the graph edit distance between two graphs, we compare the performance of the Dominant Set algorithms (DS) with the Graduated Assignment (GA) algorithm (Gold and Rangarajan 1996) and the Bipartite Graph Matching (BP) in (Riesen, Neuhaus et al. 2007). The dominant set algorithm builds the regularized association graph G_α and then run the InImDym dynamics (Bulo' and Bomze 2011) with 100 different initializations. After a training phase on a subset of the data, we set $\alpha = .8$ and $c = .45$ because these values empirically gave high probability of returning a dominant set supporting a full bijection between both graphs.

Since we wanted to evaluate the algorithms in different settings of the graph edit distance we do not, a priori, fix values for K_n (the vertex insertion/deletion cost) and K_e (the edge insertion/deletion cost); instead, we report the performance for several values.

We evaluate the competing algorithms over three datasets, previously presented in section 1: the GREC dataset, the Shapes 99 dataset and the Shapes 216. We present two different results. First results evaluate the cost value of the graph edit distance. The second ones relate the improvement that the new method achieves on a clustering application with the improvement on the graph edit distance computation. For the first experiment, we evaluate the expected matching error per assignation given two graphs. Considering two different bijections which have been computed using the same pair of graphs, the lower the error we get, the better is the bijection. To give a relative value, results are normalized by the number of assignations. The values presented are computed as follows:

$$\frac{C_{GED}(G^1, G^2, f)}{|V_1| + |V_2|} \tag{2.1}$$

where f corresponds to the bijection returned by any of the compared algorithms.

For each of the three datasets, two values are presented. The first value corresponds to the expected error per intra-class assignations, and the second value to the expected error per inter-class assignations. For the intra-class distance evaluation, we performed 4840 tests for the GREC dataset, 1089 for the shapes 99 and 2592 for the Shapes 216. For the inter-class, we performed 43596 tests with the GREC dataset, 8712 with the Shapes 99 dataset and 44064 with the shapes 216 dataset. We evaluate performance of the algorithms at points $(K_e, K_n)_1 = (22,22)$, $(K_e, K_n)_2 = (22,66)$, $(K_e, K_n)_3 = (66,22)$ and $(K_e, K_n)_4 = (66,66)$ for the GREC dataset and at points $(K_e, K_n)_1 = (5,5)$, $(K_e, K_n)_2 = (5,15)$, $(K_e, K_n)_3 = (15,5)$ and

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$(K_e, K_n)_4 = (15{,}15)$ for the Shape datasets. Results are shown in Table 2-1. Each cell of the table represents the mean expected error per assignation of all the tests. Lower values of each individual experiment are highlighted. We see that the algorithm presented in chapter 4 improves in all performed tests.

In addition, to the evaluation of the graph edit distance value, we evaluated the improvement on clustering a set of attributed graphs. We expect that achieving a better approximation of the graph edit distance leads to a better classification. For this experiment, we cluster 10 randomly chosen graphs per each class of the GREC Dataset (220 graphs), and all the graphs that compose the shapes datasets (99 and 216). To cluster the graphs, we used the dominant set peeling algorithm of (Pavan and Pelillo 2007). For each pair-wise distance matrix D computed with each of the algorithms we build a similarity matrix $W = \mathrm{e}\hat{}(-D\hat{}2/\sigma\hat{}2)$. The vertex with higher dominant set weight is chosen as the representer of the cluster. Table 2-2 presents the best results obtained for each individual algorithm. Different algorithm maximize results at different (K_e, K_n) points, these points correspond to:

$$(K_e, K_n)_{DS} = \{(66{,}66), (5{,}15), (15{,}15)\}$$
$$(K_e, K_n)_{GA} = \{(66{,}66), (5{,}15), (15{,}5)\}$$
$$(K_e, K_n)_{BP} = \{(66{,}66), (15{,}5), (5{,}15)\}$$

Columns labeled with *Val* contain intra-class sum of distances of each cluster. Columns labeled with *Err* indicate percentage of errors in clustering. We consider an error every element in the cluster which its class do not correspond to the class of the chosen *representer*. Results show significant improvement at all, except two, points on the sum of distances and best results on classification match those of the best one among (GA,BP).

		$(K_e, K_n)_1$		$(K_e, K_n)_2$		$(K_e, K_n)_3$		$(K_e, K_n)_4$	
		Intra	Inter	Intra	Inter	Intra	Inter	Intra	inter
GREC	DS	**18.9**	**42.3**	**21.2**	**72.9**	**22.4**	**80.8**	**23.6**	**105.3**
	GA	20.1	44.9	21.4	75.6	22.5	83.4	23.8	107.8
	BP	20.6	46.0	22.7	76.1	27.1	98.6	28.3	120.5
Shapes 99	DS	**3.31**	**5.58**	**4.36**	**8.37**	**5.35**	**10.0**	**6.30**	**12.5**
	GA	3.37	5.76	4.39	8.44	5.43	10.1	6.41	12.7
	BP	4.50	6.77	5.58	9.59	9.49	14.4	10.5	16.9
Shapes 216	DS	**3.74**	**5.81**	**4.99**	**8.70**	**6.21**	**10.44**	**7.32**	**12.98**
	GA	3.90	6.13	5.06	8.90	6.39	10.73	7.46	13.24
	BP	4.83	6.91	6.09	9.84	9.74	14.07	10.86	16.62

Table 2-1: Graph edit distance results using inter and intra class comparisons.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

		$(K_e, K_n)_{DS}$		$(K_e, K_n)_{GA}$		$(K_e, K_n)_{BP}$	
		Val	Err	Val	Err	Val	Err
GREC	DS	**56684**	**0.09**	**56684**	**0.09**	**56684**	**0.09**
	GA	58597	**0.09**	58597	**0.09**	58597	**0.09**
	BP	70908	0.1	70908	0.1	70908	0.1
Shapes 99	DS	**5499**	**0.24**	**5499**	**0.24**	6402	0.36
	GA	6607	0.27	6607	0.27	**6231**	0.38
	BP	9884	0.32	9884	0.32	6541	0.30
Shapes 216	DS	**18429**	**0.44**	13291	0.50	**17150**	0.59
	GA	20292	0.52	**12689**	0.47	17740	0.52
	BP	27595	0.50	14899	0.55	22573	**0.44**

Table 2-2: results of clustering.

3. EVALUATION OF THE COMMON LABELING ALGORITHMS

To evaluate the common labeling algorithms, we present several tests. Each test corresponds to the evaluation of a type of methods presented in chapter 5: methods to compute the common labeling though a consistent multiple isomorphism and methods that compute the common labeling directly. Eventually, the best method is compared with the dominant set methodology.

3.1 Evaluation of algorithms that compute a consistent multiple isomorphism

In this section, we present evaluation of the algorithms that compute the common labeling by computing a multiple isomorphism or that compute the consistent multiple isomorphism directly. To this aim, we evaluate the algorithms presented in section 1.1.1, 1.1.2, 2.2.1 of chapter 5. We compare its performance with the Average Alignment Common Labeling method (presented in section 2.2 of chapter 5) that directly computes the common labeling and with the method presented by Bonev. in (Bonev, Escolano et al. 2007; Lozano 2008).

All these algorithms share in common that they keep or obtain all pair-wise matching. This fact makes some of those algorithms to have high computational cost; we refer to P-Dim GA and Agglomerative GA or to obtain worst results than methods that compute directly a common labeling.

A summary of the algorithms that will be evaluated is given in Table 3-1. The Graduated Assignment has been the graph matching algorithm, K, used in method 2.2.

The ground truth used in this section is the multiple isomorphism (MI) computed using (Gold and Rangarajan 1996). This MI is obtained by

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

applying $P \cdot (P-1)$ times the graph-matching algorithm. Unquestionably, in most of the cases, the result is a non-consistent MI and so a final CL cannot be obtained. The aim of using this solution as a ground truth is twofold. First, we know the initial Consistency Index ((2.4) in chapter 2), which can be used as an indicator of the effort that the algorithms have to do to get a consistent MI. Second, we can deduce the increase of cost of imposing the MI to be a CMI.

Algorithm	CC	Main features
(Bonev, Escolano et al. 2007)	$O(TP^2N^4)$	• Single pair-wise error could derive to bad global result. • Global knowledge is only considered at the end of the process.
P-Dim GA	$O(TN^{2P})$	• Cost make unfeasible its use when *N > 3 & R>15.*
Agglomerative GA	$O(N^P)$	• Cost make unfeasible its use when *N > 3 & R>15.*
Least squares method	$O(P^3N^6)$	• Rely completely on previous computed pair-wise labelings, which could derive in global bad result • Global knowledge is only considered at the end of the process.
Average Alignment	$O(TP^2N^4)$	• Convergence is not mathematically guaranteed, however all tested examples have converged.

Table 3-1: Comparison of the algorithms evaluated in this section.

The selected methods have been evaluated using three datasets: the Letter, the GREC and the Synthetic. Those datasets are described in section 1. To compare the algorithms, the cost of the returned consistent multiple isomorphisms was considered. See Definition 2-5 of chapter 2. The specific distance function of the cost computation, $C_{ai,bj}^{p,q}$, was evaluated as the graph edit distance cost of assigning edge e_{ab}^{p} to edge $e_{i,j}^{q}$. That is:

$$C_{ai,bj}^{p,q} = {C_V}_{a,i}^{p,q} + {C_V}_{a,i}^{p,q} + {C_E}_{ai,bj}^{p,q} \tag{3.1}$$

and for the case of the P-Dim GA algorithm:

$$C_{ain,bjm}^{p,q,k} = C_{ai,bj}^{p,q} + C_{an,bm}^{p,k} + C_{in,jm}^{q,k} \tag{3.2}$$

like (1.8) of chapter 5 shows. Where,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$\mathcal{C}_{V_{ai}}^{p,q} = \begin{cases} \left\|\gamma_v(v_a^p) - \gamma_v(v_i^q)\right\|_2 & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \\ K_n & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \hat{\Sigma}_v^q \\ K_n & v_a^p \in \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \end{cases}$$

$$\mathcal{C}_{E_{ai,bj}}^{p,q} = \begin{cases} 0 & e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \ \wedge \ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q \\ K_e & e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \ \wedge \ e_{ij}^q \in \hat{\Sigma}_e^q \\ K_e & e_{ab}^p \in \hat{\Sigma}_e^p \ \wedge \ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q \end{cases} \tag{3.3}$$

The specific parameters of K_n and K_e are shown in Table 3-2. These costs will also be used in section 3.2.

We present two experiments. The first addressed to compare the P-Dim GA and Agglomerative GA. Results of the tests are presented in Table 3-3, Table 3-4, Table 3-5, Table 3-6 and Table 3-7. The two first rows of each table show the mean cost (mean values of equation (2.1) of chapter 2) and the mean Consistency Index (mean values of equation (2.4) of chapter 2) for the ground truth algorithm. The rest of rows show the mean cost (mean values of equation (2.7) of chapter 2) obtained using the evaluated algorithms. The Consistency index for the CL algorithms is not shown due to is always 1 since all algorithms obtain a CMI. To present a summarized version of the tests we grouped the results in three different ways. The first, groups the data by number of graphs per test. In this way (referring to Table 3-3, Table 3-4 and Table 3-5), cell [*P-Dim GA, P=3*] of Table 3-3 represents the mean of: 12 random experiments for class 'A', 12 random experiments for class 'E',…, and 12 random experiments for class 'Z'; the cell [*P-Dim GA, N=3*] of Table 3-4 represents the mean of: 12 random experiments for class '1', 12 random experiments for class '2',…, and 12 random experiments for class '22'. An empty cell means that the test could not be performed due to the computational cost of the algorithms. The second way (referring to Table 3-6) presents the data grouped by noise levels instead of by size of Γ. In this way, each cell represents the mean of 5 experiments with different sizes of Γ (P) and different sizes of graph (N) with a concrete noise level. Finally, the third way (Table 3-7) presents the results grouped by number of nodes per graph. The second and third way is only applied to the synthetic tests due to with the other datasets the grouping constant is unknown or variable.

Dataset	K_v	K_e
Letter	1	1
GREC	80	20
Synthetic	80	20

Table 3-2: constant values for the graph edit distance.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Letter	**Elements per test**					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	34.2	111.5	233.4	501.0	1175.8	2119.1
Consistency Index	0.88	0.90	0.91	0.89	0.90	0.89
Common Labeling						
P-Dim GA	**34.6**	-	-	-	-	-
Agglomerative GA	34.8	-	-	-	-	-
Average Align.	35.3	**112.5**	**235.9**	**507.2**	**1187.9**	**2138.8**
(Bonev, Escolano et al. 2007)	35.0	114.6	241.9	537.2	1297.4	2426.4

Table 3-3: results using the Letter Dataset.

GREC	**Elements per test**					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	4120.4	13454.7	28574.0	62322.5	143324.6	260500.4
Consistency Index	0.82	0.83	0.82	0.82	0.83	0.83
Common Labeling						
P-Dim GA	**4180.3**	-	-	-	-	-
Agglomerative GA	4202.4	-	-	-	-	-
Average Align.	4340.0	**14163.3**	**30497.6**	**66745.3**	**154700.5**	**282687.6**
(Bonev, Escolano et al. 2007)	4216.1	14173.9	30699.1	67383.4	157939.8	291606.3

Table 3-4: results using the GREC Dataset.

Synthetic	**Elements per test**					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	22290.6	74148.0	155692.0	333676.8	777485.5	1408808.8
Consistency Index	0.62	0.62	0.63	0.62	0.62	0.62
Common Labeling						
Average Align.	**24092.2**	**80987.7**	**170741.5**	**367224.3**	**857036.0**	**1563486.6**
(Bonev, Escolano et al. 2007)	24339.1	86899.4	190437.3	426117.1	1041782.5	1927068.1

Table 3-5: Results using the synthetic dataset, grouped by Γ size.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Synthetic	**Noise level**				
Ground Truth	$\boldsymbol{\sigma = 0.1}$	$\boldsymbol{\sigma = 0.2}$	$\boldsymbol{\sigma = 0.4}$	$\boldsymbol{\sigma = 0.5}$	$\boldsymbol{\sigma = 06}$
Cost	175013.6	330436.7	546550.2	608170.9	649713.4
Consistency Index	0.92	0.76	0.63	0.48	0.44
Common Labeling					
Average Align.	**202869.3**	**401299.9**	**732324.4**	**826659.0**	**917383.7**
(Bonev, Escolano et al. 2007)	233356.8	386921.0	594172.2	649715.7	688807.8

Table 3-6: Results using the synthetic dataset, grouped by Noise Level.

The second experiment evaluates the least squares method. In this evaluation we compare the Least Squares methodology to the Average Alignment algorithm and the method of Bonev *et al.* presented in (Bonev, Escolano et al. 2007). In this evaluation we also select as K algorithm the Graduated Assigment (Gold and Rangarajan 1996). The evaluation of these three algorithms has been performed using the Letter and the Synthetic dataset (with a single configuration of $N = 10$). With the aim of obtaining non-biased results, the experiments were performed 10 times in both datasets.

Results are presented in Table 3-8, Table 3-9 and Table 3-10. Table 3-8 show the results on the Letter dataset, Table 3-9 and Table 3-10 on the Synthetic dataset.

Analyzing the results obtained with the P-Dim GA and the Agglomerative GA, we see that the P-Dim GA algorithm obtains the best results with $P = 3$ in the Letter and GREC dataset. The performance of the P-Dim GA is nearly followed by the Agglomerative GA Assignment algorithm which obtains the second lower cost with $P = 3$.

Comparing the polynomial time algorithms, the Average alignment algorithm obtains better costs than the other methods in all experiments except for the case of $P = 3$ in the Letter and GREC (Table 3-3 and Table 3-4) datasets and on the Synthetic dataset (Table 3-10).

With respect to the Least Squares method, results improve the method presented in (Bonev, Escolano et al. 2007). However, its performance is kept below the Average alignment algorithm. It is important to notice that the Average alignment algorithm do not only rely on static pair-wise matching but also on the global knowledge of the set. It is important to see that under low noise (such as $\sigma \leq 0.1$), where no random node insertion and deletions are produced, all algorithms obtain the same results, see Table 3-9. This is an important fact since we can conclude that the algorithm's performance is closely related to outliers.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Synthetic	**Number of Nodes**					
Ground Truth	**N=5**	**N=10**	**N=15**	**N=20**	**N=25**	**N=30**
Cost	14661.5	51122.6	100589.1	181956.1	271304.5	386531.1
Consistency Index	0.78	0.69	0.67	0.61	0.61	0.60
Common Labeling						
Average Align.	**15886.0**	**55112.1**	**106980.1**	**196417.7**	**297808.7**	**429228.6**
(Bonev, Escolano et al. 2007)	24643.4	80792.1	150082.4	262657.7	376148.0	528333.2
cont ...						
Ground Truth	**N=40**	**N=50**	**N=60**			
Cost	675889.4	982758.1	14933399			
Consistency Index	0.57	0.57	0.54			
Common Labeling						
Average Align.	**759009.1**	**1096150.2**	**1638759.6**			
(Bonev, Escolano et al. 2007)	888340.0	1322784.7	1911183.7			

Table 3-7: Results using the Synthetic Dataset, grouped by number of nodes per graph.

Letter	**Elements per test**				
Ground Truth	**P=3**	**P=5**	**P=7**	**P=9**	**P=10**
Cost	19.42	66.14	137.59	236.51	295.40
Common Labeling					
Least Squares	19.97	66.58	139.17	240.27	298.75
Average Align.	**19.95**	**65.73**	**138.43**	**237.51**	**296.17**
(Bonev, Escolano et al. 2007)	19.74	67.15	141.04	241.91	306.91

Table 3-8: results using the Letter Dataset.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Synthetic	Noise Level					
Ground Truth	$\sigma = 0.1$	$\sigma = 0.2$	$= 0.2$	$\sigma = 0.5$	$\sigma = 0.6$	$\sigma = 0.7$
Cost	2280.7	7891.7	11912.9	12568.4	13764.2	13972.7
Common Labeling						
Least Squares	2280.7	8694.8	12912.6	13382.5	14886.4	15056.0
Average Align.	**2280.7**	**8298.6**	**12533.2**	**13108.1**	**14612.3**	**14777.8**
(Bonev, Escolano et al. 2007)	2280.7	11066.0	17535.6	18153.1	20629.1	21347.6

Table 3-9: Results using the synthetic dataset, grouped by Noise Level.

Synthetic	Elements per test				
Ground Truth	**P=3**	**P=5**	**P=7**	**P=9**	**P=10**
Cost	1333.0	4518.7	9470.2	16282.1	20388.2
Common Labeling					
Least Squares	**1400.4**	4838.3	10200.4	17608.9	21962.9
Average Align.	1442.7	**4716.4**	**9899.1**	**17118.5**	**21498.9**
(Bonev, Escolano et al. 2007)	1534.0	6081.7	13474.8	23641.5	31111.47

Table 3-10: Results using the synthetic dataset, grouped by Γ size.

The Consistency Index values in Table 3-3, Table 3-4 Table 3-5 and Table 3-7 seem to be stable with different values of N and P; however, in Table 3-6 quickly decreases when noise becomes higher. We could conclude that the effort that the algorithms must do to compute a CMI/CL does not depend neither on the size of Γ nor on the number of nodes per graph but highly depends on the noise of the dataset.

3.2 Evaluation of algorithms that compute directly a common labeling

In this evaluation, we compare the performance of the algorithms that explicitly compute the common labeling. To this aim, we evaluated the Common Labeling Graduated Assignment against the method in (Bonev, Escolano et al. 2007). We do not consider the Average Alignment algorithm because the Common Labeling Graduated Assignment algorithm (using algorithm 2-4 to compute each Q) is equivalent to the Average Alignment algorithm but without any numerical trick (see section 2.2.2).

In each experiment, we compute the pair-wise cost of the resulting common labeling, $\mathcal{C}_{CL}(\Gamma, \psi)$ (equation (2.1) of chapter 2). Moreover, we compute $\mathcal{C}_{MI}(\Gamma, \varphi)$ through the Graduated Assignment (Gold and

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Rangarajan 1996) and also the Consistency Index ((2.4) of chapter 2) of the related multiple isomorphism. We compute $\mathcal{C}_{MI}(\Gamma, \varphi)$ because it is a lower bound of $\mathcal{C}_{CL}(\Gamma, \psi)$ due to inequality (2.9) of chapter 2. Furthermore, the Consistency Index shows how good the lower bound is. With high Consistency, the $\mathcal{C}_{CL}(\Gamma, \psi)$ usually is close to $\mathcal{C}_{MI}(\Gamma, \varphi)$. In some specific cases which are not reflected in the results presented here, we obtain $\mathcal{C}_{MI}(\Gamma, \varphi) > C_{CL}(\Gamma, \psi)$. This is because the algorithms used are sub-optimal and the presented algorithm obtains a better solution than the solution obtained by the Graduated Assignment.

The algorithms are evaluated over three databases: synthetic, Letter and GREC. The synthetic one with parameters $N = \{5, 7, 10, 15, 20\}$, $P = \{3, 5, 7, 10, 15, 20\}$ and $\sigma = \{0.1, 0.2, 0.4, 0.5, 0.6\}$. With the aim of obtaining non-biased results, we performed 7 experiments per configuration and we averaged the results. Therefore, with the synthetic dataset, we performed a number of 7 (rounds) x 6 (N values) x 5 (P values) x 5 (Noise Levels) = 1050 experiments. The second dataset was the Letter dataset. From each class, we randomly selected $P = \{3, 5, 7, 10, 15, 20\}$ graphs, to generate the common labeling. With this dataset, we performed 12 rounds obtaining a number of 15·6·12=1080 experiments. The last dataset used was the GREC. For each class and different $P = \{3, 5, 7, 10, 15, 20\}$ values, we performed 12 rounds obtaining a number of 22·6·12=1584 experiments.

Table 3-11, Table 3-12 and Table 3-13 show the cost and the Consistency Index values obtained using the synthetic dataset.

In Figure 3-1 and associated Table 3-11, values are shown depending on the number of graphs in Γ. That is, the experiments done with different number of nodes and different levels of noise have been averaged. $\mathcal{C}_{CL}$ computed by (Bonev, Escolano et al. 2007) clearly increase with a greater rate than the Common Labeling Graduated Assignment. This last algorithm seems to be very close to the ground truth used. For those experiments the related Consistency Index is kept almost constant.

In Figure 3-2 and Table 3-12, results have been grouped by number of nodes. We see that when the number of nodes increases results show that the proposed algorithm also improves. The Consistency Index seems to be stable.

Finally, in Figure 3-3 and Table 3-13, results have been grouped by the level of noise. In these results, there is a clear increase in the cost and decrease of the Consistency Index. As a conclusion for the experiments using the Synthetic dataset, the presented method obtains lower cost than the method presented in (Bonev, Escolano et al. 2007) and moreover quite close to the $\mathcal{C}_{MI}$. It is worth emphasising that the computational cost of the Graduated Assignment Common Labeling and (Bonev, Escolano et al. 2007) are similar.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

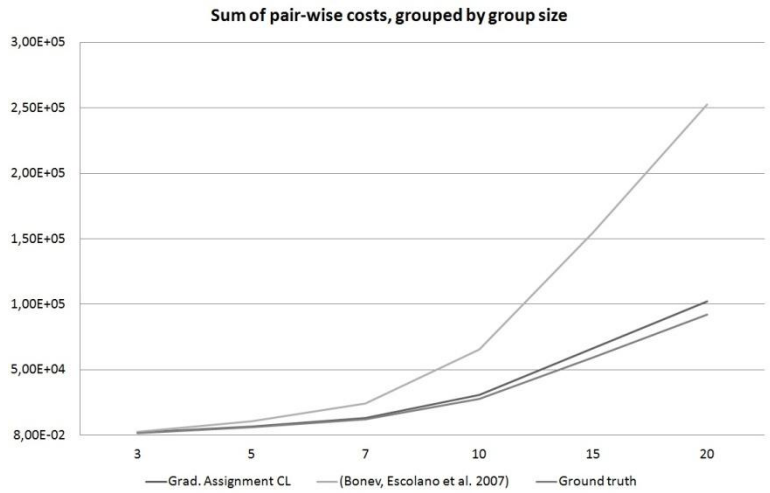


Figure 3-1: results on synthetic dataset grouped by Γ size.

Synthetic	**Elements per test**					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	1860.8	5958.6	12020.6	27773.9	59465.1	92165.0
Consistency Index	0.71	0.74	0.76	0.74	0.74	0.77
Common Labeling						
CL Grad. Ass.	**1949.9**	**6479.1**	**13136.2**	**30824.1**	**66512.2**	**102151.9**
(Bonev, Escolano et al. 2007)	2563.4	10704.9	24438.9	65694.7	155269.3	252599.1

Table 3-11: results on synthetic dataset grouped by Γ size.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

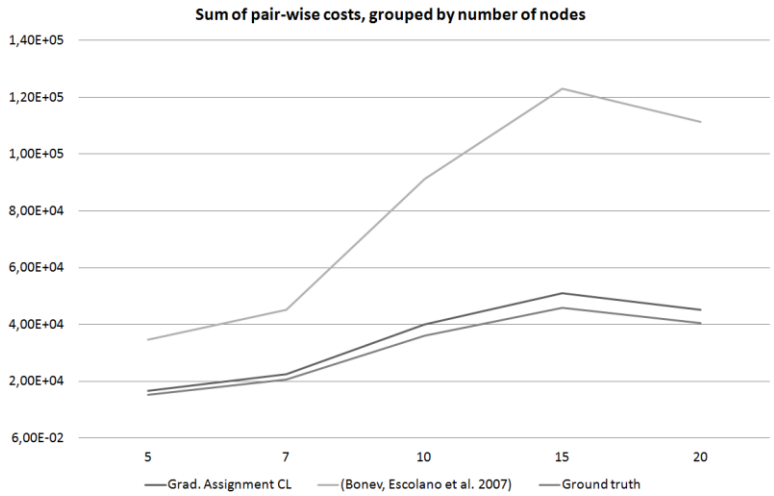


Figure 3-2: results on synthetic dataset grouped by number of nodes per graph.

Synthetic	**Number of nodes in the graphs**				
Ground Truth	**N=5**	**N=7**	**N=10**	**N=15**	**N=20**
Cost	15144.8	20636.7	35958.6	45933.1	40500.2
Consistency Index	0.79	0.80	0.72	0.71	0.65
Common Labeling					
CL Grad. Ass.	**16548.9**	**22542.5**	**40142.0**	**51024.3**	**45283.8**
(Bonev, Escolano et al. 2007)	34744.1	45231.4	91210.9	123037.4	111313.5

Table 3-12: results on synthetic dataset grouped by number of nodes per graph.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

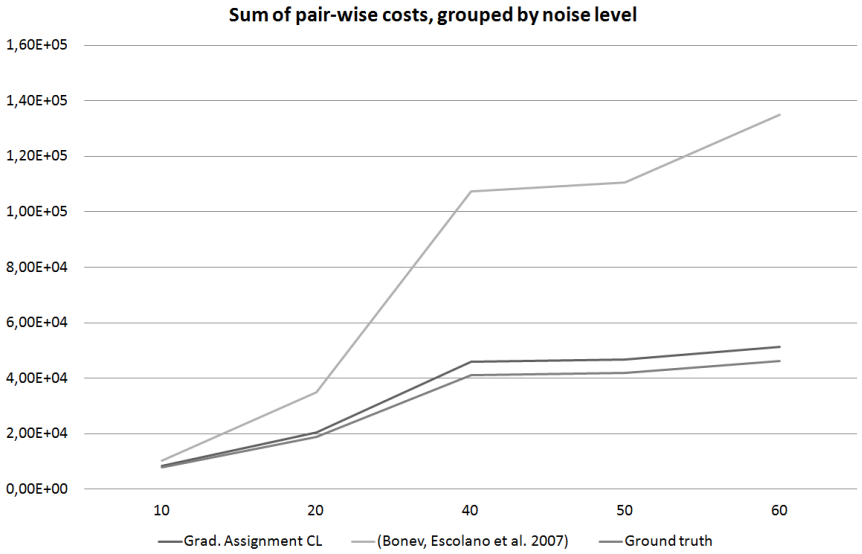


Figure 3-3: results on synthetic dataset grouped by noise level.

Synthetic	**Elements per test**				
Ground Truth	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
Cost	7933.5	18735.0	41090.1	41856.8	46204.1
Consistency Index	0.98	0.90	0.67	0.63	0.53
Common Labeling					
CL Grad. Ass.	**8322.1**	**20560.48**	**45847.6**	**46707.8**	**51424.4**
(Bonev, Escolano et al. 2007)	10214.7	34907.2	107474.1	110535.34	134925.1

Table 3-13: results on synthetic dataset grouped by noise level.

Figure 3-4, Figure 3-5, Table 3-14 and Figure 3-6, Figure 3-7, Table 3-15 show the results of the GREC and Letter datasets grouped by class number and by number of graphs per test. In all the cases, the presented algorithm obtains lower $\mathcal{C}_{CL}$ than the algorithm in (Bonev, Escolano et al. 2007).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

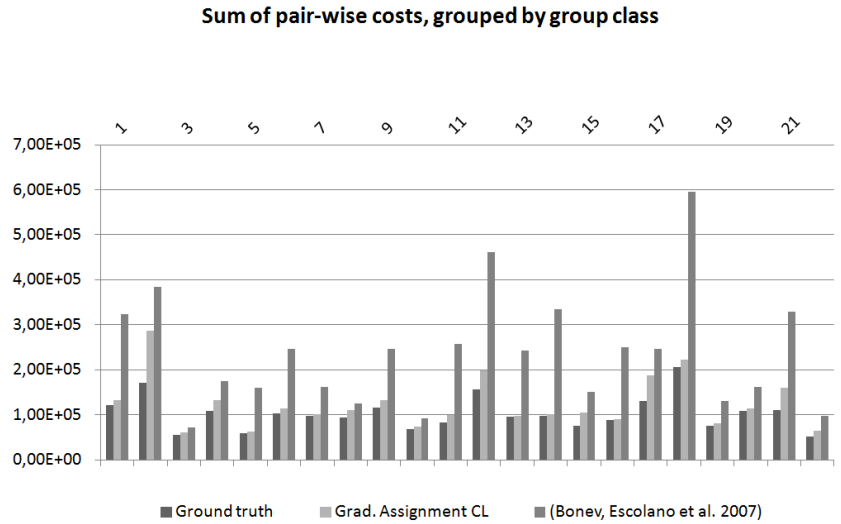


Figure 3-4: results on GREC dataset grouped by class element.

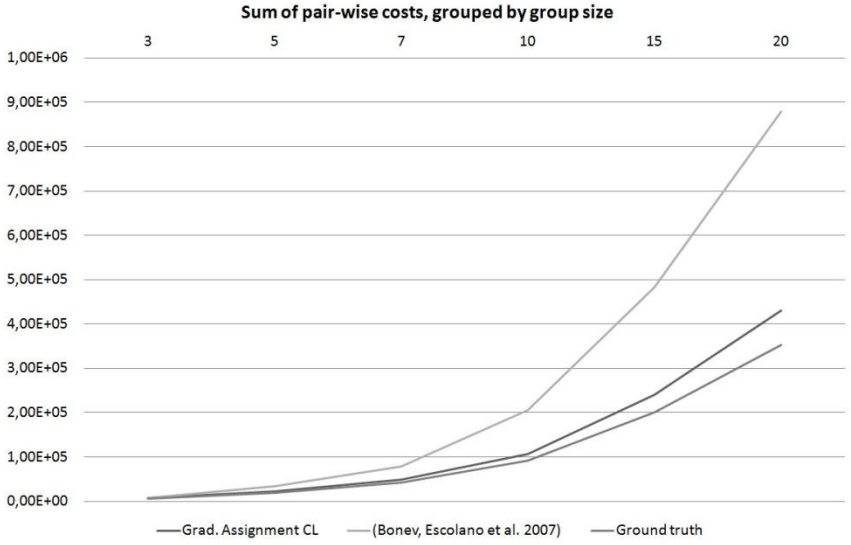


Figure 3-5: results on GREC dataset grouped by Γ size.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

GREC	Elements per test					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	6367.2	19832.0	41766.7	91447.13	199966.6	352319.8
Consistency Index	0.76	0.77	0.77	0.77	0.77	0.78
Common Labeling						
CL Grad. Ass.	**6881.4**	**22546.0**	**48692.6**	**106480.1**	**241120.1**	**430309.5**
(Bonev, Escolano et al. 2007)	7986.9	33134.7	79075.0	206428.7	482576.5	879728.5

Table 3-14: results on GREC dataset grouped by Γ size.

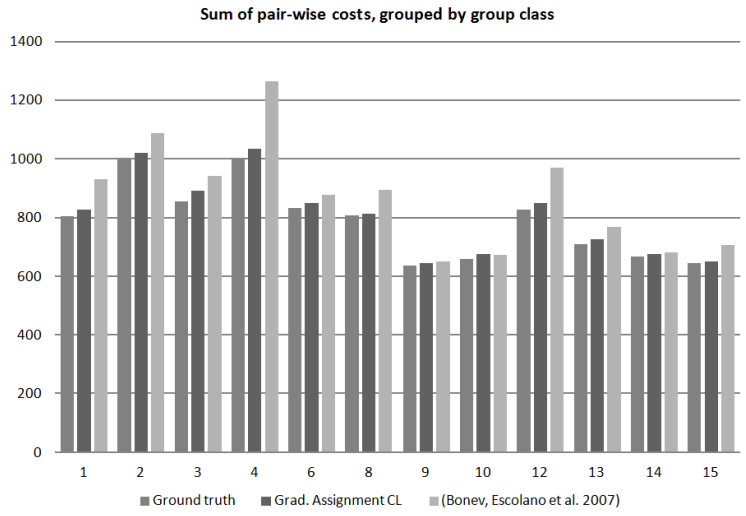


Figure 3-6: results on Letter dataset grouped by class element.

The increase of the difference between (Bonev, Escolano et al. 2007) algorithm and the ground truth is due to the presented costs in section 3.1 are normalized by the number of nodes that contain the final consistent multiple isomorphism of each independent solution. In this way, the method of (Bonev, Escolano et al. 2007), since has tendency to add extra null nodes, its normalization constant tends to be higher and therefore the final cost lower. In the experiments of section 3.2, the normalization constant is the same for all algorithms. This constant is set to the maximum cardinality of any graph in the set Γ.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

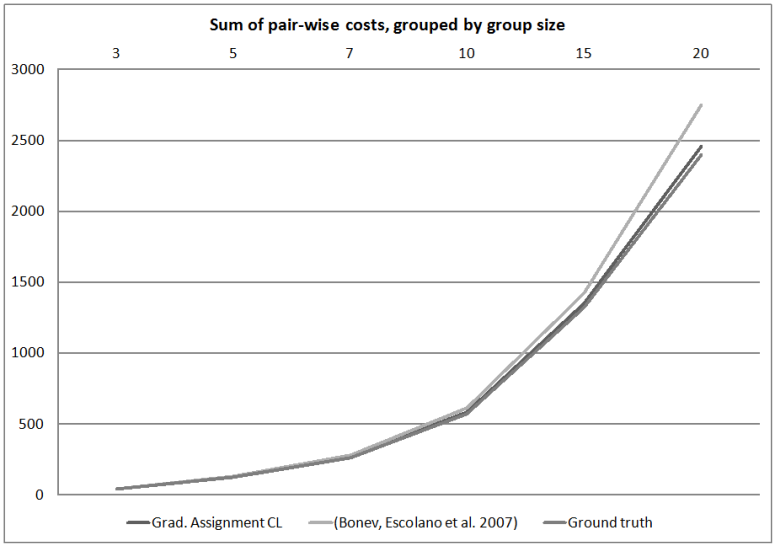


Figure 3-7: results on Letter dataset grouped by Γ size.

Letter	Elements per test					
Ground Truth	**P=3**	**P=5**	**P=7**	**P=10**	**P=15**	**P=20**
Cost	38,71196	124,0034	263,6053	569,478	1318,964	2397,597
Consistency Index	0,85	0,86	0,88	0,86	0,87	0,86
Common Labeling						
CL Grad. Ass.	39,0907	125,7602	268,2897	580,9397	1352,212	2453,062
(Bonev, Escolano et al. 2007)	38,90543	127,6072	276,0038	609,2057	1419,612	2744,525

Table 3-15: results on Letter dataset grouped by Γ size.

3.3 Evaluation of the Common Labeling Dominant Sets

This section presents preliminary evaluation of the Common Labeling Dominant Set algorithm (the dominant set was computed with the same configuration as section 2). We evaluate two different aspects; the first corresponds to the efficiency on minimizing the common labeling cost and the second evaluates the efficiency on computing the Generalized Median Graph. Intuitively, the improvement of the pair-wise distance should be related to the performance of the Median graph that will be constructed. The Median graph is computed as section 3 of chapter 2 shows, with the special characteristics that if a node does not increase the 0.5 probability of appearing it is not included in the median graph.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

As a difference between this and the other tests in the above sections the cost computation between node attributes is computed as:

$$C_{V_{ai}^{p,q}} = \begin{cases} \left\| \gamma_v(v_a^p) - \gamma_v(v_i^q) \right\|_1 & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \\ K_n & v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \hat{\Sigma}_v^q \\ K_n & v_a^p \in \hat{\Sigma}_v^p \ \wedge \ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q \end{cases} \quad (3.4)$$

The evaluation is done with two datasets; one corresponds to the Letter dataset and the second one to the Synthetic dataset (with neither node removal nor insertion).

Table 3-16 and Table 3-18 present the results of the common labeling cost. Each cell of the tables represents the mean value obtained with 12 experiments. In each experiment, *P* graphs were chosen randomly and its common labeling was computed. Table 3-17 and Table 3-19 show the results for the second experiment. Each Generalized Median Graph was constructed using the common labeling obtained in the first experiment, so each cell represents the mean sum of distance to the constructed Generalized Median Graph of 12 experiments. Best values of each experiment are highlighted.

Letter Dataset Common Labeling Cost	**Elements per test**				
Algorithm	**P=3**	**P=5**	**P=7**	**P=9**	**P=11**
Dominant Set Common Labeling	**77.1**	**343.1**	**814.4**	**1544.3**	**2492.1**
Common Labeling Graduated Assignment	94.2	383.9	884.2	1595.0	2506.4
(Bonev, Escolano et al. 2007)	116.7	480.9	1117.9	1930.9	3125.5

Table 3-16: sum of pair-wise distances evaluation on the Letter dataset

Letter Dataset SUM OF DISTANCES TO MEDIAN	**Elements per test**				
Algorithm	**P=3**	**P=5**	**P=7**	**P=9**	**P=11**
Dominant Set Common Labeling	**31.2**	**65.4**	**102.2**	**142.9**	**182.9**
Common Labeling Graduated Assignment	38.1	72.5	111.3	147.3	184.5
(Bonev, Escolano et al. 2007)	45.9	88.9	136.6	172.7	221.0

Table 3-17: sum of distances to median on the Letter dataset.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Synthetic Dataset Common Labeling Cost		**Noise Level**		
Algorithm	**|Γ|**	**σ =30**	**σ = 40**	**σ =50**
Dominant Set Common Labeling	**P=3**	**3048.5**	**3536.7**	**3503.3**
	P=5	16444.1	**15915.4**	15730.9
	P=7	35196.6	39111.7	37784.4
	P=9	64860.0	73279.2	70345.9
	P=11	104717.9	116287.8	111590.4
Common Labeling Graduated Assignment	**P=3**	3340.5	3907.0	3906.5
	P=5	**16369.8**	16336.9	**15588.5**
	P=7	**31965.6**	**36943.0**	**35756.8**
	P=9	**58085.7**	**67472.9**	**65958.0**
	P=11	**93432.9**	**106847.3**	**100534.0**
(Bonev, Escolano et al. 2007)	**P=3**	6180.1	6308.0	6496.1
	P=5	23399.4	23379.3	26514.8
	P=7	59411.8	56101.1	59845.8
	P=9	108333.4	94407.6	100613.3
	P=11	167143.1	143993.8	171575.2

Table 3-18: sum of pair-wise distances evaluation on the synthetic dataset.

Synthetic Dataset SUM OF DISTANCES TO MEDIAN		**Noise Level**		
Algorithm	**|Γ|**	**σ =30**	**σ = 40**	**σ =50**
Dominant Set Common Labeling	**P=3**	**1257.7**	**1417.2**	**1402.4**
	P=5	3181.3	3162.8	3027.9
	P=7	4559.8	5035.6	4798.1
	P=9	6216.5	7074.8	6703.3
	P=11	7962.7	8962.2	8576.0
Common Labeling Graduated Assignment	**P=3**	1350.6	1550.4	1572.9
	P=5	**3171.7**	**3136.2**	**2951.8**
	P=7	**3987.5**	**4518.4**	**4456.0**
	P=9	**5399.2**	**6176.5**	**6137.2**
	P=11	**6869.8**	**7849.6**	**7463.0**
(Bonev, Escolano et al. 2007)	**P=3**	2490.0	2414.9	2519.9
	P=5	4645.0	4555.5	5139.1
	P=7	7797.6	7304.7	7896.2
	P=9	10500.6	9351.4	9762.1
	P=11	12945.0	11480.5	13178.7

Table 3-19: sum of distances to median on the synthetic dataset.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Analyzing the results, we see that the Common Labeling Dominant Set algorithm is able to obtain better results than the Common Labeling Graduated Assignment on the Letter dataset. On the synthetic dataset, the tendency changes as we increase the number of elements per set. Considering the size of the association graph, these results are reasonable because of the complexity of the dominant set problem. Results of the dominant set are susceptible to improve by either testing more random initialization, doing some other less naive initialization or applying some heuristics to reduce size of the association graph.

With respect to the distances to the Generalized Median Graph, we see there is a clear relation with the minimization of the pair-wise distances. The lower the common labeling cost, the better the median graph that will be constructed.

4. EVALUATION OF GRAPH PROTOTYPE CONSTRUCTED USING A COMMON LABELING

4.1 Evaluation of the computation of the generalized median graph with a common labeling

Assuming that computing the common labeling of set of graph we minimize the distance to the Generalized Median Graph, in the following tests we evaluate the Common Labeling Graduated Assignment approach to construct it. In this way, we compare the Common Labeling Graduated Assignment approach with two other standard approaches to compute the common labeling: the hierarchical construction and the genetic construction[11] presented in (Bunke, Münger et al. 1999). To have some reference point on the results we also compare the results with the set median. We consider the set median as a ground truth.

To evaluate the performance of the median graph obtained with the compared methods, we perform similarity queries over a graph database. To do so, as explained in section 4.1.4 of chapter 2, we construct an m-tree where routing nodes are represented using median graphs. This median graph is computed considering the common labeling obtained with the evaluated methods.

The evaluation has been done with three datasets: the Letter, the GREC and the Fingerprint. For each dataset an m-tree was constructed taking 100 random graphs from the corresponding dataset. To evaluate the goodness of

[11] 200 chromosomes of initial population, 50000 generation, crossover set to 0.5 and mutation probability to 0.1.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

the constructed m-trees, 50 range queries considering 7 different ranges where executed. Since each datasets may distribute elements along the m-tree with different distances, the ranges of the queries where normalized among the three datasets. The range of the query was computed as $d_{max} = w \cdot D_{max}$ where D_{max} is the maximum distance between any graph in the m-tree and $w \in \{0.4, \dots, 1.0\}$. For each range, we present the mean of the 50 queries.

We split the results in two parts. The first part shows the overlap of the m-trees, this overlap is computed as section 4.1.5 of chapter 2 shows. The second part shows the performance of the queries. For this second part, we analyze the access ratio (that represents the efficiency) and the F-measure (that represents the effectiveness).

Overlap of the m-trees (Table 4-1). The Graduated Assignment and the Genetic method compute a Common Labeling of the whole sub-cluster in each level of the m-tree. For this reason, when there is a large difference between graphs (GREC) it is hard for both global methods to uniquely label all nodes to a single characteristic (in this case we consider that the set of graph should be represented by a set of prototypes instead of a single one) consequently the radius of the m-tree node d^M increase. For this reason, the overlap is higher than the other two methods. The smaller are the graphs (Fingerprint), the better the methods that compute a common labeling perform.

	Overlap			
	Set Median (Berretti, Bimbo et al. 2001)	Hierarchical M. (Serratosa, Solé-Ribalta et al. 2010)	Generalized median with Common Labeling Graduated Assignment	Genetic Median (Bunke, Münger et al. 1999)
COIL-RAG	0,16	0,02	0,04	0,52
Fingerprint	0,05	0,02	0,00	0,01
GREC	0,02	0,07	0,20	0,20

Table 4-1: Overlap of the m-trees using the four methods and the three databases.

Access ratio and F-measure (Table 4-2). Considering that in the three experiments we have obtained not correlated results, access ratio seems to be very dependent on the dataset. Nevertheless, the F-measure seems to be more constant in the three experiments. The median computed with the Graduated Assignment Common Labeling obtains the highest F-measure. It is important to consider both measures together. When the access ratio is higher than 1, it has no sense to use the m-tree since queries perform more comparisons than if the database had not structure. In general, when the cardinality of the graphs is high, and also their dissimilarity, the methods that use a global common labeling tend to have an access ration higher than 1 due to it is hard for these methods to find a good common labeling

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

between all the graphs. In the cases that the access ratio is almost zero, the m-tree decides if the query is accepted or discarded in the root of the m-tree.

4.2 Evaluation on constructing other graph prototypes

Considering the clear relation between the common labeling problem and the construction of graph prototypes, in this section, we evaluate the suitability of the Common Labeling Graduated Assignment algorithm to compute several graph prototypes. We compare the obtained results with the set median. Equivalently to the evaluation done on section 4.1, result of the common labeling are used to compute the graph prototypes. These resulting prototypes are used to construct a graph metric-tree. In this section, KNN queries are performed. Six prototypes have been used: the set median, the generalized median, the closure graph, first order random graphs, function described graphs and second order described graphs. All those prototypes are described in detail in chapter 2. The Set Median does not use the common labeling algorithm, since it does not need it. The other 5 use the common labeling algorithm and so the labelings between nodes of the involved attributed graphs are exactly the same for the 5 prototypes. This fact is important since we want to see the relation between the common labeling and every particular graph prototype. The evaluation was performed over three datasets: the Letter, the COIL, and the GREC.

For each dataset an m-tree was constructed taking 100 random graphs from the corresponding dataset. To evaluate the goodness of the constructed m-trees 50 KNN queries considering $K = \{3,5,7\}$ were performed. For each K, we present the mean of the 50 queries. Results are shown in Table 4-3.

With regard to the Generalized Median Graph performance in COIL and Letter datasets, we see that the Set Median, without the need of computing the common labeling, obtains better results than the Generalized Median. We assume that this fact is due to the graphs and classes involved are so different that the Generalized Median Graph is forced to over-generalize the cluster that represents and consequently tree nodes have large radius. This assumption is based on that with the same common labeling other graph prototypes are able to achieve better performance than the set median.

Closure Graphs are very dependent on the data and on the discretization process. For this reason, it is usual to have very different performance with different datasets in both measures: F-Measure and Access Ratio.

Finally, results on the three probabilistic prototypes (FORG, FDG and SORG) show that the common labeling succeed on providing good enough initial labeling to obtain a good representation of the cluster.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

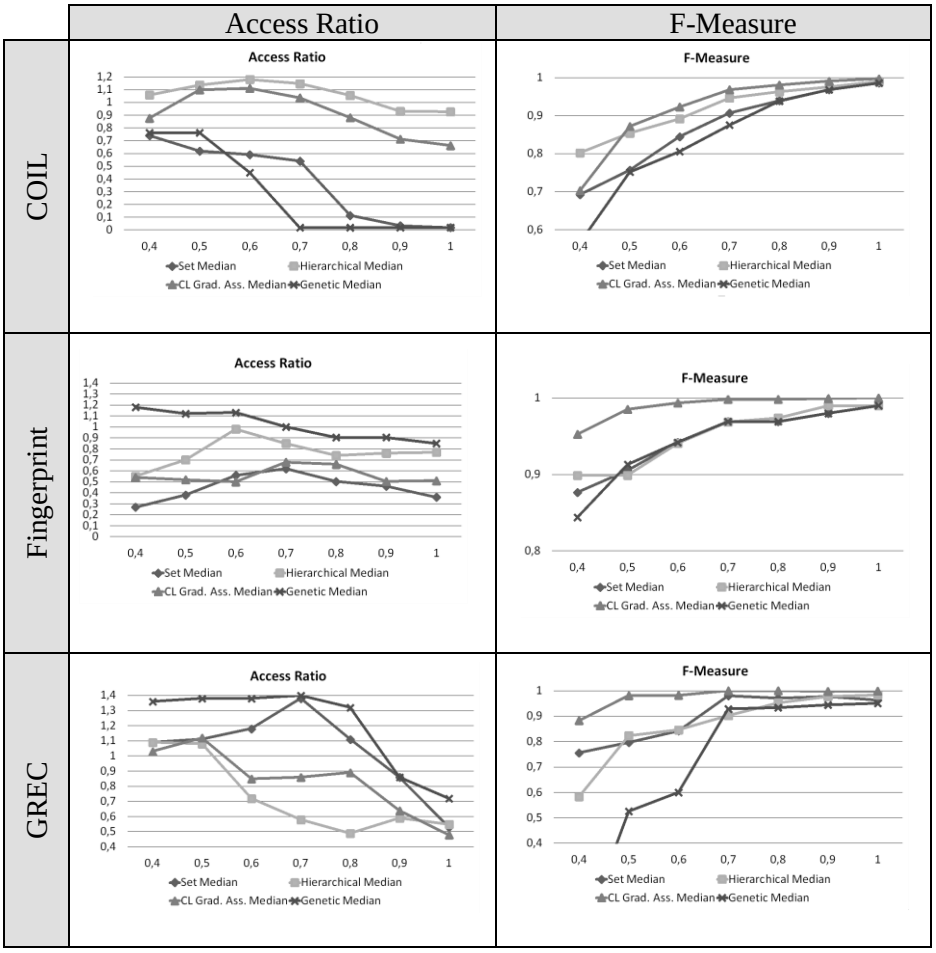


Table 4-2: Access Ratio and F-measure obtained using the three databases. The horizontal axis represents the evaluation of d_{max}.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

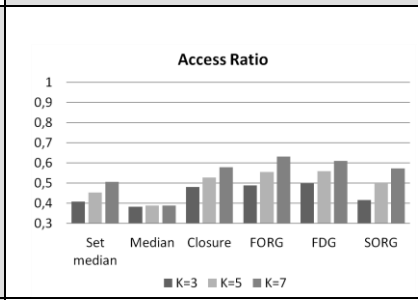
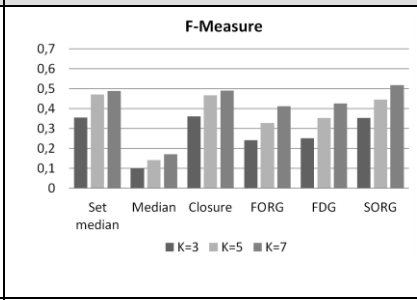
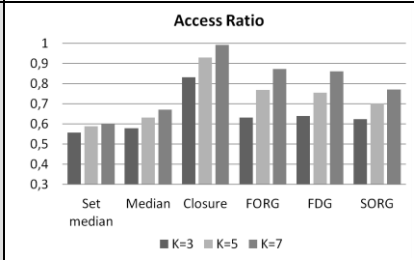
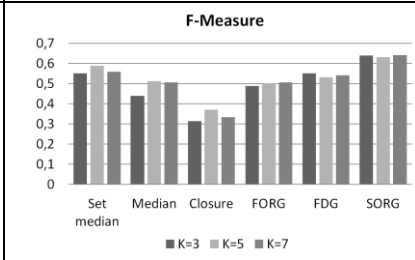
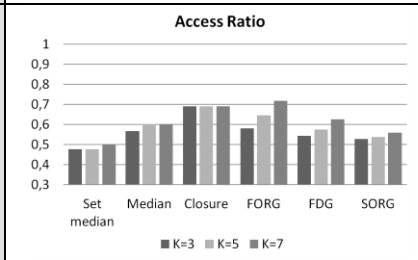
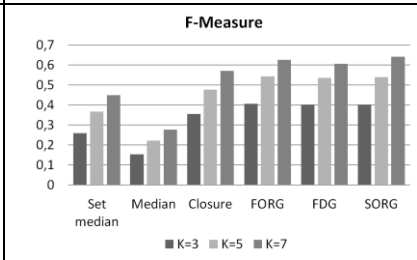
	Access Ratio	F-Measure
Letter		
COIL		
GREC		

Table 4-3: F-Measure and Access Ratio obtained using the three datasets.

5. ALIGMENT OF SEQUENTIAL IMAGES WITH THE COMMON LABELING FRAMEWORK

Determining sparse correspondences between sets of features is a recurrent problem in computer vision. It arises at the early stages of many computer vision applications such as 3D scene reconstruction, object recognition, pose recovery and image retrieval, among others. The use of local image contents may not suffice to get a reliable correspondence between points of two images under certain circumstances e.g. large rigid/non-rigid deformations. This is the case of the model fitting paradigm RANSAC (Fischler and Bolles 1981) which is extensively used in computer

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

vision to reject outliers or the Iterative Closest Point (ICP) method (ZHANG 1992) that attempt to simultaneously solve the correspondence and the alignment problem. The mentioned approaches suffer from two major drawbacks. On the one hand, most of these optimization strategies rely on reasonable initial guesses in order to find the global optimum. On the other hand, if there is too much deformation between both images, their underlying geometrical models may fail to accommodate the transformation relating them, even under a reasonable initial guess.

To solve the aforementioned drawbacks and inspired by the article (Williams, Wilson et al. 1997) it seems clear that consideration of global knowledge (when possible) instead of local pair-wise matchings should increase performance on the computation of the individual pair-wise matchings. Hence, we face the correspondence problem in a group-wise manner. In this way, the flow of information among the pair-wise relations of the group has several advantages. It helps to constrain the search of the method towards a globally convenient direction. This contributes to avoid poor local optima. In addition, it alleviates the limitations inherent to the geometrical models.

In this section, we do not work specifically with graphs but with sets of points. However, since the alignment method looks for relational information between local points and its neighbors, a sort of alignment between relational structures is performed. To solve this problem, the Common Labeling Graduated Assignment, proposed in section 2.3 of chapter 5, is adapted to consider affine transformations among the element of the set.

Related to the field of group-wise point registration when data is a sparse set of points we highlight the following work. In (Fergus, Perona et al. 2007) a method to learn objects and detect parts of objects is presented. The model is learned taking images that represent the selected object from the same point of view and without background. The method does not explicitly address the problem presented here since its aim is to construct a model for object recognition. Another related work is presented in (Wang, Vemuri et al. 2008), which performs alignment of sparse data points taking into account that points contain non-rigid deformation. The most similar method to the one presented here could be (Cootes, Twining et al. 2010). It is based on group-wise point set correspondence but it has no consideration about outlier detection, which makes its applicability not feasible with the particular problem it is presented here. This last work was evaluated using two hand-made labeled data sets.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

5.1 Particularization of the definitions of multiple isomorphism and common labeling of a set of points

We start by particularizing the matching problem between sets of points instead of between graphs. In this way, several definitions given in chapter 2 are overwritten. Let $S^p = \{\vec{v}_1^p, \vec{v}_2^p, .. \vec{v}_{N^p}^p\}$ be a set of points with N^p elements. In our method, these types of sets represent images and their elements are salient points extracted from them. Equivalently to the original definition, we represent the set of images by the set $\Gamma = \{S^1, S^2, \dots, S^P\}$. Each S^q in Γ is the characterisation of an image. Following this notation, the correspondences between salient points of a set of images are characterized by the labelings between the elements of the sets S^q in Γ. Note that outlier points in images are also represented as elements in S^q. These outlier points in the images do not correspond to other points on the other images and so the corresponding elements in the sets have not to be labeled from or to these elements.

Definition 5-1 (labeling between two sets of points). Given two sets of points $S^p = \{\vec{v}_1^p, \vec{v}_2^p, .., \vec{v}_{N^p}^p\}$ and $S^q = \{\vec{v}_1^q, \vec{v}_2^q, .., \vec{v}_{N^q}^q\}$ with N^p and N^q elements, a labeling $f^{p,q}$ between these sets assign elements of the first set to elements of the second set $f^{p,q}(\vec{v}_a^p) = \vec{v}_i^q$. Equivalently to the graph matching framework described in chapter 2, we represent this labeling as a permutation matrix:

$$F^{p,q}[a,i] = \begin{cases} 1 & if\ f(\vec{v}_a^p) = \vec{v}_i^q \\ 0 & otherwise \end{cases} \quad (5.1)$$

Note that some points may remain unlabeled, since the cardinalities of the sets are different.

Definition 5-2 (multiple labeling between sets of points). Let $\Gamma = \{S^1, S^2, \dots, S^P\}$ be a set of P sets of points, each with a particular number of elements N^p, $p \in \{1..P\}$. The set φ is an error tolerant multiple isomorphism of Γ if it contains one and only one labeling between any set of points, $\varphi = \{f^{1,2}, \dots, f^{2,1}, \dots, f^{N-1,N}\}$.

The definition of the common labeling can be extended in the same fashion.

5.2 Relation of the common labeling with support functions

Equivalently to the cost of the common labeling on graphs, we define the cost of matching a set of points as the sum of individual pair-wise matchings. That is,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$$C_{CL}(\Gamma,\Psi) = -\sum_{p=1}^{P}\sum_{\substack{q=1\\q\neq p}}^{P}\sum_{a=1}^{N^p}\sum_{i=1}^{N^q}\left(\underbrace{\sum_{w_1=1}^{O^L} H^p[a,w_1]\cdot H^q[i,w_1]}_{\equiv P_f^{p,q}[a,i]}\right)\cdot \left(\underbrace{\sum_{b=1}^{O^p}\sum_{j=1}^{O^q}\left[\underbrace{\sum_{w_2=1}^{O^L} H^p[b,w_2]\cdot H^q[j,w_2]}_{\equiv P_f^{p,q}[b,j]}\right]\cdot C_{ai,bj}^{p,q}}_{R_{ai}^{pq}}\right) \quad (5.2)$$

where $\Psi = \{H^1, H^2, \dots H^P\}$ indicates the relaxed version of the common labeling bijections as shown in chapter 2.
We can easily see in (5.2) the influence of matchings $\vec{v}_b^p \to \vec{v}_j^q$ over $\vec{v}_a^p \to \vec{v}_i^q$. This influence is identified as R_{ai}^{pq} in (5.2) and will be described in detail in the next section.

5.3 Pair-Wise Compatibility Coefficients

Given two sets of points $S^p = \{\vec{v}_1^p, \vec{v}_2^p, .., \vec{v}_{N^p}^p\}$ and $S^q = \{\vec{v}_1^q, \vec{v}_2^q, .., \vec{v}_{N^q}^q\}$, where $\vec{v}_k^p = \left(v_{k_H}^p, v_{k_V}^p\right)^T$ and $\vec{v}_l^q = \left(v_{l_H}^q, v_{l_V}^q\right)^T$ contain column vectors with the two-dimensional coordinates (horizontal and vertical) of each point, in this section we will describe the details of the computation of the compatibility coefficients $C_{ai,bj}^{p,q}$ appearing in equation (5.2).

This quantity R_{ai}^{pq}, also known as the support function, is addressed to measure the support for the match $\vec{v}_a^p \to \vec{v}_i^q$ received from the rest of the matches $\vec{v}_b^p \to \vec{v}_j^q$. This is a common strategy followed in the probabilistic relaxation approaches (Rosenfeld, Hummel et al. 1976; Hummel and Zucker 1983).

The main idea underpinning our computation of the support function is that two points $\vec{v}_a^p$ and $\vec{v}_i^q$ from two different images p and q are in correspondence as long as they show similar spatial distributions in comparison to the rest of the points around them.

Geometric evidence is widely used to solve the correspondence problem. In order to be robust to arbitrary initial poses of the point-sets under a certain geometric assumption, we need to include the estimation of the alignment

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

parameters into the problem. Thus, we redefine the support function in the following way:

$$R_{ai}^{pq} = \max_{\Phi_{ai}} \sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \cdot C_{ai,bj}^{p,q}(\Phi_{ai}^{p,q}) \tag{5.3}$$

where $F^{p,q}[b,j]$ corresponds to the globally propagated probability to match nodes b, j of graphs p, q and $C_{ai,bj}^{p,q}(\Phi_{ai}^{p,q})$ is the compatibility of the simultaneous matches $\vec{v}_a^p \to \vec{v}_i^q$ and $\vec{v}_b^p \to \vec{v}_j^q$ given the affine parameters $\Phi_{ai}^{p,q}$.

Without lose of generality, transformation parameters, in this new formulation, attain robustness to affine pose of the point-sets by selecting the pose configuration that leads to the maximum support.

With respect to classical point-set registration methods, our approach has the particularities that it is targeted to multiple point-set registration and that alignment parameters are local to each correspondence hypothesis $v_a^p \longrightarrow v_i^q$ instead of being a property global to all the points in the set.

Since we compare relational geometric measurements, we define the new coordinate vectors $\vec{x}_{ab}^p = \left(\vec{v}_b^p - \vec{v}_a^p\right)$ and $\vec{y}_{ij}^q = \left(\vec{v}_j^q - \vec{v}_i^q\right)$, that represent the coordinates of the points $\vec{v}_b^p$ and $\vec{v}_j^q$ relative to $\vec{v}_a^p$ and $\vec{v}_i^q$, respectively.

We define the compatibility between two relational geometric measurements $\vec{x}_{ab}^p$ and $\vec{y}_{ij}^q$ under the action of the affine parameters $\Phi_{ai}^{p,q}$ as:

$$C_{ai,bj}^{p,q}\left(\Phi_{ai}^{p,q}\right) = \rho - \left\| \vec{x}_{ab}^{\mathrm{p}} - \Phi_{ai}\vec{y}_{ij}^q \right\|_{\Sigma}^2 \tag{5.4}$$

where $\Phi_{ai}^{p,q}$ is a $2x2$ non-singular matrix representing affine transformation parameters (note that $\vec{x}_{ab}^p$ and $\vec{y}_{ij}^q$ are already invariant to translation), $\|\cdot\|_{\Sigma}^2$ is the squared Mahalanobis distance with covariance matrix Σ, and ρ is a thresholding quantity that controls the outlier process. The estimation of ρ parameter will be detailed in the next section.

According to the proposed measure, the more dissimilar are the relations, the lower is their compatibility. The scale of this comparison is effectively controlled by matrix $\Sigma = \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_H^2 \end{bmatrix}$, a diagonal matrix of variances which may be empirically estimated from the data. With these ingredients, the optimal transformation parameters Φ_{ai}^{p,q^*} that maximize equation (5.3) are:

$$\Phi_{ai}^{p,q^*} = \arg\min_{\widehat{\Phi}_{ai}^{p,q}} \sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j]\left(\vec{x}_{ab}^p - \widehat{\Phi}_{ai}^{p,q}\vec{y}_{ij}^q\right)^T \Sigma^{-1}\left(\vec{x}_{ab}^p - \widehat{\Phi}_{ai}^{p,q}\vec{y}_{ij}^q\right) \tag{5.5}$$

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

where $\Phi_{ai}^{p,q*} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$. We have discarded the constant quantities not depending on the alignment parameters in the substitution of (5.4) to (5.3). Note that we have turned the maximization into a minimization by reversing the sign. Consider the following residuals from the alignment of points $\vec{x}_{ab}^{p}$ and $\vec{y}_{ij}^{q}$:

$$\begin{aligned} r_{bj}^{p,qV} &= x_{ab}^{p\ V} - \left(\phi_{11} y_{ij}^{qV} + \phi_{12} y_{ij}^{qH}\right) \\ r_{bj}^{p,qH} &= x_{ab}^{p\ H} - \left(\phi_{21} y_{ij}^{qV} + \phi_{22} y_{ij}^{qH}\right) \end{aligned} \tag{5.6}$$

Then, the objective function of equation (5.5) is equivalent to the following expression:

$$\mathcal{F} = \sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \left[\left(\frac{r_{bj}^{p,qV}}{\sigma_V} \right)^2 + \left(\frac{r_{bj}^{p,qH}}{\sigma_H} \right)^2 \right] \tag{5.7}$$

Taking derivatives of $\mathcal{F}$ with respect to $\Phi_{ai}^{p,q}$ we obtain the following expressions:

$$\begin{aligned} \frac{\delta \mathcal{F}}{\delta \phi_{11}} &= -\sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \frac{y_{ij}^{qV} 2 r_{bj}^{p,qV}}{\sigma_V^2} \\ \frac{\delta \mathcal{F}}{\delta \phi_{12}} &= -\sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \frac{y_{ij}^{qH} 2 r_{bj}^{p,qV}}{\sigma_V^2} \\ \frac{\delta \mathcal{F}}{\delta \phi_{21}} &= -\sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \frac{y_{ij}^{qV} 2 r_{bj}^{p,qH}}{\sigma_H^2} \\ \frac{\delta \mathcal{F}}{\delta \phi_{22}} &= -\sum_{b=1}^{N^p} \sum_{j=1}^{N^q} F^{p,q}[b,j] \frac{y_{ij}^{qH} 2 r_{bj}^{p,qH}}{\sigma_H^2} \end{aligned} \tag{5.8}$$

The optimal transformation parameters $\Phi_{ai}^{p,q*}$ are found by solving the set of equations:

$$\frac{\delta \mathcal{F}}{\delta \phi_{11}} = 0, \ldots, \frac{\delta \mathcal{F}}{\delta_{22}} = 0, \tag{5.9}$$

with respect to the alignment parameters. This linear system can be expressed in matrix form $M\vec{a} = \vec{b}$, where M is a 4×4 matrix and $\vec{a} = (\phi_{11}, \phi_{12}, \phi_{21}, \phi_{22})^T$ and $\vec{b}$ are 4-column-vectors. This can be solved by matrix inversion (i.e., $\vec{a} = M^{-1}\vec{b}$). In some special cases where for all

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

$F^{p,q}[b,j] > 0$, all y_{ij}^{qV} and y_{ij}^{qH} are linearly dependent with the same linear relation, e.g. $y_{ij}^{qV} = a y_{ij}^{qH}$, $\forall j$, matrix M may not be full rank and so not invertible. These cases are rare to happen in real images. In fact, we expect that the feature selector algorithm implements some mechanism to detect and rectify this type of data.

5.4 Outlier Detection, setting a value of ρ

According to our purposes, a point $\vec{v}_a^p \in S^p$ (or $\vec{v}_i^q \in S^q$) is considered an outlier as far as there is no point $\vec{v}_i^q, \forall\, i \in 1..|S^q|$ (or $\vec{v}_a^p, \forall\, a \in 1..|S^p|$) which presents a support R_{ai}^{pq} (5.3) above a given threshold.

Substituting the compatibilities of equation (5.5) into equation (5.4), the final expression for the supports becomes:

$$R_{ai}^{pq} = \sum_{b=1}^{O^p} \sum_{j=1}^{O^q} F^{p,q}[b,j] \left(\rho - \left\| \vec{x}_{ab}^{\,p} - \Phi_{ai}^{p,q^*} \vec{y}_{ij}^{\,q} \right\|_{\Sigma}^2 \right) \tag{5.10}$$

where Φ_{ai}^{p,q^*} are the optimal transformation parameters computed using equation (5.5).

The parameter ρ plays the role of the *robustness parameter* used by (Rangarajan, Chui et al. 1997; Gold, Rangarajan et al. 1998). It controls whether the geometrical compatibility term contributes either positively (i.e., $\rho < \left\| \vec{x}_{ab}^{\,p} - \Phi_{ai}^{p,q^*} \vec{y}_{ij}^{\,q} \right\|_{\Sigma}^2$) or negatively to the support measure.

We model the outlier detection process as an assignment to (or from) a special point. This is similar to *null* vertex assignments in (Wong and You 1985). The concept of *null* vertices is explained in detail in chapter 2. We consider as outliers all the assignments $v_a^p \longrightarrow v_i^q$ such that $R_{ai}^{p,q} < 0$.

The threshold ρ represents the quantity from which the compatibility starts to contribute negatively. Therefore, it seems reasonable to express ρ in terms of a squared Mahalanobis distance, i.e. $\rho = \left\| \vec{d} \right\|_{\Sigma}^2$. If we express the threshold distance vector proportionally to the standard deviations of the data, i.e. $\vec{d} = (K\sigma_H, K\sigma_v)$, the expression of ρ becomes:

$$\rho = \vec{d}^{\,T} \Sigma^{-1} \vec{d} = \left(\frac{K\sigma_H}{\sigma_H} \right)^2 + \left(\frac{K\sigma_H}{\sigma_H} \right)^2 = 2K^2 \tag{5.11}$$

considering that Σ matrix is diagonal. Rangarajan *et al. (Rangarajan, Chui et al. 1997; Gold, Rangarajan et al. 1998)* do not address the estimation of this parameter in their paper. On the contrary, we define ρ as a function of the number K of standard deviations permitted in the registration errors in order to consider a relation plausible.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

5.5 The algorithm

Considering the objective function in (5.2) for multiple point set matching, we focus on substituting to it the support function deduced in Section 5.3.

The problem becomes then one of joint estimation of correspondence and alignment parameters in which the recovery of the correspondences is influenced by the pose of the point-sets and vice-versa. Most point-set registration methods consist of an iterative process that alternates alignment and correspondence updates. Several approaches exist in order to solve this chicken-egg problem. For example, the well-known ICP (ZHANG 1992), Robust Point Matching (RPM) (Rangarajan, Chui et al. 1997; Gold, Rangarajan et al. 1998) or the Expectation-Maximization Algorithm (Jian and Vemuri 2005; Myronenko and Song 2010; Horaud, Forbes et al. 2011; Jian and Vemuri 2011).

To optimize our objective function, we propose to use a similar dual step solution based on first maximizing the point-to-point alignment to later maximize the correspondences. We base our method on the Common Labeling Graduated Assignment proposed in section 2.3 of chapter 5. In this way, our proposed maximization procedure has the following steps: start with a valid $(\Psi)^t$ at time t, maximize alignment with respect to the rest of points (5.5), compute cost matrix using costs in (5.3), apply Graduated Assignment to compute next $(\Psi)^{t+1}$ and start again until convergence is reached. An outline of the procedure is given below.

141 ***Algorithm*** *MSP-Alignment* (Γ)
142 $H^i = initializeCL();$
143 $\beta = \beta_0$
144 ***repeat until*** $\beta = \beta_0$
145 ***repeat until*** H *converges* ***or*** $t > I_0)$
146 $Q^p_{a,w_1} = ComputeQ(H, p, a, w_1, \Gamma), \forall\ 2 \le p \le P, 1 \le a, w_1 \le N$
147 $H^p[a, w_1] = e^{\beta \cdot Q^p_{a,w_1}}, 2 \le p \le P, 1 \le a, w_1 \le N$
148 ***repeat until*** H *converges* ***or*** $t_2 > I_1$
149 $H^p[a, w_1] = {H^p[a, w_1]} / {\sum_{x=1}^{N} H^p[x, w_1]}, 2 \le p \le P, 1 \le a, w_1 \le N$
150 $H^p[a, w_1] = {H^p[a, w_1]} / {\sum_{x=1}^{N} H^p[a, x]}, 2 \le p \le P, 1 \le a, w_1 \le N$
151 ***end***
152 ***end***
153 $\beta = \beta \cdot \beta_r$
154 *end*
155 *Returns* Ψ

Algorithm 5-1**:** MSP-Alignment algorithm

where β_0, β_r, β_f I_0 and I_1 correspond to the parameters of (Gold and Rangarajan 1996) and are application dependant. We used the values

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

proposed in chapter 2. Function $ComputeQ$ optimizes the alignments and the point-to-point assignations, an outline of the procedure is given below:

Function $ComputeQ$ **input** P_h, p, a, w_1, Γ
returns Q^p_{a,w_1}
  For $\forall q \in 1..P, p \neq q$
    $F^{p,q} = H^p \cdot (H^q)^T$
    For $i = 1..N^q$
      For $b = 1..N^q, b \neq a$
        For $j = 1..N^q, i \neq j$
          $Q^p_{a,w_1} = Q^p_{a,w_1} + H^q[i,w_1] \cdot F^{p,q}[b.j] \cdot \left(\rho - \left\|\vec{x}_{ai} - \Phi^*_{ai}\vec{y}_{ij}\right\|^2_\Sigma\right)$
        End
      End
    End
  End
End Function

Algorithm 5-2: auxiliary function of Algorithm 5-1.

To compute the Q^p_{a,w_1} value one could also use an approximated version similar to the one given in Algorithm 2-4 in chapter 2.
Taking into account our definition of outlier detection, we require to adapt the Sinkhorn normalization (Sinkhorn 1964) to consider them. Recall first that the resulting Q^p_{a,w_1} values could be negative. However, after the exponentiation all values become strictly positive and therefore we can assume the Sinkhorn normalization can be applied. In the normalization over matrix H, we keep in mind that outliers are special assignation that only satisfy one-way constraints, in this way we can easily consider several points as outliers. To this aim, we enhance each matrix H with an extra row and column, following a similar procedure than the slacks in (Gold and Rangarajan 1996). We initialize these extra row and column with the value of 1. We aim to detect outliers, that is points which have $R^{p,q}_{ai} < 0 \; \forall \; i$ or a. We know that $e^x \geq 1$ if $x \geq 0$, thus it is expected that points which have all possible assignations negative are assigned to this special row or column. Finally, when the Sinkhorn method has finished the extra row and column are removed leading to the resulting matrices of global assignments H.

Note that now H cannot be theoretically considered a probability assignation matrix, due to $\sum_{a=1}^{N^p} H^p[a,i] \neq 1$, neither for rows nor for columns. However, we still can ensure that $\sum_{a=1}^{N^p} H^p[a,i] < 1$ and that each individual value is positive. So, what it was a probability matrix H, now it can be assumed to be a fuzzy assignation matrix.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

5.6 Evaluation of the common labeling registration algorithm

To evaluate the effectiveness of the presented method a series of group-wise image registration experiments are done. We use real images from the database in (Mikolajczyk, Tuytelaars et al. 2011). Feature points from each image have been extracted using the Harris operator (Harris and Stephens 1988). We use the following datasets: New York, Van Gogh and Asterix. This datasets are explained in detail in section 1. Each test is performed on a group of P images. The following four methods are compared. (1) Pairwise ICP+RANSAC, which applies the well-known ensemble ICP+RANSAC between each pair of images. (2) Confident ICP+RANSAC, which computes the labelings between the most similar pairs and infers the rest by composition (this method exploits the prior knowledge about the underlying order of the images). A very similar strategy is used in (Williams, Wilson et al. 1997). (3) Pair-wise common labeling, which applies the proposed approach independently to each pair of images (note that, given two images S^p and S^q this approach considers two bijections $f^{1,2}$ and $f^{2,1}$) and (4) Group-wise Labeling, which applies the proposed approach jointly to all the images of the group. This method is the prime motivation of the work. The aim of the comparison is to elucidate the benefits of the group-wise approach vs. the pair-wise one. All the methods have been initialized with the results of the matching by correlation. Regardless the labelings are computed in either pair-wise or a group-wise fashion, results are evaluated in a pair-wise basis. The DLT algorithm (Kovesi 2009) was used to compute the homography corresponding to a given labeling between two images. Since ground truth homographies are available, the accuracy was measured through the mean projection error (MPE) in pixels.

Table 5-1, Table 5-2 and Table 5-3 show the results of the New York, Van Gogh and Asterix datasets using groups of $P = 4$ images. From top to bottom, each cell contains the MPE of Pair-wise ICP+RANSAC, Confident ICP+RANSAC, Pair-wise Labeling and Group-wise Labeling. Images are arranged in rows and columns of the tables according to their logical order. The diagonal cells are empty since they correspond to self-labelings.

Analyzing the results, we see that the common labeling approach obtains usually the lowest mean projection error.

This fact is clear with distant images; see for instance row $img[X]$ and $img[X+3]$ where in all datasets the common labeling error is much lower with respect to all other methods. In some cases, with adjacent images the pair-wise labeling method obtains better labelings, e.g. row $img[X+3]$ and column $img[X+2]$ of Table 4-3 and Table 5-1. However, the difference

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

between this method and the common labeling method is low, recall that the mean projection error is in pixels.

	img[X]	img[X+1]	img[X+2]	img[X+3]
img[X]	-	24,7	127,24	199,26
	-	24,7	84,58	408,774
	-	**4,37**	131,84	161,02
	-	14,13	**16,34**	**20,88**
img[X+1]	82,21	-	44,91	108,57
	19,92	-	26,75	73,03
	4,57	-	4,37	131,83
	8,71	-	**3,45**	**7,77**
img[X+2]	170,74	23,78	-	27,4
	56,95	22,96	-	23,08
	118,27	4,56	-	**4,34**
	10,93	**3,37**	-	4,74
img[X+3]	245,95	83,3	25,4	-
	119,92	57,83	22,45	-
	174,6	118,25	**4,53**	-
	16,84	**8,55**	5,66	-

Table 5-1: results using New York. We have used 25 groups of $P = 4$ images (i.e., results are averaged over 25 experiments).

	img[X]	img[X+1]	img[X+2]	img[X+3]
img[X]	-	23,33	121,28	105,31
	-	23,33	91,62	318,97
	-	**0,85**	14,92	65,46
	-	1,44	**1,12**	**3,6**
img[X+1]	20,51	-	14,4	151,2
	13,33	-	15,39	281,13
	0,7	-	**0,66**	20,2
	1,27	-	0,69	**2,45**
img[X+2]	41,47	15,3	-	22,02
	28,86	14,15	-	17,77
	22,4	**0,52**	-	**0,52**
	0,68	0,54	-	1,61
img[X+3]	56,55	33,34	12,38	-
	132,53	49,48	11,72	-
	45,27	27,67	**0,42**	-
	1,72	**1,6**	1,17	-

Table 5-2: results using Van Gogh. We have used 14 groups of $P = 4$ images (i.e., results are averaged over 14 experiments).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

	img[X]	img[X+1]	img[X+2]	img[X+3]
img[X]	-	134	71,1	104,8
	-	125,28	710,61	53,57
	-	**1,28**	9,88	34,65
	-	**1,28**	**1,42**	**3,44**
img[X+1]	23,2	-	53,9	70,7
	19,16	-	31,33	115,75
	1,05	-	**1,16**	14,01
	1,07	-	1,21	**2,47**
img[X+2]	58,2	36,7	-	37
	74,58	52,65	-	36,51
	14,59	**1,02**	-	3,5
	0,63	1,06	-	**3,01**
img[X+3]	1135,2	145,1	45,7	-
	50,36	74,84	129,69	-
	29,04	11,4	2,63	-
	1,78	**1,47**	**2,44**	-

Table 5-3: results using Asterix. We have used 17 groups of $P = 4$ images (i.e., results are averaged over 17 experiments).

In addition to MPE, we show three particular examples (Figure 5-1, Figure 5-2, Figure 5-3, Figure 5-4, Figure 5-5 and Figure 5-6) of labelings obtained with the pair-wise method and the common labeling method. Figure 5-1 and Figure 5-2 show an example on the Asterix dataset, Figure 5-3 and Figure 5-4 an example on New York dataset and finally Figure 5-5 and Figure 5-6 an example on the Van Gogh dataset. See how the method is able to remove incorrect matches, increase the amount of point matches found and select better point matchings. The first case is clearly seen in the Asterix example, the common labeling is able to detect that the points from the belly of Obelix do not correspond to the top letters. The second case is exemplified in the New York, the common labeling is able to match a greater amount of points with a better accuracy. Finally, in the Van Gogh example, the common labeling method is able to correct several point matchings giving more than an acceptable result.

Figure 5-1: concrete labelling example of Asterix dataset obtained using pair-wise method.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Figure 5-2: concrete labelling example of Asterix dataset using obtained using common labelling method.

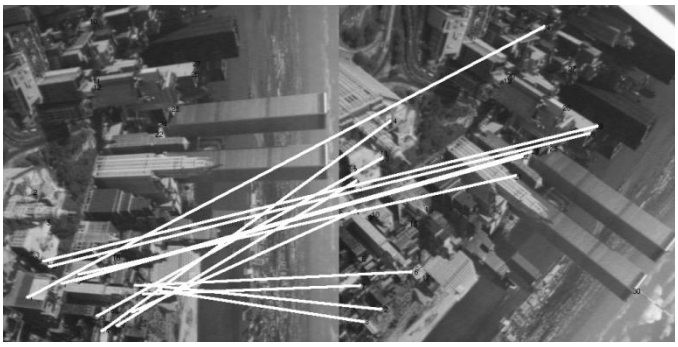


Figure 5-3: concrete labelling example of New York dataset using obtained using pair-wise method.

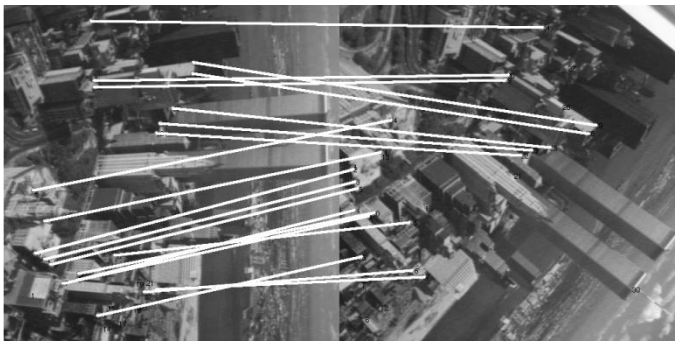


Figure 5-4: concrete labelling example of New York dataset using obtained using common labelling method.

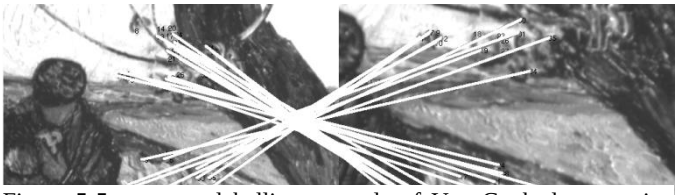


Figure 5-5: concrete labelling example of Van Gogh dataset using obtained using pair-wise method.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

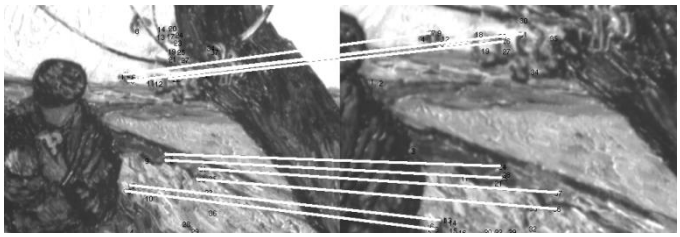


Figure 5-6: concrete labelling example of Van Gogh dataset using obtained using common labelling method.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Chapter 7

CONCLUSIONS AND FUTURE WORK

The thesis has addressed the problem of computing a common labeling among a set of graphs. The common labeling of a set of graphs is defined as the joint labeling from each graph of the set to a virtual node set. Each node of the virtual set conceptually identifies equivalent parts of the objects each individual graph represents. The common labeling concept is equivalent to the pair-wise minimization of the local labelings with transitivity restrictions between node correspondences.

The common labeling concept is of crucial importance in several graph pattern recognition applications. These applications include graph prototype synthesis and therefore all possible applications of graph prototypes.

Up to now, only few methods to compute the common labeling have been proposed. The initial ones date back to the 80s with the development of random graphs (Wong, Constant et al. 1990), and their performance highly depend on the prototype efficiency to model the data. More sophisticated algorithms appear later, in 2000 and 2008, (Jiang, Münger et al. 2001; Lozano, Escolano et al. 2009). These algorithms model the problem independently of the prototype construction. All state of the art algorithms lack on either performance or theoretical basis. The prototype based ones rely on modeling the graph set with very few data (two samples) at the initial steps, which is obviously impractical if data contains some amount of noise. In addition, they rely on the prototype performance which might be counterproductive. The main advantage is that these algorithms are faster than the others due to less graph matching operations are performed. The other two algorithms proposed to solve the common labeling problem are either too heuristic (Lozano, Escolano et al. 2009) or too computational demanding (Jiang, Münger et al. 2001). Considering the drawbacks of the existing algorithms and the performance they obtain, we have analyzed the

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

problem from the theoretical and practical point of view. This chapter summarizes the main conclusions of the work and overviews several future work.

1. CONCLUSIONS

We summarize the conclusions of this work from three points of view: theory, algorithms and applications. For each point the contributions and obtained results will be described.

1.1 Theory

Chapter 3 focuses on the graph edit distance measure to compare the similarity between two graphs. The main contribution has been to formalize several theoretical concepts related to the classes of costs. In addition, we characterize the shape of each class of cost and we develop intuition behind the labeling each class of cost produces. Besides, we give a formal description of the edit surface and we characterize its shape. Related to the new described properties of the graph edit distance several directions of its applicability on real applications are proposed. A well understanding of the graph edit distance is necessary to decide if the provided model is suitable for the problem at hand. In this way, the defined concepts enlarge the knowledge of the graph edit distance and may be determining on the solution of the problem one proposes.

With the aim of extending the pair-wise graph edit distance problem to the group-wise one, an effective method to compute the graph edit between two graphs was developed. The method provides a completely different formalization of the problem. This formalization, based on the concept of dominant sets, allows mapping the most purely combinatorial problem to an optimization one. Theoretical foundations to prove that both problems are equivalent are given. The advantage of the new formalization is twofold. On one hand, the developed theory describes a connection between dominant sets and the graph edit distance, this connection is not obvious since dominant sets usually represent a subset of node-to-node assignments and from them the edit path cannot be constructed. On the other hand, new theory describes a connection between the graph edit distance and evolutionary game theory, this connection allows using fast and reliable optimization algorithms to compute the solutions of the graph edit distance problem.

Related to the main goal of the thesis, we have formally characterized the common labeling problem and we have related it to the multiple isomorphism and the consistent multiple isomorphism problem. In addition,

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

a methodology to adapt the solutions is provided. Considering the given formalization, a probabilistic framework to compute the graph edit distance was developed. This new framework allows the construction of reliable and effective algorithms to compute sub-optimal common labeling solutions.

With respect to the application of the common labeling to real applications, experimental evidences of the relation of the common labeling with the graph prototypes synthesis problem are given. These experimental evidences show that the better the common labeling obtained, the better the prototypes that will be constructed. This connection represents a new point of view of graph prototype construction since the synthesis process and the construction process can run independently. This connection is of crucial importance in probabilistic graph prototypes, since, using the classical sequential or hierarchical synthesis, at the initial steps, a model must be constructed using very few samples of the random variable which produces sever modeling errors that enhance as the iterative synthesis goes on.

1.2 Algorithms

This thesis has provided to the scientific community several algorithms, which its efficiency has been experimentally proven. These algorithms are target several specific problems. We describe each independently in the following lines.

The first algorithm provided is addressed to compute the graph edit distance between two graphs. The proposed algorithm has been evaluated considering several state of the art algorithms. Considering the obtained results, it could be stated that the proposed algorithm represents an advance on the graph edit distance computation. Besides to the single problem of computing the distance between two graphs, the method has been evaluated on a clustering application. Results show that, in addition to obtaining better distance measures, the obtained distances allow a better separability between the clustered objects.

With respect to the main objective of the thesis, 6 algorithms have been provided: P-Dim Graduated Assignment, Agglomerative Graduated Assignment, Least squares, Average Alignment, Common Labeling Graduated Assignment and Common Labeling Dominant Sets. We can divide the algorithms in two types considering the approach of the problem they use. The first type considers 4 algorithms which are focused to compute a common labeling through a consistent multiple isomorphisms. The algorithms are: P-Dim Graduated Assignment, Agglomerative Graduated Assignment, Least squares and Common Labeling Dominant Set. The P-Dim Graduated Assignment algorithm obtains the consistent multiple isomorphism considering all graphs at once. This algorithm has been experimentally proven to be the most effective one. The Agglomerative

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Graduated Assignment and the Least squares algorithm rely on a pair-wise matching computed in an initial step. These algorithms also provide results that improve state of the art. The main advantage of these two algorithms with respect to the first one is the lower computational complexity. In addition, since individual pair-wise matchings are computed in an initial phase and they do not rely one on the other, these individual matchings can be computed in parallel, which becomes an advantage considering the exponential nature of the problem. The Common Labeling Dominant Set algorithm is proven to be more effective than methods that compute directly the common labeling in some datasets. However, its computational complexity is high. With respect to methods of the second type, we provide two algorithms: Average Alignment and Common Labeling Graduated Assignment. These methods compute directly a common labeling solution without previously computing a consistent multiple isomorphism. This direct computation allows the algorithms to reduce the computational complexity of the problem. The two algorithms improve state of the art algorithms. Best results have been obtained using the Common Labeling Graduated Assignment. Both algorithms share advantages and drawbacks. The main advantage is the speed to complete the matching process. Even the computational complexity of each iteration is higher than in the Common Labeling Dominant Set algorithm orders of magnitude less iterations are necessary if the graphs are similar. In addition, the amount of memory that the common labeling oriented algorithms use is quite lower than the algorithms that rely on computing a consistent multiple isomorphism. Moreover, often the cost matrix can be computed in an initial step of the algorithms which even increases the speed further. A specific drawback of this type of methods is that it is hard to analyze the theoretical framework, since the graduated assignment is a combination of heuristic intuitions and optimization mechanisms.

A general drawback of the algorithms presented in the thesis is the treatment of null assignations. To consider all possible matching solutions, a large amount of null nodes must be inserted into the initial graphs. This fact, even the computational complexity of the problem is the same, increases the size of the input data and considering the exponential nature of the problem this issue may produce a sever increase of the computational time.

With respect to the performance of both types of algorithms, it is important to highlight that common labeling oriented ones give better results with lower computational cost.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

1.3 Applications

Along the thesis, several applications have been used to evaluate the proposed algorithm. In most of the experiments, the proposed algorithms improved state of the art algorithms.

1.3.1 Interactive and Adaptive Graph Recognition.

Some of the new graph edit distance properties, formalized in this thesis, have been applied to learn, using an interactive and an adaptive framework, the graph edit distance constants. In each step, the framework aims to improve the knowledge on the graph edit constants given two graphs and the current bijection between them. The problem was reduced to compute the class of costs that belong to the given labeling. In particular, the shape of the class of cost was used to speed up the computation of all the classes of costs in a given region of the edit surface. Considering that each class of costs forms a convex polygon, the basic idea relies on that given two points that belong to the same class of costs all points between those also belong to the same class of costs. Using this property, the algorithm could save 90% of the cost computations to label each point of the edit grid to a class of costs. Thus, the learning algorithm becomes 90% faster than the naive solution based on computing all the bijections in the given region of the edit surface.

1.3.2 Graph clustering application.

We proposed an algorithm to compute the graph edit distance based on the concept of the dominant sets. Besides evaluating the cost improvement we also evaluate the effectivity of the computed costs in a clustering application. Thus, given a set of graphs where graphs belong to different classes, its pair-wise distances where computed using the presented algorithm and two other state of the art algorithms. Given this pair-wise matching matrix a clustering algorithm based on a peeling procedure was applied. Results show a great improvement on the minimization of the sum of intra class distances, however this improvement was expected since a improvement on the pair-wise distance matrix values was already observed. With respect to classification results some improvement were also observed. All tested algorithms, the Dominant Set matching, the Bipartite Graph Matching and the Graduated Assignment, gave an equivalent classification result when they were applied on the GREC dataset. On the Shapes 99 dataset, the presented algorithm improved the Bipartite Graph Matching and gave equivalent results than the Graduated Assignment. On the last dataset, the shapes 216, the Dominant Set algorithm improved the graduated assignment and gave equivalent results than the bipartite graph matching.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

As a conclusion, we can state that the presented algorithm, in addition of obtaining better graph edit distance results, is able to give better results on classification than state of the art algorithms. However, this improvement also depends on the problem since it may happen that improvement on the distance computation, if distance does not adequately model the problem, is not reflected with an improvement on classification.

1.3.3 Graph prototype construction and metric trees

Most of the evaluation of the common labeling algorithms has been based on the evaluation of the resulting common labeling cost. However, we also tested the common labeling graduated assignment over several real applications. In this section, we focus on graph prototype construction and an application of them, the metric trees. To evaluate the effectivity of the common labeling on computing a graph prototype, we first evaluated the goodness of the Generalized Median Graph synthesized. Thus, given a training set of graphs a Generalized Median Graph was constructed using the Common Labeling Graduated assignment, a hierarchical synthesis procedure and a classical genetic approach. With this three synthesis procedures we constructed a metric tree of graphs. The Generalized Median Graph constructed using the Common Labeling Graduated Assignment show improvement over the other methodologies. In addition, we used the proposed methodology to compute several other graph prototypes. Results also showed that the common labeling methodology could satisfactory be used to that aim.

As a general conclusion, it is important to say that global methodologies to compute a common labeling, such as the Common Labeling Graduated Assignment or the genetic approach are supposed (and we experimentally prove that) to achieve better prototype than local ones such as the Super-Graph approach in (Lozano, Escolano et al. 2009) or the Hierarchical synthesis. The basis of this affirmation relies on that they consider all the knowledge of the set.

1.3.4 Image registration

The last application that we considered is the group-wise registration of images. To that aim, the Common Labeling Graduated Assignment was modified to consider affine transformations between different images in the set. The proposed algorithm was evaluated over three different datasets that include zoom, rotation and shear. The improvement on the mean projection error was compared with the classical approach of using ICP to compute alignment and correspondences plus RANSAC for outlier detection. The common labeling approach showed a great improvement over that classical methodology. The improvement we observed was manifold. The proposed

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

method was able to discard incorrect matches and increase the amount of point matches found.

2. FUTURE WORK

In spite of the large set of contributions of the document, in each of the parts there is still room for improvement and some open questions are still left open. In the following points, we give details and directions to improve methodologies presented and some future research directions.

- In chapter 2 two different, but equivalent, formulations are given for the graph matching objective function (1.3) and (1.8). Even the objective function share the same optimal values, the shape of the hyper plane they generate seems to be quite different. In this way, several performed experiments showed that depending on the problem and the optimization algorithm both formulations give very different results. A deepest study in this direction may give theoretical foundations to conclude which one should be used.
- In chapter 3, several properties of the graph edit distance have been presented. These properties mainly characterize each class of costs and describe the shape of the associated edit surface. Several directions to apply the properties have been given in the chapter. We highlight one of them. We saw in the chapter that several suboptimal algorithms decrease its performance on computing the graph edit distance in particular areas of the edit space. Considering that, thanks to the new properties, it is possible to analyze and model this fact; filtering mechanisms can be developed to improve the decay of performance.
- In chapter 4, a new algorithm, based on Dominant sets, has been presented. The algorithm is addressed to compute the graph edit distance between a pair of graphs. The algorithm relies on a strong theoretical basis which ensures that maximal solutions of the graph edit distance problem are maximal solutions of the suggested model. The main drawback relies on the used optimization procedure. This procedure, similar to a gradient ascent method, needs on an initial starting point to start the optimization procedure. These initial points, were chosen randomly in the simplex. Better heuristics should be chosen to obtain still better results and, what is more important, in a faster way.
- Chapter 5 provides several algorithms to compute a common labeling between a set of graphs. Some of the algorithms or parts of them can be improved. Starting from the two first algorithms that compute a common labeling through a probabilistic hypercube, they need a post discretization process to obtain a discretize permutation matrix set. Consequently, the discretization process is performed by iteratively

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

choosing the best assignation possible until a bijection is found. Several approaches similar to the Hungarian algorithms could be developed to perform this discretization in a more global fashion, possibly improving the final bijection. The third method presented relies on the Least squares methodology to compute an approximation of the common labeling. This method computes the common labeling as the average of the possibly noisy local common labelings found by any independent bijection between the graphs of the set. The problem could be reformulated by modeling the system as an over-determined system where each equation is one of the noisy common labeling solutions. This approximation closely resembles the spirit of computing a linear regression between a set of samples. With respect to the dominant set algorithm, the main drawback that should be improved is the size of the association matrix. The matrix becomes too large with just few nodes per graph and few graphs per set. Analyzing the information that the association matrix contains, we see that just a small part of it contain node-to-node matching information, the resting part contains consistency restrictions. Considering this fact, we assume there must be another codification possible of the common labeling problem into the association graph framework which minimizes this large amount of consistency restrictions.

Regarding the chapter of applications, several future lines of research are considered.

- In the thesis, it is stated that the common labeling problem and the computation of graph prototypes is closely related, and experimental validation shows that in fact it is. However, theoretical connection should be found that connect both problems.
- The m-tree applications of the graph prototypes constructed using the common labeling show very different access rate and performance per prototype. We consider that an m-tree with several graph prototypes per node might improve both the access rate and the performance of the application. Besides, considering that the performance of the m-tree is related to the triangle inequality, if this property of the distance computations is not preserved, at query time, between the query object and the elements in the database most of the theoretical foundations of metric trees cannot be applied. Since the common labeling enforces this triangle inequality restrictions it would be of great interest to find a connection between querying algorithms and the common labeling problem.
- The group-wise registration framework, presented in chapter 6, have been shown to be a particular formalization of the common labeling

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

problem, where affine transformations are considered between points. This formalization of the group-wise matching problem has experimentally shown great improvement with respect to classical algorithms. However, the outlier detection is a bit tricky. New outlier detection mechanism, that theoretically fit the formalization in a clear form, should be provided.

Besides, in chapter 6, we highlighted that the main difference between the proposed method and state of the art methods to compute the transformation parameters rely on that we consider these parameters independent for each point. These considerations are useful when objects are moving in independent directions between images, however if images do not contain moving objects, this independence consideration may drive the algorithm to bad results. Some annealing parameter should be included in the algorithm to force that in the final results all transformation parameters are equivalent.

- With the aim of improving speed of some of the proposed algorithms, it will be interesting to get rid of the laborious task of increasing the graphs with a large amount of null nodes. Several works, such as (Fukagawa, Tamura et al. 2011) provided several formalizations where the null extensions of the graphs are not required. However, the work of (Fukagawa, Tamura et al. 2011) just consider the tree edit distance. It will be of great interest for the community to analyze if these theoretical foundations are extendable to the graph case.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

REFERENCES

Ambauen, R., S. Fischer, et al. (2003). "Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification." International conference on Graph based representations in pattern recognition

Bagdanov, A. D. and M. Worring (2003). "First order Gaussian graphs for efficient structure classification " Pattern Recognition 36(6): 1311-1324.

Ballabh, H. (1988). "A Fast Backtracking Algorithm to Test Directed Graphs For Isomorphism Using Distance Matrices." Information Processing Letters 29(2): 105-110.

Battiti, R. and F. Mascia (2007). An algorithm portfolio for the sub-graph isomorphism problem. International conference on Engineering stochastic local search algorithms: designing, implementing and analyzing effective heuristics.

Berretti, S., A. D. Bimbo, et al. (2004). A Graph Edit Distance Based on Node Merging. Image and Video Retrieval.

Berretti, S., A. D. Bimbo, et al. (2001). "Efficient Matching and Indexing of Graph Models in Content-Based Retrieval." Pattern Analysis and Machine Intelligence 23(10): 1089-1105.

Bin, L. and E. Hancock (2001). "Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition." Transactions on Pattern Analysis and Machine Intelligence 23(10): 1120-1136.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Björck, G. D. a. Å. (1974). Numerical Methods, Prentice-Hall.

Bomze, I. M. (1997). "Evolutions towards the maximum clique." Journal of Global Optimization 10: 143-164.

Bomze, I. M., M. Budinich, et al. (1999). "The Maximum Clique Problem." Handbook of combinatorial optimization: 1-74.

Bomze, I. M., M. Pelillo, et al. (2000). "Approximating the Maximum Weight Clique Using Replicator Dynamics." Transactions on Neural Networks 11(6): 1228 - 1241.

Bonev, B., F. Escolano, et al. (2007). Constellations and the unsupervised learning of graphs. International conference on Graph-based representations in pattern recognition: 340-350.

Bozkaya, T. and M. Ozsoyoglu (1999). "Indexing Large Metric Spaces for Similarity Search Queries." ACM transactions on Database Systems 24(3): 361-404.

Bridle, J. S. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters.

Bulo', S. R. and I. M. Bomze (2011). "Infection and immunization: a new class of evolutionary game dynamics." Games and Economic Behaviour 71: 193-211.

Bulo', S. R., M. Pelillo, et al. (2011). "Graph-Based Quadratic Optimization: A Fast Evolutionary Approach,." Computer Vision and Image Understanding 115: 984-995.

Bunke, H. (1998). Error-tolerant graph matching: A formal framework and algorithms SSPR '98/SPR '98 Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition.

Bunke, H. (1998). "On a relation between graph edit distance and maximum common subgraph." Pattern Recognition Letters 18(8): 689-694.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Bunke, H. (1999). "Error Correcting Graph Matching: On the Influence of the Underlying Cost Function." Transactions on Pattern Analysis and Machine Intelligence 21(9): 917-922.

Bunke, H. (2000). Graph matching: Theoretical foundations, Algorithms and applications. In Proceedings International Conference on Vision Interface.

Bunke, H., P. Foggia, et al. (2003). Graph clustering using the weighted minimum common supergraph. IAPR international conference on Graph based representations in pattern recognition Lecture Notes in Computer Science. 2726: 235-246.

Bunke, H., A. Münger, et al. (1999). "Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem." Pattern Recognition Letters 20(11-13): 1271–1277.

Bunke, H. and K. Riesen (2012). "Towards the unification of structural and statistical pattern recognition." Pattern Recognition Letters 37(7): 811-825.

Bunke, H. and K. Shearer (1998). "A graph distance metric based on the maximal common subgraph." Pattern Recognition Letters 19(3-4): 255-259.

Caelli, T. and S. Kosinov (2004). "An eigenspace projection clustering method for inexact graph matching." Transactions on Pattern Analysis and Machine Intelligence 26(4).

Caetano, T., J. McAuley, et al. (2009). "Learning Graph Matching." Transaction on Pattern Analysis and Machine Intelligence 31(6): 1048-1058.

Ciaccia, P., M. Patella, et al. (1997). "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces." The VLDB Journal: 426-435.

Conte, D., P. Foggia, et al. (2004). "Thirty Years Of Graph Matching In Pattern Recognition." International Journal of Pattern Recognition and Artificial Intelligence 18(3): 265-299.

Cootes, T., C. Twining, et al. (2010). "Computing Accurate Correspondences across Groups of Images " Pattern Analysis and Machine Intelligence 32(11): 1994-2005.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Cordella, L. P., P. Foggia, et al. (1999). Graph Matching: a Fast Algorithm and its Evaluation. International Conference on Pattern Recognition.

Cordella, L. P., P. Foggia, et al. (2004). "A (sub)graph isomorphism algorithm for matching large graphs " Transactions on Pattern Analysis and Machine Intelligence 26(10): 1367-1372.

Chan, K. P. and Y. S. Cheung (1992). "Fuzzy-attribute graph with application to Chinese character recognition " Transactions on Systems, Man, and Cybernetics 22(1): 153-160.

Christmas, W. J., J. Kittler, et al. (1995). "Structural Matching in Computer Vision Using Probabilistic Relaxation." Transaction on Pattern Analysis and Machine Intelligence 17(8): 749-764.

Demirci, M. F., Y. Osmanlioglu, et al. (2011). "Efficient many-to-many feature matching under the l1 norm." Journal Computer Vision and Image Understanding 115(7).

Dickinson, P., H. Bunke, et al. (2003). On graphs with unique node labels. International conference on Graph based representations in pattern recognition

Du, D.-Z. and P. M. Pardalos, Eds. (1998). Handbook of Combinatorial Optimization, Springer.

Emms, D., R. C. Wilson, et al. (2008). "Graph matching using the interference of continuous-time quantum walks." Pattern Recognition 42(5): 985–1002.

Fergus, R., P. Perona, et al. (2007). "Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition." International Journal of Computer Vision 71(3): 273-303.

Ferrer, M., E. Valveny, et al. (2009). "Median graph: A new exact algorithm using a distance based on the maximum common subgraph." 30(5): 579–588.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Ferrer, M., E. Valveny, et al. (2009). "Median graphs: A genetic approach based on new theoretical properties." Pattern Recognition 42(9): 2003-2012.

Ferrer, M., E. Valveny, et al. (2009). Graph-Based k-Means Clustering: A Comparison of the Set Median versus the Generalized Median Graph. International Conference on Computer Analysis of Images and Patterns. 5702/2009: 342-350.

Ferrer, M., E. Valveny, et al. (2010). "Generalized median graph computation by means of graph embedding in vector spaces." Pattern Recognition Letters 43(4): 1642-1615.

Finch, A. M., R. C. Wilson, et al. (1998). "An energy function and continuous edit process for graph matching." Journal Neural Computation 10(7): 1873-1894.

Fischler, M. and R. Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24(6): 381-395.

Fober, T., M. Mernberger, et al. (2009). "Evolutionary construction of multiple graph alignments for the structural analysis of biomolecules." Journal Bioinformatics 25(16).

Fosser, P., R. Glantz, et al. (2003). Swap strategies for graph matching. International conference on Graph based representations in pattern recognition

Fukagawa, D., T. Tamura, et al. (2011). "A clique-based method for the edit distance between unordered trees and its application to analysis of glycan structures." BMC Bioinformatics 12(1).

Gallagher, B. (2006). Matching Structure and Semantics: A Survey on Graph-Based Pattern Matching. AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection: 45-53.

Gao, X., B. Xiao, et al. (2010). "A survey of graph edit distance." Pattern Analysis and applications 13(1): 113-129.

Garey, M. and D. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Gautama, S., R. Bellens, et al. (2006). Relevance Criteria for Data Mining Using Error-Tolerant Graph Matching Combinatorial Image Analysis. 4040/2006: 277-290.

Gibbons, L. E., D. W. Hearn, et al. (1997). "Continuous Characterizations of the Maximum Clique Problem." Mathematics of Operations Research 22(3): 754-768.

Gold, S. and A. Rangarajan (1996). "A Graduated Assignment Algorithm for Graph Matching." Transaction on Pattern Analysis and Machine Intelligence 18(4): 377-388.

Gold, S., A. Rangarajan, et al. (1998). "New algorithms for 2d and 3d point matchin." Pattern Recognition 31: 1019-1031.

Gori, M., M. Maggini, et al. (2005). "Exact and Approximate Graph Matching Using Random Walks." Transaction on Pattern Analysis and Machine Intelligence 27(7): 1100 - 1111

Grunbaum, B. (2003). Convex Polytopes.

Harris, C. and M. Stephens (1988). A Combined Corner and Edge Detection. The Fourth Alvey Vision Conference.

Hartigan, J. A. and M. A. Wong (1979). "Algorithm AS 136: A K-Means Clustering Algorithm." Journal of the Royal Statistical Society. Series C 28(1): 100-108.

Hastie, T., R. Tibshirani, et al., Eds. (2009). The Elements of Statistical Learning.

He, H. and A. K. Singh (2006). Closure-Tree: An Index Structure for Graph Queries. International Conference on Data Engineering 38.

Hidovic, D. and M. Pelillo (2004). "Metrics for Attributed Graphs Based on the Maximal Similarity Common Subgraph." International Journal of Pattern Recognition and Artificial Intelligence 18(3).

Hlaoui, A. and S. Wang (2002). A New Algorithm for Graph Matching with Application to Content-Based Image Retrieval. International Workshop on Structural, Syntactic, and Statistical Pattern Recognition 291-300.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Hlaoui, A. and S. Wang (2006). "Median graph computation for graph clustering." Soft Computing 10: 47-53.

Horaud, R., F. Forbes, et al. (2011). "Rigid and articulated point registration with expectation conditional maximization." Pattern Analysis and Machine Intelligence 33: 587-602.

Hummel, R. and S. Zucker (1983). "On the foundations of relaxation labling processes." Pattern Analysis and Machine Intelligence 5(3): 267-287.

Jain, A. K. and D. Maltoni (2003). Handbook of Fingerprint Recognition, Springer-Verlag New York.

Jain, B. J. and F. Wysotzki (2005). "Solving inexact graph isomorphism problems using neural networks." Neurocomputing 63: 45-67.

Jian, B. and B. Vemuri (2005). A robust algorithm for point set registration using mixture of gaussians. International Conference on Computer Vision.

Jian, B. and B. Vemuri (2011). "Robust point set registration using gaussian mixture models." Pattern Analysis and Machine Intelligence 33: 1633-1645.

Jiang, X. and H. Bunke (1996). Including geometry in graph representations: A quadratic-time graph isomorphism algorithm and its applications. SSPR '96 Proceedings of the 6th International Workshop on Advances in Structural and Syntactical Pattern Recognition

Jiang, X., A. Münger, et al. (2001). "On Median Graphs: Properties, Algorithms, and Applications." Pattern Analysis and Machine Intelligence 23(10): 1144-1152.

Justice, D. and A. Hero (2006). "A binary linear programming formulation of the graph edit distance." Transactions on Pattern Analysis and Machine Intelligence 28(8): 1200-1214.

Keselman, Y., A. Shokoufandeh, et al. (2003). Many-to-many graph matching via metric embedding. Computer Vision and Pattern Recognition 850-857.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Kim, D. H., I. D. Yun, et al. (2010). "Attributed relational graph matching based on the nested assignment structure." Pattern Recognition 43(3): 914–928.

Klein, P., S. Tirthapura, et al. (2000). A Tree-Edit-Distance Algorithm for Comparing Simple, Closed Shapes. ACM-SIAM Symposium on Discrete Algorithms 696-704.

Kovesi, P. (2009). "http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/."

Krissinel, E. B. and K. Henrick (2004). "Common subgraph isomorphism detection by backtracking search." Journal Software Practice & Experience 34(6): 591-607.

Kuhn, H. (1955). "The Hungarian method for the assignment problem." Naval Research Logistics Quarterly 2(1-2): 83-97.

Lohmann, G. and D. Y. v. Cramon (2000). "Automatic labelling of the human cortical surface using sulcal basins." Medical Image Analysis 4(3): 179–188.

Lozano, M. (2008). Kermelized graph matching and clustering, Universidad de Alicante.

Lozano, M. A. and F. Escolano (2003). ACM attributed graph clustering for learning classes of images. International conference on Graph based representations in pattern recognition.

Lozano, M. A. and F. Escolano (2006). "Protein classification by matching and clustering surface graphs." Pattern Recognition 39(4): 539-551.

Lozano, M. A., F. Escolano, et al. (2009). "Region and constellations based categorization of images with unsupervised graph learning." Image and Vision Computing 27: 960-978.

Lozano, M. A., F. Escolano, et al. (2009). "Region and constellations based categorization of images with unsupervised graph learning." Image and Vision Computing 27(7): 960–978.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Lladós, J., E. Martí, et al. (2001). "Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs." TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 23(10): 1137-1143.

Llados, J., E. Marti, et al. (2001). "Symbol recognition by error-tolerant subgraph matching between region adjacency graphs " Transactions on Pattern Analysis and Machine Intelligence 23(10): 1137 - 1143

Macrini, D. A. (2003). Indexing and matching for view-based 3-d object recognition using shock graphs, Computer Science University of Toronto.

Massaro, A. and M. Pelillo (2001). "A Complementary Pivoting Approach to Graph Matching." Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition 469-479.

Massaro, A. and M. Pelillo (2003). "Matching graphs by pivoting." Pattern Recognition Letters 24(8): 1099-1106.

McGregor, J. (1982). "Backtrack search algorithms and the maximal common subgraph problem." journal of Software Practice and Experience 12(1): 23-34.

Messmer, B. and H. Bunke (1997). Fast error-correcting graph isomorphism based on model precompilation. Image Analysis and Processing, Lecture Notes in Computer Science. 1310/1997: 693-700.

Mikolajczyk, K., T. Tuytelaars, et al. (2011). "http://www.featurespace.org/." Retrieved 23/02/2011.

Motzkin, T. S. and E. G. Straus (1965). "Maxima for graphs and a new proof of a theorem of Turán " Canadian Journal of Mathematics 17: 533-540.

Mukherjee, L., V. Singh, et al. (2009). "Generalized median graphs and applications." JOURNAL OF COMBINATORIAL OPTIMIZATION 17(1): 21-44.

Munkres, J. (1957). "Algorithms for the Assignment and Transportation Problems." ournal of the Society for Industrial and Applied Mathematics 32-38.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Myers, R. and E. R. Hancock (2000). Least Committment Graph Matching by Evolutionary Optimisation. European Conference on Computer Vision. 1842/2000: 203-219.

Myers, R. and E. R. Hancock (2000). Selection Strategies for Ambiguous Graph Matching by Evolutionary Optimisation. International Workshops on Advances in Pattern Recognition

Myers, R., R. C. Wilson, et al. (1999). "Hancock: Bayesian Graph Edit Distance." Pattern Analysis and Machine Intelligence 22(6): 628-635.

Myronenko, A. and X. Song (2010). "Point Set Registration: Coherent Point Drift." Pattern Analysis and Machine Intelligence 32(12): 2262-2275.

Navarro, G. (2001). "A guided tour to approximate string matching." ACM Computing Surveys 33(1).

Neuhaus, M. and H. Bunke (2005). A graph matching based approach to fingerprint classification using directional variance. Audio- and Video-Based Biometric Person Authentication.

Neuhaus, M. and H. Bunke (2006). "Automatic learning of cost functions for graph edit distance." Information Sciences 177(1): 239-247.

Neuhaus, M. and H. Bunke (2007). A Probabilistic Approach to Learning Costs for Graph Edit Distance. International Conference on Pattern Recognition. 3.

Neuhaus, M. and H. Bunke (2007). A Quadratic Programming Approach to the Graph Edit Distance Problem G.-B. R. i. P. Recognition. 4538/2007.

Othman, F., R. Abdullah, et al. (2008). Bipartite Graph for Protein Structure Matching Modeling & Simulation: 928 - 933

Pavan, M. and M. Pelillo (2003). Dominant sets and hierarchical clustering. Proceedings of the Ninth IEEE International Conference on Computer Vision. 2.

Pavan, M. and M. Pelillo (2007). "Dominant Sets and Pairwise Clustering." IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1).

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Pawar, V. S. and M. A. Zaveri (2011). Graph Based Pattern Matching. Eighth International Conference on Fuzzy Systems and Knowledge Discovery.

Pelillo, M. (1999). "Replicator Equations, Maximal Cliques, and Graph Isomorphism." Neural Computation 11(8).

Pelillo, M. and A. Jagota (1995). "Feasible and Infeasible Maxima in a Quadratic Program for Maximum Clique." Journal of Artificial Neural Networks 2: 411-420.

Qiu, H. and E. Hancock (2006). "Graph matching and clustering using spectral partitions." Pattern Recognition 29(1): 22-34.

Rangarajan, A., H. Chui, et al. (1997). The softassign procrustes matching algorithm. International Conference on Information Processing in Medical Imaging.

Rangarajan, A., A. Yuille, et al. (1999). "Convergence properties of the softassign quadratic assignment algorithm." Neural Computation 11(6): 1455-1474.

Raveaux, R., J.-C. Burie, et al. (2010). "A graph matching method and a graph matching distance based on subgraph assignments." Pattern Recognition Letters 31(5): 394-406.

Riesen, K. and H. Bunke (2008). IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science: 287-297.

Riesen, K. and H. Bunke (2009). "Approximate graph edit distance computation by means of bipartite graph matching." Image and Vision Computing 27(4): 950-959.

Riesen, K., M. Neuhaus, et al. (2007). Bipartite Graph Matching for Computing the Edit Distance of Graphs. GRAPH-BASED REPRESENTATIONS IN PATTERN RECOGNITION. 4538/2007.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Rosenfeld, A., R. A. Hummel, et al. (1976). "Scene Labeling by Relaxation Operations." Transactions on Systems, Man, and Cybernetics 6: 420-443.

Rota-Bulò, S. and M. Pelillo (2008). "A generalization of the Motzkin–Straus theorem to hypergraphs." Optimization Letters 2(3): 287-295.

Sanfeliu, A. and K.-S. Fu (1983). "A Distance measure between attributed relational graphs for pattern recognition." IEEE transactions on systems, man, and cybernetics 13(3): 353-362.

Sanfeliu, A., F. Serratosa, et al. (2004). "Second-Order Random Graphs for modelling sets of Attributed Graphs and their application to object learning and recognition." International Journal of Pattern Recognition and Artificial Intelligence 18(3): 375-396.

Schellewald, C. and C. Schnörr (2005). Probabilistic subgraph matching based on convex relaxation. Energy Minimization Methods in Computer Vision and Pattern Recognition: 171--186.

Sebastian, T. B., P. N. Klein, et al. (2004). "Recognition of Shapes by Editing their Shock Graphs." IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5): 550 - 571

Serratosa, F., R. Alquézar, et al. (2002). "Synthesis of Function-Described Graphs and Clustering of Attributed Graphs." International Journal of Pattern Recognition and Artificial Intelligence 16(6): 621-656.

Serratosa, F., R. Alquézar, et al. (2003). "Function-Described Graphs for modelling objects represented by attributed graphs." Pattern Recognition 36(3): 781-798.

Serratosa, F. and A. Sanfeliu (2006). "Signatures versus histograms: Definitions, distances and algorithms." Pattern Recognition Letters 39(5): 921-934.

Serratosa, F., A. Solé-Ribalta, et al. (2011). Automatic Learning of Edit Costs Based on Interactive and Adaptive Graph Recognition. Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science. 6658: 152-162.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Serratosa, F., A. Solé-Ribalta, et al. (2011). K-nn Queries in Graph Databases Using M-Trees. Computer Analysis of Images and Patterns, Lecture Notes in Computer Science. 6854: 202-210.

Serratosa, F., A. Solé-Ribalta, et al. (2010). Graph Indexing and Retrieval Based on Median Graphs Mexican Conference on Pattern Recognition, Lecture Notes in Computer Science. 6256: 311-321.

Shapiro, L. and R. Haralick (1985). "A Metric for Comparing Relational Descriptions." Transactions on Pattern Analysis and Machine Intelligence 7(1): 90-94.

Shasha, D., J. T. L. Wang, et al. (2002). Algorithmics and applications of tree and graph searching. ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems 39-52.

Siddiqi, K., A. Shokoufandeh, et al. (1999). "Shock Graphs and Shape Matching." International Journal of Computer Vision 35(1): 13-32.

Sinkhorn, R. (1964). "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices." The Annals of Mathematical Statistics 35(2): 876-879.

Tang, J., B. Jiang, et al. (2011). "Graph Matching Based on Dot Product Representation of Graphs." Graph matching based on dot product representation of graphs: 175-184.

Todorovic, S. and N. Ahuja (2007). "Region-Based Hierarchical Image Matching." International Journal of Computer Vision 78(1).

Torsello, A. and E. Hancock (2007). "Graph embedding using tree edit-union." Pattern Recognition 40(5): 1393-1405.

Torsello, A. and E. R. Hancock (2003). "Computing approximate tree edit distance using relaxation labeling." Pattern Recognition Letters 24(8): 1089–1097.

Torsello, A., A. Robles-Kelly, et al. (2007). "Discovering Shape Classes using Tree Edit-Distance and Pairwise Clustering." International Journal of Computer Vision 72(3): 259-285.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Tsai, W.-H. and K.-S. Fu (1979). "Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis." Transactions on Systems, Man and Cybernetics 9(12): 757 - 768.

Ullmann, J. R. (1976). "An algorithm for subgraph isomorphism." Journal of ACM 23(1).

Umeyama, S. (1988). "An eigen decomposition approach to weighted graph matching problems." Transaction on Pattern Analysis and Machine Intelligence 10(5): 695 - 703

Wang, F., B. Vemuri, et al. (2008). "Simultaneous Nonrigid Registration of Multiple Point Sets and Atlas Construction " Pattern Analysis and Machine Intelligence 30(11): 2011-2022.

Wang, H. and E. Hancock (2009). "Probabilistic relaxation labelling using the Fokker-Planck equation." Pattern Recognition 41(11): 3393-3411.

Watson, C. and C. Wilson (1992). NIST Special Database 4, Fingerprint Database. N. I. o. S. a. Technology.

Weskamp, N., E. Hullermeier, et al. (2007). "Multiple Graph Alignment for the Structural Analysis of Protein Active Sites." Transactions on Computational Biology and Bioinformatic 4(2).

White, D. and R. C. Wilson (2008). Parts Based Generative Models for Graphs. International Conference on Pattern Recognition.

White, D. H. (2009). Generative Models for Graphs, Univeristy of York.

Wilson, R., A. Evans, et al. (1995). "Relational matching by discrete relaxation " Image and Vision Computing 13(5): 411-421.

Wilson, R. C. and E. R. Hancock (1997). "Structural Matching by Discrete Relaxation." Transactions on Pattern Analysis and Machine Intelligence 19(6): 634-648.

Williams, G. (1990). "Overdetermined Systems of Linear Equations." The American Mathematical Monthly 97(6): 511-513

Williams, M., R. Wilson, et al. (1997). "Multiple Graph Matching with Bayesian Inference." Pattern Recognition Letters 18: 1275-1281.

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

Wong, A. and M. You (1985). "Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition." Transaction on Pattern Analysis and Machine Intelligence PAMI-7(5): 599-609.

Wong, A. K. C., J. Constant, et al. (1990). Random Graphs, World Sci. Pub. Co.

Wyk, B. and M. Wyk (2004). "A POCS-Based Graph Matching Algorithm." 26(11): 1526-1530.

Xia, S. and E. Hancock (2008). Clustering Using Class Specific Hyper Graphs Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science. 5342: 318–328.

Xia, S. and E. Hancock (2009). Learning Class Specific Graph Prototypes. Image Analysis and Processing. 5716/2009: 269-277.

Yan, X., P. S. Yu, et al. (2004). Graph indexing: a frequent structure-based approach. ACM SIGMOD international conference on Management of data

ZHANG, Z. (1992). "Iterative Point Matching for Registration of Free-form Curves." International Journal of Computer Vision 13(2): 119-152.

Zhao, G., B. Luo, et al. (2007). Using Eigen-Decomposition Method for Weighted Graph Matching. International conference on Advanced intelligent computing theories and applications.

Zhu, Y., L. Qin, et al. (2001). High efficiency and quality: large graphs matching. International conference on Information and knowledge management

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

RELATED PUBLICATIONS

Journals

2 David Sánchez, Albert Solé-Ribalta, Montserrat Batet, Francesc Serratosa: Enabling semantic similarity estimation across multiple ontologies: An evaluation in the biomedical domain. Journal of Biomedical Informatics 45(1): 141-155 (2012) (Impact Factor: 1.719 5-Year: 2.245, quartile: Q1, area: computer science)

1 Albert Solé-Ribalta, Francesc Serratosa: Models and algorithms for computing the common labelling of a set of attributed graphs. Computer Vision and Image Understanding 115(7): 929-945 (2011) (Impact Factor: 2.404 5-Year: 2.730, quartile: Q2, area: computer science)

Conferences

12 Nicola Rebagliati, Albert Solé-Ribalta, et al. Title Computing the Graph Edit Distance Using Dominant Sets. International Conference on Pattern Recognition 2012. (Accepted. To appear in 2012).

11 A. Solé, G. Sanromà, F. Serratosa and R. Alquézar, "Group-wise sparse correspondences between images based on a common labelling approach", ", International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP2012, Rome, Italy, Volume 1, pp: 269-278, 2012

10 Francesc Serratosa, Albert Solé-Ribalta, Xavier Cortés: K-nn Queries in Graph Databases Using M-Trees. CAIP (1) 2011: 202-210

9 David Ródenas, Francesc Serratosa, Albert Solé-Ribalta: Parallel Graduated Assignment Algorithm for Multiple Graph Matching Based on a Common Labelling. GbRPR 2011: 132-141

UNIVERSITAT ROVIRA I VIRGILI
MULTIPLE GRAPH MATCHING AND APPLICATIONS
Albert Sole Ribalta
Dipòsit Legal: T.1211-2012

8 Francesc Serratosa, Albert Solé-Ribalta, Xavier Cortés: Automatic Learning of Edit Costs Based on Interactive and Adaptive Graph Recognition. GbRPR 2011: 152-163

7 Albert Solé-Ribalta, Francesc Serratosa: Exploration of the Labelling Space Given Graph Edit Distance Costs. GbRPR 2011: 164-174

6 David Ródenas, Francesc Serratosa, Albert Solé-Ribalta: Graph Matching on a Low-Cost and Parallel Architecture. IbPRIA 2011: 508-515

5 Albert Solé-Ribalta, Francesc Serratosa: A Probabilistic Framework to Obtain a Common Labelling between Attributed Graphs. IbPRIA 2011: 516-523

4 Francesc Serratosa, Albert Solé-Ribalta, Enric Vidiella: Graph Indexing and Retrieval Based on Median Graphs. MCPR 2010: 311-321

3 Albert Solé-Ribalta, Francesc Serratosa: Graduated Assignment Algorithm for Finding the Common Labelling of a Set of Graphs. SSPR/SPR 2010: 180-190

2 Albert Solé-Ribalta, Francesc Serratosa: On the Computation of the Common Labelling of a Set of Attributed Graphs. CIARP 2009: 137-144

1 Albert Solé-Ribalta, Francesc Serratosa: A Structural and Semantic Probabilistic Model for Matching and Representing a Set of Graphs. GbRPR 2009: 164-173