Graph Clustering Using the Weighted Minimum Common Supergraph

Horst Bunke[1], P. Foggia[2], C. Guidobaldi[2], and M. Vento[3]

[1] Institut für Informatik und angwandte Mathematik,
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
bunke@iam.unibe.ch
[2] Dipartimento di Informatica e Sistemistica,
Università di Napoli "Federico II", Via Claudio, 21 I-80125 Napoli (Italy)
cguidoba@unina.it
[3] Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica,
Università di Salerno, via P.te Don Melillo, I-84084 Fisciano (SA), Italy
mvento@unisa.it

Abstract. Graphs are a powerful and versatile tool useful for representing patterns in various subfields of science and engineering. In many applications, for example, in pattern recognition and computer vision, it is required to measure the similarity of objects for clustering similar patterns. In this paper a new structural method, the Weighted Minimum Common Supergraph (*WMCS*), for representing a cluster of patterns is proposed. Using this method it becomes easy to extract the common information shared in the patterns of a cluster and separate this information from noise and distortions that usually affect graphs representing real objects. Moreover, experimental results show that *WMCS* is suitable for performing graph clustering.

1 Introduction

Clustering has become a mature discipline, that is important to a number of areas, including pattern recognition, machine learning, computer vision and related fields. There are applications in these fields in which a structural representation of patterns is particularly suitable. In those cases the representation of patterns by attributed graphs (simply graphs, in the following) is extremely convenient and currently very investigated. In unsupervised learning or clustering there is no explicit teacher, and the system forms clusters or *natural groupings* of input patterns, where *natural* is always defined explicitly or implicitly in the clustering system itself [4].

If graphs are used for the representation of structured objects, then grouping similar objects becomes equivalent to find those graphs that are similar to each other in a set of graphs. Even if a number of graph clustering methods are known from the literature [5,7,8,10,11,13,14,15], the investigation on clustering of graphs has only recently started and it is still widely unexplored.

E. Hancock and M. Vento (Eds.): IAPR Workshop GbRPR 2003, LNCS 2726, pp. 235–246, 2003.

In many pattern recognition and artificial vision applications, a cluster is composed of many distorted representations of the same object. In these cases the information contained in the cluster is the representation of the original object itself, while the differences that occur in the distorted objects can be considered as a noise.
When graph clustering is performed and a grouping is achieved, a very interesting open question is how to represent the structural information contained in the cluster. In case the patterns are represented by vectors and hence there is no structural information, the cluster representation problem is very well established and many possibilities have been exploited [4]. Instead relatively new methods for representing a cluster of graphs have been proposed in [8,11,15].
In this paper a new method for representing a cluster of graphs is presented. A cluster of graphs will be described by a graph that is, in general, different from each graph of the given cluster. This new graph, that we call the Weighted Minimum Common Supergraph (*WMCS*) of the cluster, is a graph summarizing all (and only) the properties of all the graphs belonging to the cluster. Moreover weights indicating the occurence of each property in the cluster are added during the construction process, so obtaining a weighted graph. The main advantage of using *WMCS* for representing a cluster of graphs is that the structural information contained in the cluster is preserved and that it is also easy to separate this information from the noise.
The main contribution of this paper is a formal definition of *WMCS* and an approximate procedure for its computation. Moreover two experiments have been conducted. The aim of the first experiment is to evaluate the quality of *WMCS* approximation for representing a cluster of graphs. A second experiment has been conducted to investigate if *WMCS* is suitable to perform graph clustering tasks.
The remainder of the paper is organized as follows. In Section 2, basic terminology and concepts are introduced. Next, in Section 3 a new approach for representing a cluster of graphs is proposed. In Section 4 an approximated method is derived from the proposed approach, while experimental results are reported in Section 5. Finally future work is discussed and some conclusions are drawn in Section 6.

2 Basic Definitions

A *graph isomorphism* is a bijective mapping between the nodes of two graphs that have the same number of nodes, identical labels and identical edge structure. Similarly, a *subgraph isomorphism* between two graphs g_1 and g_2 is an isomorphism between g_1 and a subgraph of g_2.
The *maximum common subgraph* of two graphs g_1 and g_2, $mcs(g_1,g_2)$, is a subgraph of both g_1 and g_2 and has, among all those subgraphs, the maximum number of nodes. The *Minimum Common Supergraph* of two graphs g_1 and g_2, $MCS(g_1,g_2)$, is a supergraph of both g_1 and g_2 and has, among all those supergraphs, the minimum number of nodes. Notice that, both $mcs(g_1,g_2)$ and $MCS(g_1,g_2)$ are not necessarily unique for two given graphs. The *difference* between g_2 and a subgraph g_1, $g_2 - g_1$ is obtained by removing g_1 from g_2, including edges connecting g_1 with the rest of the graph. These

edges are called the *embedding* of g_1 in g_2, $E = emb(g_1,g_2)$. The *union* of g_1 and g_2 including E, $g_1 \cup_E g_2$ is a graph composed from the graphs g_1 and g_2 jointed by edges E. In [3] more details are given on these definitions and a formal proof of the following theorem has been given.

Theorem 2.1: Let g_1 and g_2 be graphs. Then

$$MCS(g_1,g_2) = mcs(g_1,g_2) \cup_{E1} (g_1\text{-}mcs(g_1,g_2)) \cup_{E2} (g_2\text{-}mcs(g_1,g_2)) \quad (1)$$

where

$E_1 = emb(mcs(g_1,g_2),g_1)$ and $E_2 = emb(mcs\ (g_1,g_2),g_2)$

The computation of *mcs* is a NP-compete problem. Several algorithms for *mcs* are known from literature. Theorem 2.1 shows a way how *MCS* can be actually computed for two given graphs g_1 and g_2.

3 Optimal Representation of a Cluster of Graphs

Let G be a set of n graphs. Let us suppose that the graphs of G represent the same object with different distortions, due, for instance, to a not optimal system used to obtain the graph representations from real objects. The graphs of this set are different instances of the same pattern and we can call this set a cluster. The easiest way to represent a cluster is to choose a criterion to decide which element of the cluster is the most representative. Of course the representation of a cluster by means of one of the patterns can cause a loss of information and the inclusion of noise. Thus we want to represent a cluster of graphs using a graph that is in general different from each graph of the cluster. For our purpose, firstly we need to define a *weighted graph*, i.e. a graph with a label on each node (edge) for storing the occurrences of nodes (edges) of the graphs of the cluster.

Def. 3.1: A *weighted graph* is a 6-tuple $g = (V , \lambda , E , \varepsilon , \alpha , \beta)$, where

- V is the finite set of vertices (also called nodes)
- λ: $V \rightarrow N^+$ is a function assigning positive weights (labels) to the nodes
- $E \subseteq V \times V$ is the set of edges
- ε: $E \rightarrow N^+$ is a function assigning positive weights (labels) to the edges
- $\alpha : V \rightarrow L$ is a function assigning attributes to the vertices
- $\beta : E \rightarrow L$ is a function assigning attributes to the edges

In the following, a graph g and a weighted graph g' in which $\lambda(v)=1\ \forall v \in V$ and $\varepsilon(e)=1\ \forall e \in E$, will be considered equivalent.

On the basis of Def 3.1, it is possible to define the Weighted Maximum Common Subgraph of the cluster G, *WMCS*(G). The *WMCS*(G) is a suitable instrument for representing a cluster of graphs. It is the smallest graph containing as subgraph each

graph of the given cluster. As a consequence it is the smallest graph representing each property appearing in the cluster. The *WMCS*(G) is formally defined in Def.3.2.

Def. 3.2: Let $G = \{g_1,\ldots,g_n\}$be a set of graphs; a *Weighted Common Supergraph* of G, *WCS*(G), is a weighted graph $g = (V, \lambda, E, \varepsilon, \alpha, \beta)$ such there exist subgraph isomorphisms from g_i to g $\forall i \in \{1,\ldots,n\}$. We call g a *Weighted Minimum Common Supergraph* of the set of graphs G, *WMCS*(G), if there exists no other weighted common supergraph of G that has fewer nodes than g.
Let $f_i : V_i \rightarrow V$ be an isomorphism between g_i and a subgraph of g. The weight of each node v of g is k if there exist exactly k distinct isomorphisms f_j such that $f_j(v_i) = v$.
Let (u,v) be a pair of nodes of V. The weight of each edge $e = (u,v)$ is h if there exist exactly h distinct isomorphisms $f_j(e_{il}) = (f_j(u_i), f_j(v_l))$ such that $f_j(e_{il}) = e$.

It is worth notice that in general there exists more than an isomorphism $f_i : V_i \rightarrow V$ $\forall i \in \{1,\ldots,n\}$. Thus the *WMCS*($G$) is not uniquely defined. In [16], weights on nodes of attributed graphs are introduced. If the graphs are patterns of a cluster, then these weights represent the occurrence of a node (edge) in the cluster. According to Def.3.2 the weight of a node (edge) of *WMCS*(G) is k iff that node (edge) is contained exactly in k different graphs of the cluster; thus k is always included between 1 and n.

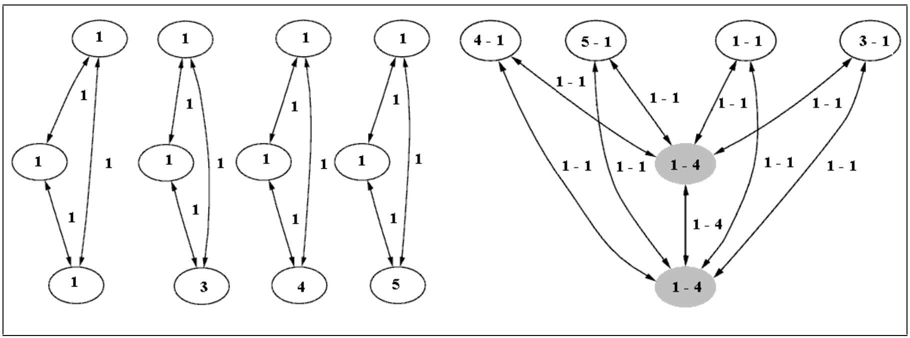

Fig. 3.1. A set of graphs G and a *WMCS*(G). On each node (edge) of *WMCS*(G) there is a second label for the weight. This weight is the occurrence of the node (edge) in the cluster.

A very interesting problem is to extract from the graphs of the cluster the properties belonging to the original object and to distinguish them from noise introduced by the system. A property of a pattern represented by a graph can be represented by a subgraph of that graph. In the rest of this paragraph *subgraph* and *property* will be used as synonymous. A possibility to select the more representative properties in a cluster of graphs G, i.e. properties appearing more than p times, is to select nodes (edges) of *WMCS*(G) having a weight greater than p. This is formally obtained in Def .3.3.

Def. 3.3: Let g be a *WMCS*(G). A *WMCS*$_p$(G), is a subgraph of g containing all the nodes of g whose weight is at least p and all edges of g having weight at least p, that connect those nodes in V having weight at least p.

The *WMCS*(*G*) is a suitable instrument for finding the common properties of a cluster of graphs for several reasons:

- Nodes (edges) of those subgraphs that are present in many different graphs of the cluster, i.e. the information of the cluster, will correspond to nodes and edges of the *WMCS*(*G*) having large weights. Conversely nodes (edges) of subgraphs that are present in a few graphs of the cluster, i.e. the noise of the cluster, will correspond to nodes and edges of the *WMCS*(*G*) having small weights.
- A possibility to distinguish the information from the noise is the computation of the $WMCS_p(G)$, where p is a threshold of noise-rejection. The selection of an optimal threshold p is useful to restore the real information contained in the cluster.
- The construction of the $WMCS_p(G)$ can be viewed as a process of generalization. Indeed only those subgraphs present at least in p graphs of the given cluster, are also present in $WMCS_p(G)$. The $WMCS_p(G)$ summarizes all and only those properties appearing at least in p different graphs of the cluster. $WMCS_p(G)$ with high value of p may be considered as some kind of mean graph; conversely, if we select all the vertices and edges which have a value lower than a given threshold can such a graph be interpreted as the variance of the cluster.
- The $WMCS_p(G)$ can be used to build an expert system. Indeed a $WMCS_p(G)$ not only represents the common properties found in a cluster of graphs, but these properties can be easily displayed and interpreted by an expert of the underlying domain. In other words, the knowledge of the system is not hidden, but is easily interpretable by an expert.

4 WMCS Approximation

The proposed definition of *WMCS* is expensive to compute. Indeed the computation of a *WMCS* on a pair of graph is an NP-complete problem, that is exponential in the number of nodes of one of the two given graphs. Moreover the complexity of computing a *WMCS* of a set of graphs is exponential in the number of graphs. Consequently we propose a procedure to approximately compute a *WMCS* of a set of graphs *G*.

Firstly, it is necessary to extend the definition of the *WMCS* to introduce the *WMCS* of a pair of weighted graphs. Similarly to the *MCS* of a pair of graphs, that can be derived from the *mcs*, also the *WMCS* of a pair of weighed graphs can be derived from the *wmcs*, i.e. the *weighted maximum common subgraph*. Both *wmcs* and *WMCS* of a pair of weighted graphs are defined below.

Def. 4.1: Let $g_1 = (V_1, \lambda_1, E_1, \varepsilon_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, \lambda_2, E_2, \varepsilon_2, \alpha_2, \beta_2)$ be two weighted graphs. A *weighted common subgraph* of g_1 and g_2, $wcs(g_1, g_2)$, is a weighted graph $g=(V,\lambda,E,\varepsilon,\alpha,\beta)$ such there exist subgraph isomorphisms from g to g_1 and from g to g_2. We call g a *weighted maximum common subgraph* of g_1 and g_2, $wmcs(g_1, g_2)$, if there exists no other common subgraph of g_1 and g_2 that has more nodes than g. Let m be the size of $wcs(g_1,g_2)$, and let $V=\{v_1,\ldots,v_m\}$ the set of nodes of g. Furthermore let $V_i^{'}=$

$\{v'_{i1},\ldots,v'_{im}\} \subseteq V$, i ∈ {1,2} be the subsets of nodes corresponding to $wcs(g_1,g_2)$. There exist subgraphs isomorphisms $f_i(v_j):V\rightarrow V_i^{'}$, $\forall i\in\{1,2\},\forall j\in\{1,\ldots,m\}$. The weight of each node v_j is $\lambda(v_j) = \lambda_1(f_1(v_j)) + \lambda_2(f_2(v_j))$. Let $f_i(e_{jk}) = (f_i(v_j), f_i(v_k))$, $i\in\{1,2\}$ and $\forall j$, $\forall k \in\{1,\ldots,m\}$. The weight of each edge e_{jk} is $\varepsilon(e_{jk})=\varepsilon_1(f_1(e_{jk}))+ \varepsilon_2(f_2(e_{jk}))$.

The computation of $wmcs(g_1,g_2)$, is based on for the computation of the *mcs* of two graphs. For this aim a version of McGregor algorithm [12] has been used.

Def. 4.2: Let g_1 and g_2 be weighted graphs. A *Weighted Common Supergraph* of g_1 and g_2, $WCS(g_1,g_2)$, is a weighted graph g such that there exist subgraph isomorphisms from g_1 to g and from g_2 to g. We call g a *Weighted Minimum Common Supergraph* of g_1 and g_2, $WMCS(g_1,g_2)$, if there exists no other weighted common supergraph of g_1 and g_2 that has less nodes than g.

Theorem 2.1 can be also extended in case of weighted graphs, obtaining a procedure for constructing $WMCS(g_1,g_2)$. The weights of $WMCS(g_1,g_2)$ are defined according to the construction procedure of Fig 4.1.

```
input: g1,g2; output: WMCS(g1,g2);
procedure WMCS(g1,g2)
begin
   E1 = emb(wmcs(g1,g2),g1); E2 = emb(wmcs(g1,g2),g2)
   WMCS(g1,g2) =
      wmcs(g1,g2)∪E1(g1-wmcs(g1,g2))∪E2(g2-wmcs(g1,g2));
end procedure
```

Fig. 4.1. Construction procedure for the weights of $WMCS(g_1,g_2)$.

Firstly, each graph of the set G is transformed in a weighted graph, assigning to each node (edge) the weight 1.Then an ordering θ is chosen for the n graphs of the set G: $\theta = ord(G)$. After this step the procedure $WMCS(g_1,g_2)$ is iterated n-1 times for computing an approximated *WMCS* (G). This procedure is shown in Fig 4.2.

```
procedure WMCS(θ)
input: g1,…,gn,θ output: WMCS(θ);
begin
   assign to each node and edge the weight 1;
   order the graphs according to θ;
   W = g1 ;
   for i = 2 to n
   begin
      W = WMCS(W,gi);
   end for
   WMCS(θ)= W;
end procedure
```

Fig. 4.2. Linear approximation of $WMCS(G)$

The main advantage of this approximate procedure is that the complexity of the computation becomes linear in the number of graphs. Using this procedure, the built graph is surely a common supergraph, but the main disadvantage is that in most cases it is not minimal, i.e. it will usually include some extra nodes. It is also possible that the weight of some node has not anymore the proper value but it becomes smaller. Indeed, it can happen that a node already present in the graph under construction is not recognized and thus it is inserted once more. The weight is updated, for each iteration of the procedure, on only one of the two copies of the node, thus it is not guaranteed that at the end of the procedure the weight will be correct. More in details, it is only ensured that the sum of the weights of all the copies of a node will be equal to the correct value. In general some nodes could have a weight smaller than the real occurrence of the property in the cluster. As a consequence, the capability of the approximated *WMCS* to learn from examples is reduced, since some more node is introduced, and the occurence of the some real property is not always correctly represented, but could be underestimated. This phenomenon can be interpreted as a kind of hypergeneralization.

A possibility to reduce the error generated by the proposed procedure, is to perform a random search algorithm. The first step is to realize a number m of different randomly generated orderings $\theta_1,\ldots,\theta_m$ of the graphs of the cluster (*shuffles*). For each shuffle θ_j, the $WMCS(\theta j)$ have to be evaluated and then the best approximation is chosen.

The quality of the approximation is measured through an entropy function $E(WMCS(\theta j))$:

$$E = -\sum_{i=1}^{t} \frac{w_i}{n} \cdot \log\left(\frac{w_i}{n}\right) - \sum_{i=1}^{t}\sum_{j=1}^{t} \frac{w_{ij}}{n} \cdot \log\left(\frac{w_{ij}}{n}\right) \tag{2}$$

where n is the number of graphs of the cluster, t is the number of nodes of the graph $WMCS(\theta j)$, w_i is the weight of the node i and w_{ij} is the weight of the edge (i,j). Each weight is associated to the occurrences of the node (edge) in the graph, thus the ratio w_i/n (w_{ij}/n) can be interpreted as a probability. The minimum of E is obtained if the cluster is composed of all isomorphic graphs, while the maximum is obtained if there is no overlap between any pair of graphs of the cluster. The $WMCS(\theta j)$ minimizing the entropy will be assumed as the best $WMCS(\theta j)$.

5 Experimental Results

The aim of the first experiment, is the evaluation of the quality of the *WMCS* approximation for representing the cluster G. For this aim, let us suppose that the graphs of a cluster G represent the same pattern with different distortions, due, for instance, to a not optimal system used to obtain the graph representations from real objects.

Firstly an attributed graph that we call *seed* is generated. This graph is a *mesh* [2,9] in which the number of nodes is chosen and the size of the alphabet of attributes is fixed. The method for generating this graph is explicated in details in [1]. A degree of distortion is added to the seed, using a graph distortion algorithm, obtaining a second graph. This operation is repeated k times, thus a cluster of k distorted graphs is obtained. These k graphs are the elements of the cluster G. The seed is not included in

the cluster, but it is only used in the generation process. In our experiments a distortion is defined as the substitution of an attribute of a node (edge) with another attribute randomly chosen from the same alphabet, the considered alphabet has $M = 50$ different symbols. For each node (edge) of the seed a probability p of distortion is defined. For instance, if $p = 0.3$, this means that the probability that the attribute of each node (edge) of the seed is changed is 0.3. Experiments are repeated with 1, 10 and 100 shuffles.

In Fig.5.1 the size of the approximated *WMCS* is shown when the distortion increase. Obviously, when the graph distortion increases, also the *WMCS* size increases, due the smaller overlap of the graphs in the cluster. It is worth noting that there is no significant improvement when the number of shuffles of the graphs is increased. Moreover, it is interesting to notice, in Fig.5.2, that the maximum weight of the approximated *WMCS*, is increasing when the number of shuffles increases form 1 to 10, thus a better approximation is obtained, but only a small further improvement is achieved when the number of shuffles becomes 100. From our experiments, it follows that a good approximation of the *WMCS* can be obtained with a small number of shuffles, and further shuffles do not cause very significant variations. We assume that using shuffles, the algorithm can avoid most of the local maxima; the fact that increasing the number of shuffles there is only a small improvement in the approximation, can be an indication that the approximate WMCS is not too far from the exact one. Finally it should be pointed out that, when there is a significant overlap in the graphs of the cluster, the *WMCS* is very compact.

A second experiment has been conducted to investigate if *WMCS* is suitable to perform graph clustering tasks. In this experiment, patterns are graphs extracted from the pictures obtained using the plex grammar tool [5] (see Fig.5.3).

Each component of the picture is represented using a node. If a picture represents, for instance, a man, then there will be a node representing the hat, another node representing the head and so on. On each node one attribute is used: the area of the bounding box of the component. The area is normalized between 0 and 99 and two nodes are assumed to have the same area, if the difference between the two areas is smaller than 5. Nodes are connected through edges. Each edge has two attributes: sine and cosine of the angle between the segment connecting the centers of the two components connected by the edge and the horizontal line. Both the two attributes are normalized between 0 and 19 and two attributes are assumed to have the same value if their difference is smaller than 2. It is interesting to notice that only a part of the information of a picture is stored in the graph: for instance no information on the colors and on the shapes of the components are taken into account. The classification is obtained only using areas of the components and their relative positions.

Three different classes of patterns are considered: men, ships and houses. For each class, 10 patterns are considered and the *WMCS* is computed, using 100 shuffles.

What we expect is that the entropy of the *WMCS* has a small variation if a new pattern, similar to the patterns of the class, is added to the cluster. Indeed, in this case, weights of those nodes (edges) with larger frequencies will be incremented. Conversely if a pattern very dissimilar from the patterns of the class is added to the cluster, we expect a larger entropy variation, because nodes (edges) with small frequencies will be added

to the *WMCS*. In Tab.5.1 results are summarized. Each column represents the entropy variation ΔE of a *WMCS* when a new pattern (i.e. different from the 10 patterns used to built the *WMCS*) is added to a cluster. It is noteworthy that, for each row, the minimum ΔE is obtained when a new pattern of the same class of the *WMCS* is added to the cluster. This result confirms that *WMCS* can be used to distinguish patterns of a given class from patterns of other classes. Results shown in Tab.5.1 are in average: the insertion of a new pattern in a cluster has been repeated 100 times for each category.
In Tab.5.2 the classification results (in percentage) are shown. Each row summarizes the classification of patterns belonging to a class, i.e. the second row represents the classification results for the houses. Results are obtained using a nearest-neighbour criterion, i.e. firstly a pattern is assigned to each *WMCS*, then the entropy variation ΔE is evaluated and finally the pattern is assigned to the class minimizing ΔE. Non-diagonal elements represent misclassifications.

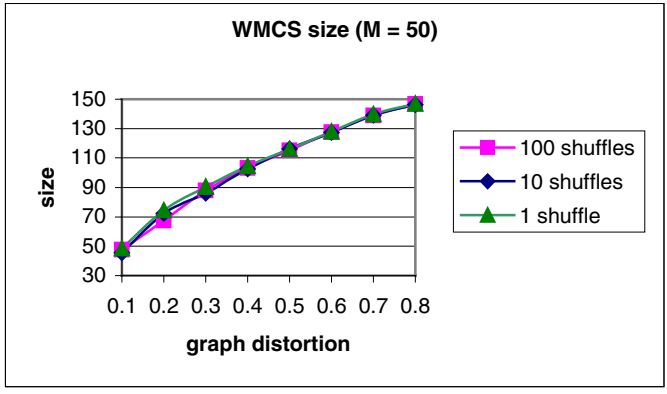

Fig. 5.1. The size of the approximated *WMCS* increase for an increasing distortion of the graphs. It is worth noting that a very small improvement can be obtained raising the number of shuffles of the graphs. Results are in average, each computation has been repeated 100 times.

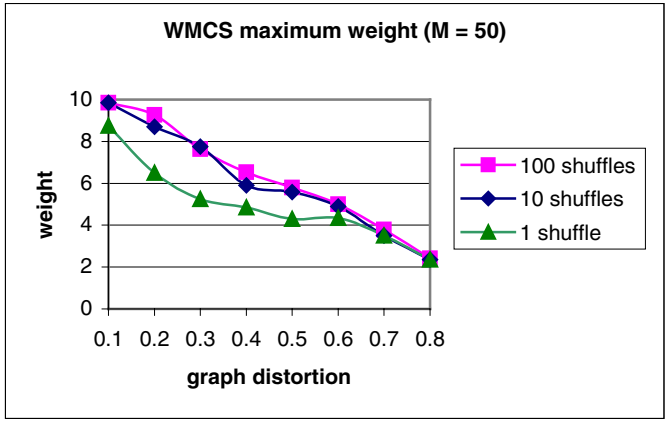

Fig. 5.2. When the distortion of the graph increases, then the maximum weight of the approximated *WMCS* decrease, due a smaller overlap of the graphs. Moreover, if more then 10 shuffles are produced, than only a small improvement can be obtained in the approximation. Results are in average, each computation has been repeated 100 times.

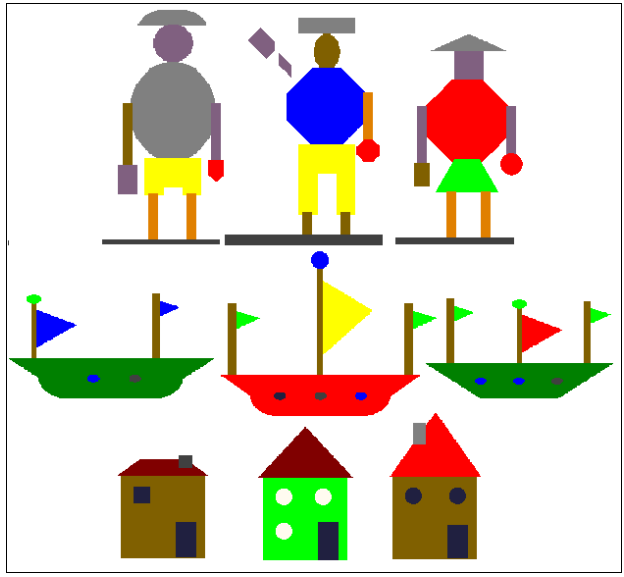

Fig. 5.3. Examples of men, ships and houses obtained using the plex grammar tool.

Table 5.1. Each column represents the entropy variation ΔE of a *WMCS* when a pattern is added to a cluster. Lower values of ΔE are obtained if a ship is added to the cluster of ships, or if an house is added to the cluster of houses or if a man is added to the cluster of men.

ENTROPY VARIATION

		Element		
		ship	**house**	**man**
Cluster	**ship**	**0.9922**	1.913	3.6501
	house	3.6129	**0.2275**	4.3503
	man	2.4141	2.4310	**1.1613**

Table 5.2. In each row the classification percentages of the patterns are shown. Bold values are the correct classifications.

CLASSIFICATION RATES

		Class		
		ship	**house**	**man**
Element	**ship**	**90**	8	2
	house	0	**100**	0
	man	0	0	**100**

It is worth notice that for *houses* and *men* there is no misclassification, even if no information is used for describing shapes and colours in the classification system. The worst case is for ships: 8% of ships are classified as houses.

6 Conclusions and Perspectives

Clustering has become a mature discipline that is important to a number of areas, including pattern recognition and related fields. But the clustering of graphs is still widely unexplored. In this paper a new method for representing a cluster of graph, the *WMCS*, has been described. The main advantage of using *WMCS* for representing a cluster is that the structural information contained into the cluster is preserved and that is also easy to separate this information from the noise. The main disadvantage is the high computational complexity of the method, thus an approximation is proposed. The accuracy of the proposed approximation is dependent on the chosen number of orderings of the elements of the cluster. Experiments have been realized to describe the behavior of *WMCS* when the number of orderings is changed and to describe the behavior of *WMCS* when the noise of the cluster is increasing. Preliminary test shows that a good *WMCS* approximation can be obtained with a small number of orderings of the set of graphs. Moreover, an entropy function $E(WMCS)$ has been defined and the variation ΔE has been used to perform graph clustering on a three classes problem. Preliminary tests show that *WMCS* is a suitable approach for performing graph clustering even if the representation is rough and doesn't contain all the information of the original patterns.

References

[1] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, M. Vento: A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs SSPR/SPR2002, pp 123–132. 2002.

[2] H. Bunke, M. Vento, Benchmarking of Graph Matching Algorithms, Proc. 2nd Workshop on Graph-based Representations, Haindorf, pp. 109–114, 1999.

[3] H. Bunke, X. Jiang, A. Kandel, On the Minimum Common Supergraph of two Graphs, Computing 65, No. 1, pp. 13–25, 2000.

[4] R.O. Duda, P.E. Hart, D.G. Stork. Pattern Classification, II Ed., Wiley-Interscience Publication. 2000.

[5] M. Hagenbuchner, A.C. Tsoi, The Traffic Policeman Benchmark. European Symposium on Artificial Neural Networks, Michel Verleysen, D-Facto, pp. 63–68, April, 1999.

[6] C. De Mauro, M. Diligenti, M. Gori, M. Maggini, Similarity Learning for Graph-Based Image Representations. Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, pp. 250–259. 2001.

[7] R. Englert, R. Glantz, Towards the Clustering of Graphs. Proceedings of the 2nd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, pp. 125–133. 1999.

[8] S. Günter, H. Bunke, Self-organizing Map for Clustering in the Graph Domain, Pattern Recognition Letters 23, pp. 401–417. Accepted for Publication.

[9] P. Foggia, C. Sansone, M. Vento, A Database of Graphs for Isomorphism and Sub-Graph Isomorphism Benchmarking, Proc. of the 3rd IAPR TC-15 International Workshop on Graph-based Representations, Italy, pp. 176–187, 2001.

[10] B. Luo, A. Robles-Kelly, A. Torsello, R.C. Wilson, E. R. Hancock, Clustering Shock Trees. Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, pp. 217–228. 2001.

[11] B. Luo, R.C. Wilson, E. R. Hancock, Spectral Feature Vectors for Graph Clustering. Proceeding of the Joint IAPR International Workshops SSPR and SPR, pp. 83–93. 2002.

[12] J.J. McGregor, Backtrack Search Algorithms and the Maximal Common Subgraph Problem, Software Practice and Experience, Vol. 12, pp. 23–34, 1982.

[13] D. Seong, H. S. Kim, K.H. Park, Incremental clustering of attributed graphs. IEEE Transactions on Systems, Man and Cybernetics. Vol. 23 - 5, pp. 1399–1411. 1993.

[14] A. Sanfeliu, F. Serratosa, R. Alquezar, Clustering of attributed graphs and unsupervised synthesis of function-described graphs. Proceedings of 15th International Conference on Pattern Recognition, pp. 1022–1025. 2000.

[15] A. Sanfeliu, F. Serratosa, R. Alquezar, Synthesis of Function-Described Graphs and Clustering of Attributed Graphs. International Journal of Pattern Recognition and Artificial-Intelligence, Vol.16, No.6, pp. 621–655. 2002.

[16] A.K.C. Wong, J. Constant, M.L. You, Random Graphs, in Syntactic and Structural Pattern Recognition - Theory and Applications 4:197–234, H. Bunke and A. Sanfeliu (Eds.), World Sci. Pub. Co., 1990.