Graduated Assignment Algorithm for Finding the Common Labelling of a set of Graphs[1]

Albert Solé-Ribalta, Francesc Serratosa

Universitat Rovira i Virgili (Tarragona, Spain)
{francesc.serratosa, albert.sole}@urv.cat

Abstract. In pattern recognition applications, it is useful to represent objects by attributed graphs considering their structural properties. Besides, some graph matching problems need a Common Labelling between vertices of a set of graphs. Computing this Common Labelling is an NP-complete problem. State-of-the-art algorithms are composed by two steps: in the first, they compute all pairwise labellings among the graphs and in the second, they combine this information to obtain a Common Labelling. The drawback of these methods is that global information is only considered in the second step. To solve this problem, by reducing the Common Labelling problem to the quadratic assignment one, all graphs nodes are labelled to a virtual structure whereby the Common Labeling is generated using global information. We tested the algorithm on both real-world and synthetic data. We show that the algorithm offers better performance than a reference method with same computational cost.

Keywords: Graduated Assignment, Multiple graph matching, graph common labelling, inconsistent labelling, softassign.

1 Introduction

From 80's, graphs have increase its importance in Pattern Recognition, being one of the most powerful characteristics the abstraction they achieve. Therefore, the same structure is able to represent a wide sort of problems from image understanding to interaction networks. Consequently, algorithms based on graph models are suitable in a very large problem space. There is an interesting review of graph representation models, graph matching algorithms and its applications in [1].

Sometimes in graph based Pattern Recognition applications, given a set of graphs, which all represent equivalent or related structures, it is required to find global consistent correspondences among all those graphs. These correspondences are called a Common Labelling (CL). Reference applications could be found in [2], where representations obtained from Infra-red, Optical, Cartographic and SAR images must

[1] This research is supported by Consolider Ingenio 2010 (CSD2007-00018), by the CICYT (DPI 2007-61452) and by the Universitat Rovira I Virgili through a PhD research grant.

be combined or in [3] where a prototype has to be synthesized from noisy data representing the same object.

Unfortunately, only a few techniques to compute these correspondences have been developed when the elements are represented by Attributed Graphs (AGs). Among them we could name: [4] where optimal pairwise labelings are required or [5] and [6] where, in this case, the CL computation is based on sub-optimal pairwise labelings. Although [5] is quite more effective than [6], both share the same weakness: the use of pairwise labelings, where a simple labeling error taken at initial stages could derive in a bad global result. Moreover [5] have tendency to add extra nodes in the final CL, which might be not desired in some applications. In [2], Williams et al. introduce a method, which could induce a solution for this problem. However, this method is not extensible to N graphs. Another method, which seems to solve both problems, was published in [7]. Nevertheless, its high computational complexity makes its use infeasible with large graphs sets.

In this article, we present an energy function that represents the global cost of a given CL. Moreover, we present an algorithm, similar to the Graduated Assignment algorithm presented in [8] that iteratively seeks for a CL that maximizes this energy.

The document is structured as follows. In Section 2, we present some theoretical basis of the CL problem. In Section 3 and 4, the Graduated Assignment algorithm for graph matching [8] and our new algorithm are presented. The evaluation of our method is presented in Section 5. Finally, Section 6 finalizes the article with some conclusions.

2 Definitions

Definition 1. Attributed Graph: Let Δ_v and Δ_e denote the domains of possible values for attributed vertices and arcs, respectively. An attributed graph AG over $(\Delta_v$ and $\Delta_e)$ is defined by a tuple $AG=(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v=\{v_k \mid k = 1,...,R\}$ is the set of vertices (or nodes), $\Sigma_e \in \{e_{ij} \mid i,j \in \{1,...,R\}, i \neq j\}$ is the set of arcs (or edges) and $\gamma_v:\Sigma_v \rightarrow \Delta_v$, $\gamma_e : \Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs respectively. In case it is required, any AG can be extended with null nodes. A null node is a special AG node which has special attribute Ø ∈ Δv

Definition 2. Isomorphism between AGs: 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 AGs. If the selected graphs have initially different node size or it is desired to permit some extra null to vertex labelings, G^p and G^q can be extended with any number of null nodes. Besides, let T be a set of isomorphisms between two vertex sets Σ_v. The isomorphism $f^{p,q}:\Sigma_v^p \rightarrow \Sigma_v^q$, $f^{p,q} \in T$, assigns each vertex from G^p to only one vertex of G^q. There is no need to define the arcs isomorphism since they are mapped accordingly to the node isomorphism of their terminal nodes.

Definition 3. Cost and Distance between AGs: Let f^{pq} be the isomorphism $f^{p,q}:\Sigma_v^p \rightarrow \Sigma_v^q$ that assigns each vertex from G^p to a vertex of G^q. The cost of this isomorphism, $C^G(G^p,G^q,f^{p,q})$ is a function that represents how similar are the AGs and how correct is the isomorphism. We consider this cost to be:

$$C^G\left(G^p, G^q, f^{p,q}\right) = \sum_{a=1}^{R}\sum_{b=1}^{R}\sum_{i=1}^{R}\sum_{j=1}^{R} F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \qquad (1)$$

where $F^{p,q}[a,i]$ is a permutation matrix which values are *1* if $f^{pq}(v_a^p) = v_i^q$ and $C_{ai,bj}^{p,q}$ represents the cost of matching nodes v_a^p to v_i^q and v_b^p to v_j^q plus the cost of matching the corresponding edge $e_{ab}{}^p$ to $e_{ij}{}^q$.
Usually, $C^G=0$ represents that both AGs are identical and that the isomorphism captures this similarity. The distance D between two AGs is defined to be the minimum cost of all possible isomorphisms $f^{p,q}$. That is, $D\left(G^p, G^q\right) = \min_{f^{p,q} \in T} C^G\left(G^p, G^q, f^{p,q}\right)$. We say that the isomorphism $f^{p,q}$ is *optimal* if it is the one used to compute the distance.

Definition 4. Multiple Isomorphism (MI) of a set of AGs: Let $\Gamma=\{G^1, G^2, ..., G^N\}$ be a set of N AGs. We say that the set φ is a Multiple Isomorphism of Γ if it contains one and only one isomorphism between elements, $\varphi = \{f^{1,2}, ..., f^{2,1}, ..., f^{N,N}\}$.

We assume that the AGs have R nodes. If it is not the case, the AGs would have to be extended with null nodes. We say that a multiple isomorphism is *consistent* if concatenating all the isomorphisms we can define disjoint *partitions* [5] of vertices. Every *partition* is supposed to contain one and only one vertex per each AG and, in addition, every vertex must belong to only one partition. **Fig 1a** shows a *Consistent Multiple Isomorphism* between three AGs, being $R=2$. We can distinguish two partitions, *P1* and *P2*. **Fig 1b.** shows the same AGs with an *Inconsistent Multiple Isomorphism*, consequently partitions can not be defined.

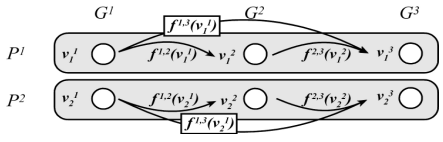

Fig. 1a: Consistent MI.

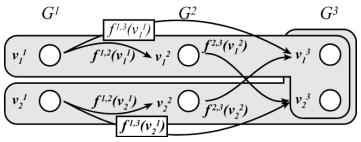

Fig. 1b: Inconsistent MI.

We define the cost of a MI as the addition of the costs of all isomorphisms in φ:

$$C^{MI}(\varphi) = \sum_{p=1}^{N}\sum_{q=1}^{N}\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R} F^{p,q}[a,i] \cdot F^{p,q}[b,j] \cdot C_{ai,bj}^{p,q} \quad (2)$$

Definition 5. Consistent Multiple Isomorphism of a set of AGs (CMI): Let φ be a Multiple Isomorphism of Γ. φ is a CMI of Γ if it fulfils that:

$$f^{q,k}\left(f^{p,q}\left(v_i^p\right)\right) = f^{p,k}\left(v_i^p\right), \ 0 < p,q,k \le N, 0 < i \le R \quad (3)$$

We define the cost of a CMI as the cost of the related MI. The Optimal Consistent Multiple Isomorphism (OCMI) is the CMI with the minimum cost. Note that, the cost of the OCMI may be obtained by non-optimal isomorphisms since it is restricted to be consistent.

Definition 6. Optimal Consistent Multiple Isomorphism of a set of AGs (OCMI): Let φ be a CMI of Γ. φ is an Optimal Consistent Multiple Isomorphism (OCMI) of Γ if it fulfils that $\varphi = \arg\min_{f^{pq} \in T} \sum_{\forall G^p, G^q} C^G\left(G^p, G^q, f^{p,q}\right)$.

Given Γ, we can define a Common Labelling (CL) which is a bijective mapping between all graph nodes in the AGs to a virtual structure. We construct this CL through a CMI. The CMI requirements are mandatory due to if not, the CL would not

be a bijective since an AG node would have to be labeled to several nodes of the virtual structure.

Definition 7. Common Labelling of a set of AGs (CL): Let φ be a CMI of Γ and let L be a vertex set, $L \in \Sigma_v$. The Common Labelling $\psi = \{ h^1, h^2, \ldots, h^n \}$ is defined to be a set of bijective mappings from the vertices of AGs to L as follows:

$$h^1(v_i^1)=i,\ h^p(v_i^p)=h^{p-1}(v_j^{p-1}),\ 1 \le i,j \le R,\ 2 \le p \le N, \text{ being } f^{p-1,p}(v_j^{p-1})=v_i^p. \quad (4)$$

Fig. 2 illustrates this definition.

Finally, the Optimal Common Labelling of a set is a CL computed through an OCMI. The prototype or representative of the set synthesized using this CL would be the best representative, from the statistical point of view, since the sum of the costs of each pair of AGs, considering the global consistency requirement, is the lowest among all possible CL.

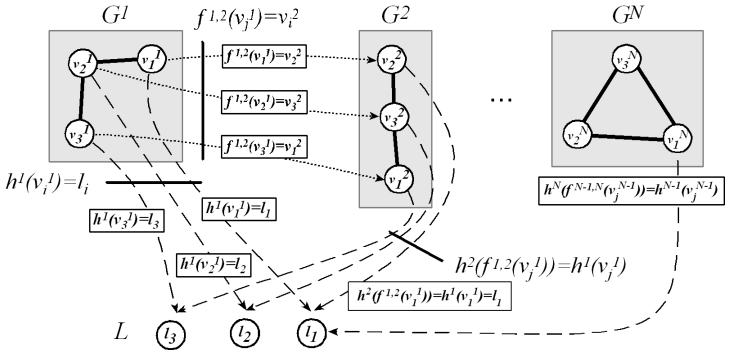

Fig. 2: From a CMI to a CL.

Definition 8. Optimal Common Labelling of a set of AGs (OCL): Let ψ be a CL of Γ computed by a CMI φ. We say that ψ is an Optimal Common Labelling (OCL) of Γ if φ is an OCMI of Γ.

3 Common Labelling framework

Given two graphs G^p and G^q, there are several error-tolerant graph matching algorithms that return the best isomorphism $f^{p,q}$ between them, given a minimization criteria. Considering that these graphs have a degree of disturbance and also the exponential complexity of the problem, some of these algorithms [8], [9], [10], [11] do not return exactly isomorphism $f^{p,q}$ but a probability matrix related to it. We represent this matrix by $P_f^{p,q}$ where each cell contains:

$$P_f^{p,q}[i,j] = Prob(f^{p,q}(v_i^p) = v_j^q) \quad (5)$$

To adapt the CL problem to the matching problem, we define the probability of match between a graph node v_i^p and a virtual node l_j as $P_h^p[i,j]=Prob(h^p(v_i^p)=l_j)$ (Fig. 3). Both, $P_f^{p,q}$ and P_h^p are stochastic matrices [12], note that $F^{p,q}$ in (1) and (2) is a special case of $P_f^{p,q}$, when $P_f^{p,q}$ is composed by zeros and ones. At the end of the proposed

algorithm, it is necessary to convert $P_f^{p,q}$ into $f^{p,q}$ and $P_h^{\ p}$ into h^p. There are several techniques to find these isomorphisms, e.g. [8], [13], which are out of the scope of this paper. We will indentify this discretization process as Λ.

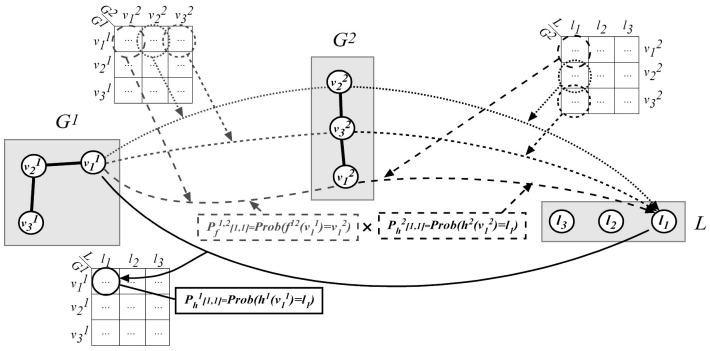

Fig. 3: Probability of matching $v_1^{\ 1}$ to l_1.

We consider, as Fig 3 depicts, that the probability of matching a vertex v_i^p of graph G^p, to a vertex l_j of the virtual structure L is the probabilistic union of all the paths that goes through the nodes of a third graph G^q. That is,

$$Prob\left(h^p\left(v_i^p\right)=l_j\right)=Prob\left\{\begin{array}{c}\left[f^{p,q}\left(v_i^p\right)=v_1^q\wedge h^q\left(v_1^q\right)=l_j\right]\vee\left[f^{p,q}\left(v_i^p\right)=v_2^q\wedge h^q\left(v_2^q\right)=l_j\right]\vee\\ \vee\ldots\vee\left[f^{p,q}\left(v_i^p\right)=v_R^q\wedge h^q\left(v_R^q\right)=l_j\right]\end{array}\right\}, \tag{6}$$

Combining (6) with P_f and P_h definitions and assuming independence of events we have:

$$P_h^p[i,j]=\sum_{k=1}^{R}P_f^{p,q}[i,k]\cdot P_h^q[k,j]\quad\text{from where}\quad P_h^p=P_f^{p,q}\cdot P_h^q \tag{7}$$

In a similar way, we could infer that $P_f^{p,q}=P_h^p\cdot(P_h^q)^T$.
Hence, following (7) we could obtain $P_h^{\ p}$ in several equivalent ways if $\Lambda(\varphi)$ is a CMI,

$$\begin{aligned}&h^1=\Lambda(P_h^1)=\Lambda(identity\_matrix)\\&h^2=\Lambda(P_h^2)=\Lambda(P_f^{2,N}\cdot P_h^N)=\Lambda(P_f^{2,N-1}\cdot P_h^{N-1})=\ldots=\Lambda(P_f^{2,1}\cdot P_h^1)=\Lambda(P_f^{2,1})\\&\ldots\\&h^N=\Lambda(P_h^N)=\Lambda(P_f^{N,N-1}\cdot P_h^{N-1})=\Lambda(P_f^{N,N-2}\cdot P_h^{N-2})=\ldots=\Lambda(P_f^{N,1}\cdot P_h^1)=\Lambda(P_f^{N,1})\end{aligned} \tag{8}$$

However, in real data (due to distortion on the object representation and distortion induced by sub-optimality of the matching algorithms), it is usual that [2]:

$$\begin{aligned}(P_h^p)'=P_f^{p,1}\cdot P_h^1,(P_h^p)''=P_f^{p,2}\cdot P_h^2,\ldots,(P_h^p)^{''\ldots'}=P_f^{p,N}\cdot P_h^N\rightarrow\\\rightarrow\Lambda((P_h^p)')\neq\ldots\neq\Lambda((P_h^p)^{''\ldots'})\end{aligned} \tag{9}$$

For this reason, probabilities P_h^p cannot be computed directly through matrices $P_f^{p,q}$, as in (8), when we cannot assume that $P_f^{p,q}$ will compose a CMI.

In this article, we propose and algorithm for the computation of a suboptimal solution to the CL problem, the algorithm is inspired in the Graduated Assignment. For ease of understanding, we first present an overview of the Graduated Assignment algorithm to later introduce the proposed algorithm.

4 The Graduated Assignment algorithm

The Graduated Assignment algorithm is probably the most popular algorithm to compute a suboptimal solution for the graph isomorphism problem. Its cornerstone is how it reduces the referenced problem to the quadratic assignment problem. The proposed development starts by defining the energy of an isomorphism as:

$$E^{G^p,G^q} = -\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R} P_f^{p,q}[a,i]\cdot P_f^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \tag{10}$$

They approximate E^{G^p,G^q}, at point $\left(P_f^{p,q}\right)^0$, using Taylor series expansion as:

$$\begin{aligned} E^{G^p,G^q} \approx (E^{G^p,G^q})' = &-\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R}\left(P_f^{p,q}[a,i]\right)^0\cdot\left(P_f^{p,q}[b,j]\right)^0\cdot C_{ai,bj}^{p,q} \\ &-\sum_{a}^{R}\sum_{i}^{R}\left[\sum_{b}^{R}\sum_{j}^{R}\left(P_f^{p,q}[b,j]\right)^0\cdot C_{ai,bj}^{p,q}\right]\left(P_f^{p,q}[a,i]-\left(P_f^{p,q}[a,i]\right)^0\right) \end{aligned} \tag{11}$$

analyzing the approximation it is seen that:

$$\arg\min\{E'\} \equiv \arg\max\left\{\sum_{a=1}^{R}\sum_{i=i}^{R} Q_{a,i}^{p,q}\cdot P_f^{p,q}[a,i]\right\},\ Q_{a,i}^{p,q} = \left[\sum_{b}^{R}\sum_{j}^{R}\left(P_f^{p,q}[b,j]\right)^0 C_{ai,bj}^{p,q}\right] \tag{12}$$

The algorithm proposed in [8] minimizes (10) under the assumption that it minimizes at same point as (12) is maximized. In this way, the problem is equivalent to the quadratic assignment one, where Q represents a cost matrix, and P_f represents a stochastic matrix [12] which contains the desired assignation probability.

The Graduated Assignment algorithm proceeds in the following way: start with a valid P_f, compute cost matrix Q given by (12), apply softassign to compute P_f and start again. A pseudo code of the Graduated Assignment is listed in Algorithm 1.

```
Program Graduated_Assignment input G^p,G^q returns f
 Initialise P_f^{p,q}
 Begin A: (Do A until β ≥ β_f)
  Begin B: (Do B until Q_{a,i}^{p,q} converges)
   Q_{a,i}^{p,q} = Σ_{b=1}^{R} Σ_{j=1}^{R} (P_f^{p,q}[b,j])^0 · C_{ai,bj}^{p,q}
   P_f^{p,q}[a,i] = exp(β · Q_{a,i}^{p,q})
   Begin C:
    P_f^{p,q}[a,i] = P_f^{p,q}[a,i] / Σ_{a=1}^{R} P_f^{p,q}[a,i]
    P_f^{p,q}[a,i] = P_f^{p,q}[a,i] / Σ_{i=1}^{R} P_f^{p,q}[a,i]
   End C
  End B
 End A
 f = Λ(P_f^{p,q})
End Program
```

Algorithm. 1. GA algorithm.

```
Program CL input Γ returns ψ
 Initialise P_h
 Begin A: (Do A until β ≥ β_f)
  Begin B: (Do B until Q^p_{a,ω_1} converges)
   Compute Q^p_{a,ω_1} (Alg. 3 or Alg. 4)
   P_h^p[a,i] = exp(β · Q^p_{a,ω_1})   1<p<=N
   Begin C:
    P_h^p[a,i] = P_h^p[a,i] / Σ_{a=1}^{R} P_f^p[a,i]   1<p<=N
    P_h^p[a,i] = P_h^p[a,i] / Σ_{i=1}^{I} P_h^p[a,i]   1<p<=N
   End C
  End B
 End A
 Ψ = Λ(P_h^p)
End Program
```

Algorithm 2. CL algorithm.

5 N-Graduated Assignment for the CL problem

The methodology that we present applies a similar procedure as the Graduated Assignment methodology to solve the CL problem. The proposed algorithm instead of computing an isomorphism of two graphs, it computes the isomorphism of a set of graphs Γ and in addition, it imposes to those isomorphisms to be consistent (3).

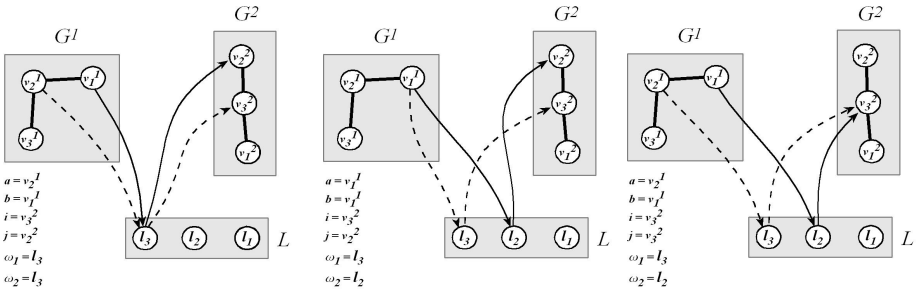

Fig. 4a: non valid: $\omega_1=\omega_2$. **Fig. 4b:** non valid: $a=b$. **Fig. 4c:** non valid: $i=j$.

Due to our objective is to compute a CL, our new energy function depends on the probabilities P_h instead of P_f. Nevertheless, the CL has to represent consistent and bijective isomorphisms between the involved graphs and the virtual structure, for this reason, we impose the restrictions $a\neq b$, $i\neq j$ *and* $\omega_1\neq\omega_2$. Fig. 4a,b,c shows non valid isomorphisms. Our new energy function is,

$$E^{CL}=\sum_{\forall p\in\Gamma}\sum_{\forall q\in\Gamma}^{q\neq p}\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1,b\neq a}^{R}\sum_{j=1,j\neq i}^{R}\left(\sum_{\omega_1=1}^{R}P_h^p[a,\omega_1]\cdot P_h^q[i,\omega_1]\cdot\left[\sum_{\omega_2=1,\omega_2\neq\omega_1}^{R}P_h^p[b,\omega_2]\cdot P_h^q[j,\omega_2]\right]\right)\cdot C_{ai,bj}^{p,q} \tag{13}$$

From (13) we compute the Taylor series expansion deducing that in our case Q is given by:

$$\begin{aligned}Q_{a,\omega_1}^{p}&=\frac{\partial E^{CL}}{\partial P_h^p[a,\omega_1]}=\\&=\sum_{\forall q\in\Gamma}^{q\neq p}\sum_{i=1}^{R}\left(\sum_{b=1,b\neq a}^{R}\sum_{j=1,j\neq i}^{R}P_h^q[i,\omega_1]\cdot\left[\sum_{\omega_2=1,\omega_2\neq\omega_1}^{R}P_h^p[b,\omega_2]\cdot P_h^q[j,\omega_2]\right]\cdot C_{ai,bj}^{p,q}\right)\end{aligned} \tag{14}$$

Finally, Algorithm 2 obtains a *CL* ψ given a set of graphs Γ. Note that in (4), we impose $h^1(v_i^1)=l_i$. For this reason, in the algorithm we present, we impose P_h^1 to be the identity matrix throughout the iterative process. This requirement is due to the fact that the virtual structure does not contain any type of attributes nor structure. Forcing nodes of G^1 to concrete nodes of the virtual structure L, we force the other graphs to label each other according to this prior labeling. The other probability matrices can be initialized to any stochastic matrix.

Function *Exact_Q* computes Q (14) with a cost of $O(N^2\cdot R^6)$(Algorithm 3). But, with the aim of reducing this cost, we have relaxed constraint $\omega_1\neq\omega_2$ in (14) (Fig. 4.a) which allows to compute an approximation of (14) with a cost of $O(N^2\cdot R^4)$. Algorithm 4 shows the pseudocode. We don't show evaluation results of the *Exact_Q* due to the results are equivalent to the approximation ones. Moreover, it can be proven that, with large size of Γ, the noise introduced by the non valid isomorphism when $\omega_1=\omega_2$ in the approximation algorithm is not significant.

```
Function Exact_Q input P_h, Γ returns Q
for ∀p ∈ Γ
 for a = 1..R
  for ω_1 = 1..R
   Q^p_{a,ω_1} = 0
   for ∀q ∈ Γ, q ≠ p
    for i = 1..R
     for b = 1..R, b ≠ a
      for j = 1..R, j ≠ i
       v_2 = 0
       for ω_2 = 1..R, ω_2 ≠ ω_1
        v_2 = v_2 + P_h^p[b, ω_2] · P_h^q[j, ω_2]
       end
       Q^p_{a,ω_1} = Q^p_{a,ω_1} + P_h^q[i, ω_1] · v_2 · C^{p,q}_{ai,bj}
      end
     end
    end
   end
  end
 end
end Function
```

Algorithm 3. Calculus of Q.

```
Func Approx_Q input P_h, Γ returns Q
for ∀p ∈ Γ
 Q^p = [0]
 for ∀q ∈ Γ, q ≠ p
  P_f^{p,q} = P_h^p · (P_h^q)^T
  for a = 1..R
   for i = 1..R
    v_1 = 0
    for b = 1..R, b ≠ a
     for j = 1..R, j ≠ i
      v_1 = v_1 + P_f^{p,q}[b, j] · C^{p,q}_{ai,bj}
     end
    end
    for ω_1 = 1..R
     Q^p_{a,ω_1} = Q^p_{a,ω_1} + v_1 · P_h^q[i, ω_1]
    end
   end
  end
 end
end Function
```

Algorithm 4. Calculus of approx Q.

6 Evaluation

To evaluate the presented algorithm we have compared to the algorithm presented by Bonev *et al.* [5]. We consider it is the only one generic enough. The method applies the Graduated Assignment algorithm [8] to compute the N^2 possible probabilistic assignation matrices P_f between the graphs. Next, the $N^2 \cdot R^2$ probabilities $P_f^{p,q}$ are sorted and processed in descending order to compute what they call a Super-graph. The cost of the algorithm is $O(N^2 \cdot (\#iterations \cdot R^4))$.

We evaluate both algorithms using two datasets composed by AGs that represent objects embedded in the plane. In both cases, nodes are defined over a two-dimensional domain that represents its plane position (x, y). Edges have a binary attribute that represents the existence of a line between two terminal points. The former dataset, created synthetically, is composed by 35 classes. The number of graphs per class is $N \in [3, 5, 7, 9, 11]$ and the noise level between graphs is $v \in [10, 20, 40...80]$. Therefore, we defined 5 x 7 = 35 different classes: seven classes with four graphs (with different noise levels), seven classes with five graphs (with different noise levels), and so on. Each class was created as follows. We randomly generate a base graph composed of $R=10$ nodes with 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 N other graphs by: 1, generating Gaussian noise at every node with standard deviation $\sigma = v/100$. 2, removing $v\%$ nodes randomly. 3, inserting $v\%$ nodes (with random attributes) and 4, changing the state of $v\%$ edges. The latter dataset, created at the University of Bern [14], is called Letter. It is composed of 15 classes and 150 graphs per class representing the Roman alphabet i.e. A, E, F, …, X, Y, Z.

From each class, we randomly selected $N \in [3, 5, 7, 9, 11]$ graphs, to generate the CL. To compute the cost C in equations (2), (12) and (14) we used the Edit Distance [15] applied to the sub-graphs induced by $\{v_a^p, v_b^p\}$ and $\{v_i^q, v_j^q\}$. Finally, with the aim of obtaining non-biased results, each experiment was performed 7 times.

The ground truths of our experiments are the MIs in which each isomorphism has been computed through Algorithm 1. Note that these MIs are not restricted to be consistent (3) and so, do not compose a CL (Def. 7). Their costs are computed through C^{MI} (2) and they are supposed to be the lowest ones due to the consistency restriction are not imposed. The results of the evaluation procedure are presented in Fig. 5 for the Letter dataset and in Table 2 for the Synthetic dataset. Each point in Fig. 5 shows the mean cost minus the cost of the ground truth of an experiment set performed by both algorithms. Each set is constructed by 7 random experiments with letter 'A', 7 with letter 'B', … and 7 with letter 'Z' given a concrete size of Γ. Besides, we present the results using the synthetic dataset in Table 1. In this case, each cell of the table represents the percentage of increment of the proposed algorithm in comparison with [5]. Each value is computed using the mean of 7 random experiments using a concrete size of Γ and a concrete noise level.

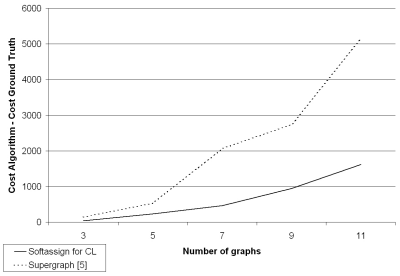

Fig. 5: Results of Letter dataset.

Size of Γ	Noise Level 10	20	40	50	60	70	80
3	0,0	18,9	8,9	3,3	10,0	2,2	15,8
5	0,0	5,8	12,8	10,7	27,7	21,5	18,8
7	0,0	8,3	11,9	16,9	15,7	22,5	15,4
9	0,0	13,2	12,2	12,7	19,9	17,0	16,9
11	0,0	13,2	11,7	17,0	16,2	19,2	18,6

Table 1: Results of Synthetic dataset.

We see in Fig. 5 that the presented algorithm achieve better CLs than [5], we observe that as the size of Γ increases the performance of the presented algorithm tens also to increase respect [5]. In the results performed over the synthetic dataset (Table 1), we can observe that with noises greater than 10, the percentage of increment is considerable and also tends to increase together with the size of Γ and the noise level.

7 Conclusions and further work

Graphs are a very flexible representation of data capable of representing a large sort of problems related to pattern recognition. Examples could be found in image databases, video analysis, biomedical and biological applications and so on. A nice review can be found in [1]. In some of these applications, it is usual the need of finding a structure that represents a set of graphs. This structure is used as a representative of the set. The first step to generate this structure is to find a Common Labeling between the vertices of all the graphs such that a general cost is minimized. It is crucial to find a good common labeling to generate a good representative. In addition some works [2] deduce that, due to the noise, in some applications it is more useful to find a CL (of three graphs) instead of just pairwise labellings.

Known algorithms to compute a Common Labeling consist on first finding the labeling between any pairs of graphs and then combining this information to compute the Common Labeling. The presented algorithm differs from others because it computes the Common Labeling at the same time as the pairwise labelings, mixing the local and global knowledge at each step of the algorithm.

We have compared our algorithm with the most popular one in the literature and we present and evaluation which shows that our method finds better common labelings with similar computational cost. This means that, the approaches that need a Common Labeling between graphs would perform better. Moreover, the proposed iterative approach allows using the current Common Labeling at each step of the algorithm, in comparison with [5] which must wait for all pairwise computations before concluding a solution.

As a future work, we will apply this new technique to the representative of a set of graphs called Structurally-Defined Random Graph [3] and we will analyze its ability to keep the structural and semantic knowledge of the Γ set.

References

[1] Donatello Conte et al..Thirty Years Of Graph Matching In Pattern Recognition. IJPRAI 18(3): 265-298 (2004).

[2] Mark L. Williams and Richard C. Wilson and Edwin R. Hancock. Multiple Graph Matching with Bayesian Inference. PRL, 18(11-13):1275-1281, 1997.

[3] A. Solé-Ribalta & F.Serratosa. A structural and semantic probabilistic model for matching and representing a set of graphs. GBR, LNCS 5534 : 164-173, 2009.

[4] A.K.C. Wong et al.. Entropy and distance of random graphs with application to structural pattern recognition. IEEE TPAMI, 7 : 599-609, 1985.

[5] B. Bonev et. al.. Constellations and the unsupervised learning of graphs. GBRPR, LNCS 4538 : 340-350, 2007.

[6] F. Serratosa et al..Synthesis of function-described graphs and clustering of attributed graph. IJPRAI 16(6):621-655, 2002.

[7] A. Solé-Ribalta & F.Serratosa. On the Computation of the Common Labelling of a set of Attributed Graphs. 14th CIARP, LNCS 5856 : 137-144, 2009.

[8] S. Gold and A. Rangarajan. A Graduated Assignment Algorithm for Graph Matching. IEEE TPAMI, 18(4) : 377 - 388, 1996

[9] W.J. Christmas, J. Kittler and M. Petrou. Structural matching in computer vision using probabilistic relaxation. IEEE TPAMI, 17(8):749-764, 1995.

[10] A. Rosenfeld, R.A. Hummel and S.W. Zucker. Scene labeling by relaxation operators. IEEE Transactions on Systems, Man and Cybernetics, 6:420-443, 1976.

[11] D.P. O'leary and S. Peleg. Analysis of relaxation processes: The two-node label case. IEEE Transactions on Systems, Man and Cybernetics, 13 : 618-623, 1983.

[12] R. Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. The Annals of Mathematical Statistics, 35(2):876-879, 1964.

[13] H. W. Kuhn. The Hungarian method for the assignment problem Export. Naval Research Logistics Quarterly, 2(1-2) : 83-97, 1955.

[14] Riesen, K. and Bunke. IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. SSPR, LNCS 5342 : 287-297, 2009.

[15] A. Sanfeliu and K.S. Fu, "A Distance Measure Between Attributed Relational Graphs for Pattern Recognition", Trans. Systems, Man, and Cybernetics, vol. 13, pp. 353-362, 1983.