Median Graphs: A Genetic Approach Based on New Theoretical Properties

M. Ferrer [a,*], E. Valveny [a], F. Serratosa [b]

[a] *Centre de Visió per Computador, Departament de Ciències de la Computació. Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain*

[b] *Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, 43007 Tarragona, Spain*

Abstract

Given a set of graphs, the median graph has been theoretically presented as a useful concept to infer a representative of the set. However, the computation of the median graph is a highly complex task and its practical application has been very limited up to now. In this work we present two major contributions. On one side, and from a theoretical point of view, we show new theoretical properties of the median graph. On the other side, using these new properties, we present a new approximate algorithm based on the genetic search, that improves the computation of the median graph. Finally, we perform a set of experiments on real data, where none of the existing algorithms for the median graph computation could be applied up to now due to their computational complexity. With these results, we show how the concept of the median graph can be used in real applications and leaves the box of the only-theoretical concepts, demonstrating, from a practical point of view, that can be a useful tool to represent a set of graphs.

Key words: Median Graph, Genetic Search, Maximum Common Subgraph, Graph Matching, Structural Pattern Recognition

1 Introduction

In structural pattern recognition, the concept of median graph [22] has been presented as a useful tool to represent a set of graphs. Given a set of graphs S, the median graph is defined as the graph that minimizes the sum of distances

* Corresponding author: Tel.: +34 93 581 23 01 / Fax.: +34 93 581 16 70
Email address: mferrer@cvc.uab.cat (M. Ferrer).

(SOD) to all the graphs in S. It aims to extract the essential information of a set of graphs into a single prototype. Potential applications of median graphs include graph clustering and prototype learning. For instance, it has been successfully applied to different areas such as the synthesis of graphical symbols [21], image clustering [20], optical character recognition [22] and graphical symbol recognition [16].

Nevertheless, the computation of the generalized median graph is a highly complex task. In the past some exact and approximate algorithms have been developed. Optimal algorithms include a tree search approach called multimatch [27] and a more efficient algorithm which takes advantage of certain conditions about the distance between two graphs [18] . Suboptimal methods include genetic algorithms [8,22], greedy-based algorithms [20,19] and spectral-based approaches such that of [14] and [35]. In spite of this wide offer of algorithmic tools, all of them are very limited in their application. They are often restricted to use small graphs and with some particular conditions. None of them have been applied using real data.

In spite of these efforts to develop new and more efficient algorithms, only few work about the theoretical properties of the median graph exists. In [22], some interesting properties of the median graph related to their size and their SOD have been presented. Concretely, they show the general limits for both the size and the SOD of the median graph. Unfortunately, these original bounds are sometimes very coarse and they can not be easily used to reduce the complexity of its computation. Thus, the reduction of such bounds may be crucial to be able to compute the median graph more efficiently or to obtain better approximations.

In this paper we make theoretical and algorithmic contributions to the computation of the median graph that result in a new genetic algorithm, computationally more efficient than existing approaches, that can be applied to real sets of data with large graphs. The most important contribution of this work is that, from a theoretical point of view, we show that under a particular cost function and a distance based on the maximum common subgraph, the original bounds given in [22] can be reduced [1] . After that, we use these new theoretical results to present the second major contribution of this paper: a new approximate algorithm for the median graph computation based on a genetic search. It validates the new bounds not only from a theoretical point of view, but also giving them a practical application, implementing a new strategy for the median graph computation. As a result, the computation time of the median graph is reduced. In order to show the usefulness of the new approach, we perform a set of preliminary experiments using a real database of 2,340 webpages, split into 6 classes. Each webpage is represented as a graph

[1] Preliminary versions of such theoretical proofs have appeared in [15] and [17].

with a number of nodes between 100 and 300. In a first experiment we show how the median graph can be computed in a reasonable time, compared with the previous existing algorithms. Furthermore, in a second experiment we assess the accuracy of the median comparing its SOD with the SOD of the set median graph. We show that with this new approach we obtain graphs with lower SOD than the set median, which demonstrates that we are obtaining good approximations of the median graph. Finally, although it is not the main objective of this work, we try to validate the median graph as a representative of a class of graphs. Up to now, existing algorithms could only be applied to very limited sets of graphs and the median graph could not be evaluated from a practical point of view as a good representative of a class. To that extent, we perform a preliminary classification experiment using the median graph. In some cases, we obtain slightly better results than a nearest-neighbor classifier with a much lower computation time. In this way, we demonstrate, for the first time, that the median graph can be a feasible alternative to represent a set of graphs in real applications.

The rest of the paper is organized as follows. In Section 2, we present some theoretical concepts required to understand the rest of this work. Then, in Section 3 the concept of the median graph and its theoretical properties are introduced. After that, in Section 4, we present the new theoretical properties of the median graph. Section 5 introduces a new genetic algorithm for the median graph computation, that takes advantage of the new theoretical results. Then, Section 6 is devoted to present our experiments and the results we obtained. Finally, we terminate with some conclusions and possible future research lines.

2 Definitions and notation

2.1 Basic definitions

Let L be a finite alphabet of labels for nodes and edges. A **graph** is a four-tuple $g = (V, E, \alpha, \beta)$ where V is the finite set of nodes, E is the set of edges, α is the node labelling function ($\alpha : V \longrightarrow L$), and β is the edge labelling function ($\beta : E \longrightarrow L$). We assume that our graphs are fully connected. Consequently, the set of *edges* is implicitly given (i.e. $E = V \times V$). Such assumption is only for notational convenience, and it does not impose any restriction in the generality of our results. In the case where no edge exists between two given nodes, we can include the special null label ε in the set of labels L to model such situation. If $V = \phi$, then g is called the *empty graph*. Finally, the number of nodes of a graph g is denoted by $|g|$.

Given two graphs, $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$, g_2 is a **subgraph** of g_1, denoted by $g_2 \subseteq g_1$ if, $V_2 \subseteq V_1$, $\alpha_2(v) = \alpha_1(v)$ for all $v \in V_2$ and $\beta_2((u, v)) = \beta_1((u, v))$ for all $(u, v) \in V_2 \times V_1$.

From this definition, it follows that, given a graph $g_1 = (V_1, E_1, \alpha_1, \beta_1)$, a subset $V_2 \subseteq V_1$ of its vertices uniquely defines a subgraph. Such subgraph is called the subgraph *induced* by V_2.

In order to check whether two graphs are identical or not, we use the **graph isomorphism**. Given two graphs $g_1 = (V_1, E_1, \alpha_1, \beta_1)$, and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$, a **graph isomorphism** between g_1 and g_2 is a bijective mapping $f : V_1 \longrightarrow V_2$ such that, $\alpha_1(x) = \alpha_2(f(x))$ for all $x \in V_1$ and $\beta_1((x, y)) = \beta_2((f(x), f(y)))$ for all $(x, y) \in V_1 \times V_1$. Two graphs, g_1 and g_2, are isomorphic if there exists a graph isomorphism between them.

Related to graph isomorphism there is the concept of subgraph isomorphism. Given two graphs $g_1 = (V_1, E_1, \alpha_1, \beta_1)$, and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ an injective function $f : V_1 \longrightarrow V_2$ is called a **subgraph isomorphism** from g_1 to g_2 if there exists a subgraph $g \subseteq g_2$, such that f is a graph isomorphism between g_1 and g.

2.2 Maximum Common Subgraph and Minimum Common Supergraph of a Set of Graphs

Let $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ be two graphs. A graph g is called a **common subgraph** (cs) of g_1 and g_2 if there exists a subgraph isomorphism from g to g_1 and from g to g_2. A common subgraph of g_1 and g_2 is called **maximum common subgraph** (mcs) if there exists no other common subgraph of g_1 and g_2 with more nodes than g. We will also denote the $mcs(g_1, g_2)$ by g_m.

Intuitively, it is the largest part of them that is identical in terms of structure and labels. It is clear that the more similar two graphs are the largest their maximum common subgraph is. In the last years some papers have been presented related to the computation of maximum common subgraph [2,12,25,33] based on different approaches and algorithms. An explanation of such methods and a comparison between them can be found in [5,10].

A graph g is called **common supergraph** (CS) of g_1 and g_2 if there exists a subgraph isomorphism from g_1 to g and from g_2 to g. A common supergraph of g_1 and g_2 is called **minimum common supergraph** (MCS) if there exists no other common supergraph of g_1 and g_2 having less nodes than g. We will also denote the $MCS(g_1, g_2)$ by g_M.

The minimum common supergraph of two graphs can be seen as the graph with the minimum structure such that both graphs are contained in it as subgraphs. The computation of the minimum common supergraph can be reduced to the computation of the maximum common subgraph [7].

Let us now generalize such definitions to a set of n graphs. Let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs. A graph $g_m(S)$ is called a **maximum common subgraph of S** if $g_m(S)$ is a common subgraph of $\{g_1, g_2, \cdots, g_n\}$ and there is no other common subgraph of $\{g_1, g_2, \cdots, g_n\}$ having more nodes than $g_m(S)$. In addition, a graph $g_M(S)$ is called a **minimum common supergraph of S** if $\{g_1, g_2, \cdots, g_n\}$ are subgraphs of $g_M(S)$ and there is no other common supergraph of $\{g_1, g_2, \cdots, g_n\}$ having less nodes than $g_M(S)$. We will also denote the $g_m(S)$ and the $g_M(S)$ as $mcs(S)$ and $MCS(S)$ respectively.

The computation of these graphs is still an open question. Approximate algorithms for their computation have been given in [6].

2.3 Graph Distance

In real applications, the graph-based representations of the same object may be different. In this case we need a measure of the dissimilarity between two given graphs instead of simply knowing whether they are identical, as graph isomorphism does. One of the methods most widely used to compute the dissimilarity between two graphs is the **graph edit distance** [4,31]. The main advantage over other graph matching methods is that graph edit distance can be applied to arbitrary graphs with any type of node and edge labels. The basic idea behind the graph edit distance is to define a dissimilarity measure between two graphs by the minimum amount of distortion required to transform one graph into the other [4]. To this end, a number of distortion or edit operations e, consisting of the insertion, deletion and substitution of both nodes and edges needs to be defined. Then, for every pair of graphs (g_1 and g_2), there exists a sequence of edit operations, or edit path $p(g_1, g_2) = (e_1, \ldots, e_k)$ (where each e_i denotes an edit operation) that transforms one graph into the other. In general, several edit paths exist between two given graphs. This set of edit paths is denoted by $\wp(g_1, g_2)$. To quantitatively evaluate which edit path is the best, edit cost functions are introduced. The basic idea is to assign a penalty cost c to each edit operation according to the amount of distortion it introduces in the transformation. The edit distance d between two graphs g_1 and g_2 denoted by $d(g_1, g_2)$ is the cost of the edit path with minimum cost that transforms one graph into the other.

2.4 Graph edit distance and the Maximum Common Subgraph

Several exact and approximate algorithms have been presented in the past related to the computation of the distance between two graphs [4,28–30]. Some of them are related to the maximum common subgraph [3,9,13] – an excellent review of such distances is [32] –. In this work, we will use one of theses distances based on the maximum common subgraph given in [3] taking advantage of a particular cost function where the cost of node deletion and insertion is always 1, the cost of edge deletion and insertion is always 0 and the cost of node and edge substitution takes the values 0 or ∞ depending on whether the substitution is identical or not, respectively. In [3] they show that, using this cost function, the edit distance between two graphs can be expressed as:

$$d(g_1, g_2) = |g_1| + |g_2| - 2\,|mcs(g_1, g_2)| = |g_1| + |g_2| - 2\,|g_m| \tag{1}$$

This result demonstrates its validity and applicability as it states the intuitive idea that the more two graphs have in common, the lower their distance is. In the rest of the paper we will assume that the distance between two graphs is computed according to Equation (1).

3 Generalized Median Graph

Given a set of graphs, the concept of median graph has been presented as a useful tool to compute a representative of the set. Let U be the set of graphs that can be constructed using labels from L. Given $S = \{g_1, g_2, ..., g_n\} \subseteq U$, the **generalized median graph** $\bar{g}$ of S is defined as the graph $g \in U$ such that its sum of distances (SOD) to all the graphs in S is minimum:

$$\bar{g} = arg\ \min_{g \in U} \sum_{g_i \in S} d(g, g_i) = arg\ \min_{g \in U}\ SOD(g) \tag{2}$$

Notice that $\bar{g}$ is not usually a member of S and, in general, more than one generalized median graph can be found for a given set S.

The computation of the generalized median graph can only be done in exponential time, both in the number of graphs in S and their size [22]. Some exact and approximate algorithms for the median graph computation have been developed so far. Optimal strategies include a tree search approach called multimatch [27] and a more efficient algorithm [18] which takes advantage of certain conditions of the cost function. Sub-optimal approaches include ge-

netic algorithms [22,27], greedy-based algorithms [20], and algorithms based on the spectral graph theory [14,35].

Another alternative to represent a set of graphs is to use the set median graph (usually denoted by $\hat{g}$) instead of the generalized median graph. The difference between them is only the search space where the median is looked for. While the search space for $\bar{g}$ is U, that is, the whole universe of graphs, the search space for $\hat{g}$ is simply S, that is, the set of graphs in the learning set. The set median graph is usually not the best representative of a set of graphs, but it is often a good starting point towards the search of the generalized median graph [2].

3.1 Theoretical Properties of the Median Graphs

Beyond the algorithmic solutions for the median graph computation there are some interesting theoretical properties related to the median graph. Such properties include the bounds on the size of the median graph and the bounds on the SOD. Both properties originally appeared in [22]. For the sake of completeness, they are presented in the following sections.

3.1.1 Bounds on the Size of the Median Graph

In [22] it is shown that the minimum and maximum number of nodes of the median graph is between the following limits:

$$0 \leq |\bar{g}| \leq \sum_{i=1}^{n} |g_i| \tag{3}$$

That is, it states that the size of the median graph has to be greater or equal than 0 and less or equal than the sum of nodes of all the graphs in S. The proof of such bounds can be found in [22].

3.1.2 Bounds on the SOD of the Median Graph

The bounds for the SOD of the median graph are slightly more difficult to derive. For the upper bound, it is assumed in [22] that the empty graph g_e and the union graph g_u are meaningful candidates for the median graph. Then, the upper limit for the SOD of the median graph is:

[2] Unless explicitly mentioned, from now on the term median graph will refer to the generalized median graph.

$$SOD(\bar{g}) \leq min\{SOD(g_e), SOD(g_u)\} \tag{4}$$

The lower bound is out of the scope of this work. The reader is referred to [22] for more details.

4 New Theoretical Results on the Median Graph

The theoretical properties mentioned above can be used to bound the search space of the median, either by limiting the size of the candidate medians or discarding some of these candidate medians based on the bounds of the SOD, for instance. Nevertheless, as mentioned in [22], these bounds are sometimes too coarse and may not be very useful to reduce the complexity of the median graph computation. Using the concepts of $mcs(S)$ and $MCS(S)$ defined in section 2.2, the cost function introduced in section 2.4 and the distance measure defined in expression (1), we will prove in this section that it is possible to reduce the limits on the size of the median graph given in expression (3) (Section 4.1), and the upper bound for the SOD (Section 4.2).

4.1 Reduction of the Bounds on the Size of the Median Graph

Theorem 1: Let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs and $\bar{g}$ a possible median graph of S. Under the cost function and the distance measure given in section 2.4, the number of nodes of $\bar{g}$ is in the limits,

$$0 \leq |g_m(S)| \leq |\bar{g}| \leq |g_M(S)| \leq \sum_{i=1}^{n} |g_i| \tag{5}$$

Proof: To demonstrate the first part of the equation (5) (i.e. $|g_m(S)| \leq |\bar{g}|$), suppose that $|\bar{g}| < |g_m(S)|$. If we compute the term $SOD(g_m(S))$, we will arrive to the next expression:

$$SOD(g_m(S)) = \sum_{i=1}^{n} d(g_i, g_m(S)) = \sum_{i=1}^{n} (|g_i| + |g_m(S)| - 2|g_m(S)|) = \sum_{i=1}^{n} |g_i| - n|g_m(S)| \tag{6}$$

Notice that $g_m(S)$ is the maximum common subgraph of S and, then, it is a subgraph of any graph g_i in S. Therefore, if we compute $d(g_i, g_m(S))$ using expression (1) the term $|g_m|$ is exactly $|g_m(S)|$.

For the computation of $SOD(\bar{g})$ we will follow a similar reasoning. Assuming that $|\bar{g}| < |g_m(S)|$, we can determine the minimum value that $SOD(\bar{g})$ can take:

$$SOD(\bar{g}) = \sum_{i=1}^{n} d(g_i, \bar{g}) \geq \sum_{i=1}^{n} (|g_i| + |\bar{g}| - 2|\bar{g}|) = \sum_{i=1}^{n} |g_i| - n|\bar{g}| \tag{7}$$

Notice that, in this case, if $|\bar{g}| < |g_m(S)|$ then $|\bar{g}| < |g_i|$. Consequently the maximum value for $|g_m|$ in (1) will be precisely $|\bar{g}|$ and the minimum value for $SOD(\bar{g})$ will be obtained when $|\bar{g}| = |g_m|$ as expressed in equation (7).

At this point, using equations (6) and (7) and assuming that $|\bar{g}| < |g_m(S)|$ we arrive to the following conclusion:

$$SOD(\bar{g}) \geq \sum_{i=1}^{n} |g_i| - n|\bar{g}| > \sum_{i=1}^{n} |g_i| - n|g_m(S)| = SOD(g_m(S)) \tag{8}$$

But this is a contradiction because, by definition of the median, $SOD(\bar{g})$ must be minimum. Thus $|\bar{g}|$ must be greater or equal than $|g_m(S)|$.

Let's now proof the second part of equation (5) (i.e. $|\bar{g}| \leq |g_M(S)|$). Suppose now that $|\bar{g}| > |g_M(S)|$. In this case the term $SOD(g_M(S))$ will take this value:

$$SOD(g_M(S)) = \sum_{i=1}^{n} (|g_i| + |g_M(S)| - 2|g_i|) = n|g_M(S)| - \sum_{i=1}^{n} |g_i| \tag{9}$$

Again, equation 9 holds because if $g_M(S)$ is the minimum common supergraph of S, then any g_i will have precisely g_i as a maximum common subgraph between itself and $g_M(S)$ and consequently the term $|g_m|$ in (1) is exactly $|g_i|$.

To compute the minimum value of $SOD(\bar{g})$, if $|\bar{g}| > |g_M(S)|$ then every graph g_i can share at most $|g_i|$ nodes with $\bar{g}$ and then the maximum value for $|g_m|$ in (1) is $|g_i|$. Then:

$$SOD(\bar{g}) \geq \sum_{i=1}^{n} (|g_i| + |\bar{g}| - 2|\bar{g}_i|) = n|\bar{g}| - \sum_{i=1}^{n} |g_i| \tag{10}$$

Then, from equations (9) and (10), and assuming that $|\bar{g}| > |g_M(S)|$ we obtain:

$$SOD(\bar{g}) \geq n|\bar{g}| - \sum_{i=1}^{n} |g_i| > n|g_M(S)| - \sum_{i=1}^{n} |g_i| = SOD(g_M(S)) \tag{11}$$

Again, this is a contradiction and, thus $|\bar{g}|$ must be less or equal than $|g_M(S)|$.■

4.2 Reduction of the Upper Bound on the SOD of the Median Graph

Theorem 2: Let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs and $\bar{g}$ a possible median graph of S. Given the cost function and the distance measure presented in section 2.4, the $SOD(\bar{g})$ falls in the limits

$$SOD\left(\bar{g}\right) \leq SOD(g_m(S)) \leq min\left\{SOD\left(\bar{g_e}\right), SOD\left(\bar{g_u}\right)\right\} \tag{12}$$

Proof: First, we start by computing the term $min\left\{SOD\left(\bar{g_e}\right), SOD\left(\bar{g_u}\right)\right\}$. Using the definition of distance given in expression (1):

$$SOD\left(\bar{g_e}\right) = \sum_{i=1}^{n} d(g_i, \bar{g_e}) = \sum_{i=1}^{n} \left(|g_i| + |\bar{g_e}| - 2|\bar{g_e}|\right) = \sum_{i=1}^{n} |g_i|$$

Notice that, in this expression $\bar{g_e}$ is the empty graph. Then, the *mcs* between any graph g_i and $\bar{g_e}$ in expression (1) is $\bar{g_e}$, and $|\bar{g_e}| = 0$. A similar reasoning can be done for $SOD\left(\bar{g_u}\right)$. In this case, the *mcs* between any graph g_i and $\bar{g_u}$ is g_i, and $|\bar{g_u}| = \sum_{i=1}^{n} |g_i|$. Therefore,

$$SOD\left(\bar{g_u}\right) = \sum_{i=1}^{n} d(g_i, \bar{g_u}) = \sum_{i=1}^{n} \left(|g_i| + |\bar{g_u}| - 2|g_i|\right) = (n-1)\sum_{i=1}^{n} |g_i|$$

Thus, for $n \geq 2$

$$min\left\{SOD\left(\bar{g_e}\right), SOD\left(\bar{g_u}\right)\right\} = min\left\{\sum_{i=1}^{n} |g_i|, (n-1)\sum_{i=1}^{n} |g_i|\right\} = \sum_{i=1}^{n} |g_i|$$

Now we derive an expression for the term $SOD(g_m(S))$. If $g_m(S)$ is the maximum common subgraph of S, then any g_i will have precisely $g_m(S)$ as a maximum common subgraph between itself and $g_m(S)$. Therefore,

$$\begin{aligned} SOD(g_m(S)) &= \sum_{i=1}^{n} d(g_i, g_m(S)) = \sum_{i=1}^{n} \left(|g_i| + |g_m(S)| - 2|g_m(S)|\right) = \\ &= \sum_{i=1}^{n} |g_i| - n|g_m(S)| \end{aligned} \tag{13}$$

Thus, we have that $SOD(g_m(S)) \leq min\{SOD(\bar{g_e}), SOD(\bar{g_u})\} = \sum_{i=1}^{n} |g_i|$. In addition, by the definition of median graph, the inequality $SOD(\bar{g}) \leq SOD(g_m(S))$ must be satisfied. Consequently, Equation (12) holds.■

5 Genetic Search Algorithm

In this section we show how the new bounds presented in the previous section can be used to develop a new sub-optimal algorithm for the computation of the generalized median graph that takes advantage of these theoretical results to reduce the search space of the median graph. Computing the median graph is an optimization problem where a search space has to be explored to find the optimal solution. Among the several optimization techniques – such as Tabu search, genetic algorithms, etc.– that could be used, we have adopted a genetic search approach. This decision was motivated by different factors. Firstly, genetic search has already been successfully applied to the median graph computation. The use of the same strategy gives us the opportunity to be able to compare the two algorithms in a common framework. Secondly, as we will see in the next section, the codification of one possible solution as a chromosome is very intuitive and simple. Although this codification is not unique to the genetic algorithms, it will permit to manipulate and evaluate possible solutions in a very straightforward way.

5.1 Basics on Genetic Search

Genetic search techniques are general-purpose optimization methods inspired by the theory of the biological evolution. They have been successfully applied to difficult search tasks, optimization problems, machine learning, etc. It has also been shown that they are good candidates to give good approximate solutions to general NP-complete problems [23]. They have been applied to solve graph matching problems [1,11,34] and to compute approximate solutions for the generalized median graph [22].

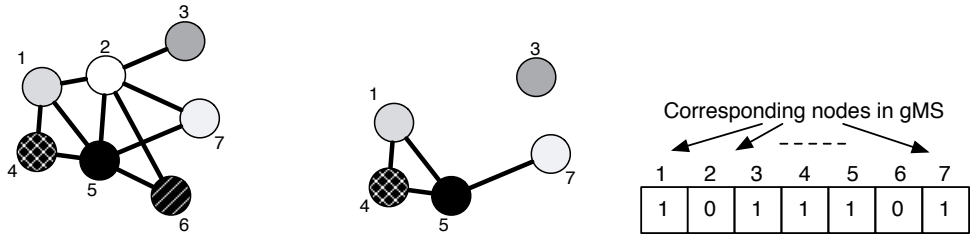
The basics of genetic algorithms are as follows. A possible solution of the problem is encoded using chromosomes. Each chromosome has a cost. Such cost is computed by means of a fitness function. Given an initial population of chromosomes, genetic algorithms use genetic operators to alter chromosomes in the population, generating a new population. The genetic operators are typically the crossover and mutation. In the former, a pair of chromosomes of the current population are randomly chosen and some of their positions are interchanged. The latter takes only one chromosome and alter some of its positions randomly. The process is iteratively repeated until one or more stop

conditions are satisfied. For more information about genetic algorithms the reader is referred to [26].

5.2 Our Approach

5.2.1 Chromosome Representation

In [18] it is shown that using the cost function and the distance based on the maximum common subgraph given in section 2.4, the possible candidate medians are all the possible induced subgraphs of $g_M(S)$. That is, the search space is composed only of all these induced subgraphs. This is an implication of the cost function given before: it is demonstrated in [7] that there exists an optimal edit path between two given graphs that implies neither non-identical node substitutions nor non-identical edge substitutions. Then, only node insertions and deletions need to be considered and all candidate medians can be obtained exploring only the induced subgraphs of $g_M(S)$. Then, the chromosome representation should be able to encode all of these induced subgraphs of $g_M(S)$. By the definition of induced subgraph given in Section 2.1, a subset of nodes of a given graph uniquely defines a subgraph. The set of edges linking the nodes will be determined by the set of edges of $g_M(S)$. Thus, we have chosen the size of the chromosome equal to the size of the $g_M(S)$. Each position in the chromosome is associated to one node of $g_M(S)$, and may store either a value of "1" or a value of "0" depending on whether that node belongs or not to the candidate median. In order to clarify such representation an example is given in Figure 1.



(a) (b) (c)

Fig. 1. A graph $g_M(S)$ (a), an induced subgraph g of $g_M(S)$ (b) and the chromosome representing g (c).

Assume that $g_M(S)$ is the graph shown in figure 1(a). As we can see, a number is assigned to each node of $g_M(S)$. A possible induced subgraph of $g_M(S)$ is shown in figure 1(b), which is composed only of the nodes 1,3,4,5 and 7 of $g_M(S)$. Then, the chromosome representation of such induced subgraph is shown in the figure 1(c). In such chromosome the total number of positions is equal to the number of nodes of $g_M(S)$. Notice that the chromosome has only

set to "1" the positions of the nodes of $g_M(S)$ which are present in the induced subgraph. The edges connecting these nodes will be the same as in $g_M(S)$. In this way, the chromosome represents the induced subgraph of $g_M(S)$ of the figure 1(b).

5.2.2 Fitness Function

The fitness function of each chromosome corresponds to the SOD of the induced subgraph of $g_M(S)$ represented by the chromosome. That is, if the chromosome c represents a graph g, then its fitness function $f(c)$ is:

$$f(c) = SOD(g, S) = \sum_{i=1}^{n} d(g, g_i) = \sum_{i=1}^{n} (|g| - |g_i| - 2|mcs(g, g_i|) \tag{14}$$

Clearly, the lower its fitness function is, the better the chromosome is. The computational complexity of this fitness function is related to the computational complexity of the maximum common subgraph of two graphs, which is exponential in the general case. Nevertheless, such computational complexity becomes polynomial when the considered graphs have unique node labels [24].

5.2.3 Genetic Operators

We apply the classical operators of the genetic algorithms adapted to this particular case in order to include the new bounds that we have presented in Section 4. In our algorithm, the roulette wheel sampling implementing fitness-proportionate selection is chosen to create the descendants (also called offspring). Conceptually, it is equivalent to give a slice of a circular roulette wheel to each chromosome, proportional in area to the fitness of the chromosome. The crossover operator simply interchanges an arbitrary position of two chromosomes (selected with a uniform probability) to form two offspring. Mutation is accomplished by changing randomly a number in the array with a mutation probability. After the genetic operators have been applied and a new population is created, every chromosome is checked in order to validate whether it fulfils the bounds given in the last section. If the chromosome is out of such limits, it is randomly altered until it fulfils the conditions. This procedure has two effects. On the one hand it reduces the search space from all the possible induced subgraphs of $g_M(S)$ to only the induced subgraphs that fulfil the conditions given in the last section. On the other hand, as the search space is reduced and the non-admissible candidate medians will never appear in the population, the convergence of the algorithm is expected to be faster compared with the same algorithm without taking into account the new limits.

5.2.4 Population Initialization

The length of the initial population is set according to a predefined value K, determined empirically. Then, the first n chromosomes (with $n \leq K$) are set as the induced subgraphs of $g_M(S)$ corresponding to the n graphs in S. It assures that the initial population includes the graphs in S and by extension it includes the set median graph, which is a potential generalized median graph. The remaining K-n chromosomes are generated randomly but all of them must fulfil the new bounds given in Section 4.

5.2.5 Termination Condition

The population evolution process is continued until one of the two following conditions is fulfilled. The first criterion is that the maximum number of generations (which is set according to a predefined constant at the beginning of the algorithm) is reached. The second stop condition is related to the best SOD in the population. If the chromosome in the population has a SOD less than the SOD of the set median graph, then the algorithm finishes too.

6 Experimental Results

In order to experimentally evaluate both the new theoretical properties and the new genetic approach, we present in this section three experiments using a real database of graphs representing webpages. Such graphs have a large number of nodes (around 200) but they are a particular class of graphs with unique node labels. Such kind of graphs allow the computation of the maximum common subgraph of two graphs in polynomial time [24]. That makes the computation of the edit distance based on the maximum common subgraph (and for extension, the computation of the median graph) applicable to large graphs. Due to the large size of graphs that we manage in this experiment, all other methods for the median graph computation are not applicable.

6.1 Dataset

Our dataset is composed of 2,340 documents representing webpages belonging to 6 main classes (Business (B), Entertainment (E), Health (H), Politics (P), Sports (S) and Technology (T)). It is the same as that used in [32]. These web documents were originally hosted at Yahoo as news pages (http://www.yahoo.com). The graph-based representation of these webpages is as follows. First, all words appearing in the web document are converted

into nodes in the web graph, except for those which contain little information, i.e. conjunctions, stop words, etc. The nodes are attributed with the corresponding word and its frequency. That is, even if the word appears more than once in the web document, only one node is added to the graph and the frequency of the word is used as an additional attribute. Then, if a word w_i immediately precedes the word w_j in the document, a directed edge between the nodes corresponding to both words is added to the graph. In order to keep the essential information of the document, only the most frequently used words (nodes) are kept in the graph and the terms are combined to the most frequently occurring form. Table 1 shows the number of graphs in each class.

Table 1
Number of graphs in each class.

	Class					
	B	E	H	P	S	T
Number of graphs	142	1389	494	114	141	60

6.2 Parameters of the Genetic Algorithm

Table 2 shows the basic configuration parameters for the genetic algorithm. We have chosen the value for these parameters empirically. For the two first parameters, we have chosen the same value as in [22]. The value of the population size has been set in such a way that both the computation time of each iteration and the convergence speed are optimized. Finally, as the size of the chromosome can be large, the maximum number of iterations has been set in order to allow the genetic algorithm to iterate a sufficient number of times to deal with such a huge search space.

Table 2
Configuration Parameters for the Genetic Algorithm.

Parameter	Value
Mutation probability	0.1
Crossover probability	0.9
Initial population size (K)	20
Maximum number of iterations	400

6.3 Experiments

In this section we present three experiments. In the first one, we show the computational effort that is needed to synthesize a median graph. In a second

experiment, we evaluate the quality of the median graph according to the SOD. Finally, in a third experiment, we conduct a preliminary classification experiment in order to assess the median as a good representative of a given set of graphs.

6.3.1 Experiment 1: Median Computation

This experiment was intended to quantitatively evaluate the median graph computation achieved by the genetic approach. To this end, we computed the median graph of each class using 3, 4, 5, 6 and 7 graphs for each class, randomly selected.

Tables 3 and 4 show some interesting results of the median graph computation. In both tables, the first row represents the sum of nodes of all the graphs used to compute the median, the second row depicts the number of iterations needed to achieve a graph with a SOD better than the set median graph and the third row shows the total computation time. While in the first table the results are grouped by class, in the second table the results are shown as a function of the number of graphs used to compute the median. In both cases, the results are the mean values over every class or over the number of graphs, respectively.

Table 3
Statistics for median graph grouped by class

	Class					
	B	E	H	P	S	T
$\sum \lvert g_i \rvert$	1,021.2	777	845	940.4	806.2	566.2
# iterations	66.40	8.40	2.40	11.80	32.20	3.20
Computation time (sec)	4,636	274.1	65.05	179.6	1,428.748	75.8

Table 4
Statistics for median graph grouped by the number of graphs used to compute the median.

	Number of graphs in S				
	3	4	5	6	7
$\sum \lvert g_i \rvert$	467.1	701.1	813.3	1,041.1	1,107.1
# iterations	1.5	58.50	2.83	21.6	19.1
Computation time (sec)	13.8	3,362.4	82.3	1,278.2	2,159.6

The results show that the number of iterations needed to find a median better than the set median graph is very low (less than 100 in all cases). It is not proven that the obtained graph is the true median graph, but it means that the genetic algorithm always find a graph with a SOD better than the SOD of the set median graph.

In addition, in the first and third row of both tables, we can observe that the sum of nodes of the graphs used to compute the median range from 400 to 1,000, while the computation times range from 13 to 3,000 seconds. Such numbers show that the computation of the median graph can be applied to a real dataset of large graphs with reasonable computation times. It is important to notice that previously existing methods could not be applied to this kind of data due to their high computational requirements.

One of the possible problems of the genetic algorithms is the premature convergence. Some additional experiments performed on very limited data (where the true median can be computed) show that our approach is able, in most of the cases, to obtain the optimal solution. In the experiments reported in this paper, due to the size of the graphs, we are not able to compute the true median. However the algorithm performs quite good in terms of both the number of comparisons and the SOD (as we will see in the next section). Therefore, this result gives us the proof that the premature convergence does not exist here as well. For this reason we have not adopted any specific strategy to solve this problem.

6.3.2 Experiment 2: Median Accuracy

The results shown above are suitable to quantitatively evaluate the proposed algorithm in terms of computation time with respect to the size of the set S. Nevertheless, it is also of interest to qualitatively evaluate the median according to the final SOD. In this case, due to the size and the number of graphs of the dataset, it is not possible to compare our method with other approaches, since the existing methods cannot deal with such large sets and graphs. Thus, using the same dataset as in the first experiment, we have decided to compare the SOD of the median obtained using the genetic algorithm with the SOD of the set median. This comparison can give a good idea of whether it is potentially a good median.

Figures 2 and 3 show the results of this comparison as a function of the classes in the dataset, and the number of graphs in the set S, respectively.

The results of Figure 2 show that we always obtain medians with a SOD lower than the set median SOD for all the classes. These results validate that the method is able to obtain good approximations of the median graph regardless of the class.

Figure 3 shows that we also obtain a better SOD with our method than with the set median for any number of graphs in the set. What is important in this figure is the tendency in the difference between the set median SOD and the SOD of the approximate median. This difference increases as the number of graphs in S increases. This tendency suggests that the more information of

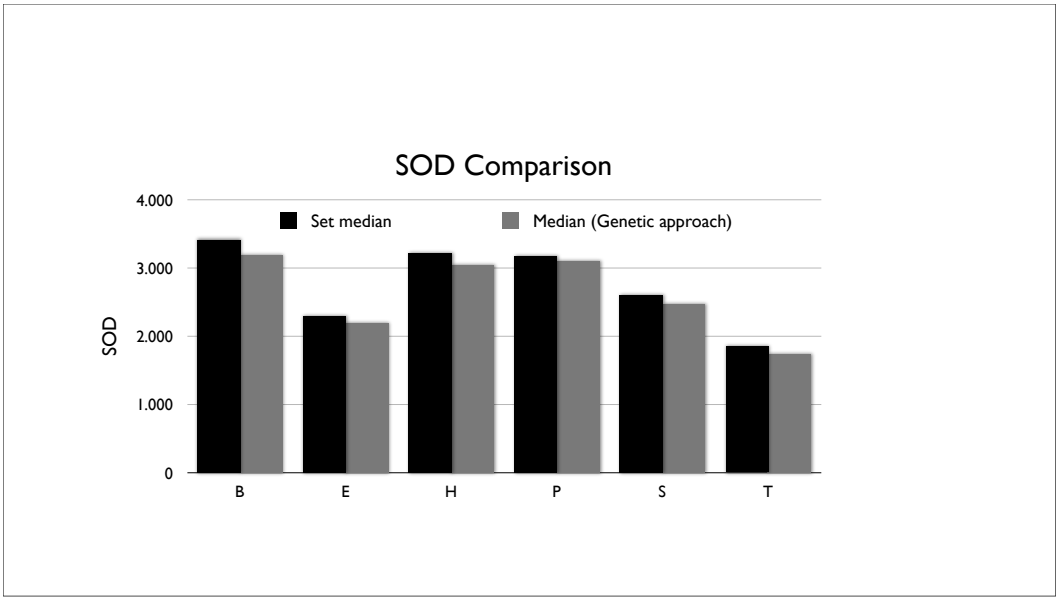


Fig. 2. SOD comparison for the Set Median and the Approximate Genetic algorithm, function for the different classes.

the class the method has (more elements in S), better representations is able to obtain.

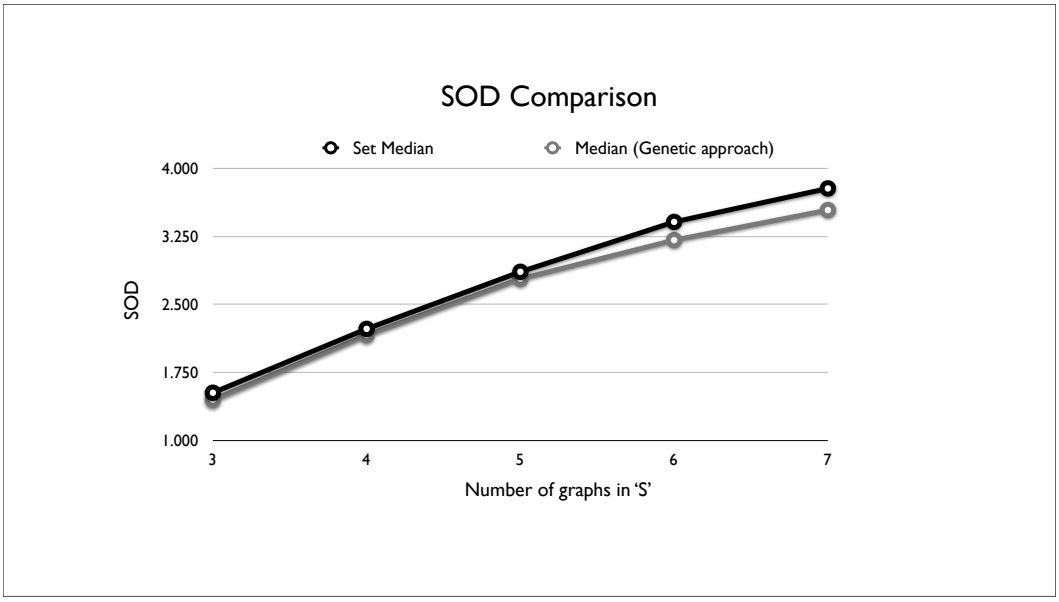


Fig. 3. SOD comparison function of the number of graphs in S.

With these results at hand, we can conclude that we obtain good approximations of the median graph with this new genetic approach.

6.3.3 Experiment 3: Classification Accuracy

In the previous experiments we have shown that, with the new algorithm, we are able to compute good approximations of the median graph in a reasonable time for sets of real data with large graphs. In this experiment we present a limited and preliminary classification experiment that intends, for the first time, to evaluate the median graph as a representative of a set of graphs in a real application. The median is used to obtain the representative of each class using the training set. Then, every element in the test set is classified according to the class of the most similar median. Thus, we avoid comparing all the elements in the test set against all the elements in the training set. The results are compared with those obtained using a classical nearest-neighbor classifier.

The experiment setup is as following. For every class, except for class T where all the elements are used, we took randomly 60 out of all the elements in the class. Thus, we have a total number of 360 elements to perform the classification experiment. In order to better generalize the results, we performed 10 repetitions of the classification task. To this end, we divided each class into 10 sets of 6 elements each. For a given repetition, the training set is composed of one of these sets of 6 elements per class (a total number 36 elements). It is used both for the 1NN classification and to compute the median graph. The remaining 324 elements (54 elements per class) are used as the test set.

Figure (4) shows, for each repetition of the experiment, the classification accuracy both the nearest neighbor and the median graph approach. The results are the mean values over all classes. These results show that in general, the 1NN approach clearly outperforms the median approach, except for the repetition number 9. Nevertheless it is important to remark two issues: firstly, the number of comparisons (and therefore, the total computation time) needed using the median graph is 6 times lower that the number of comparisons needed using the nearest neighbor (specifically, 1,944 against 11,664 comparisons). Secondly, the results of the median graph show a great variability depending on the set of graphs used as training set.

In order to better analyze these results, we have performed an extended classification experiment, combining the computation of the median graph with the classical 1NN classifier. We compare every element of the test set against the median graph of all the classes and we use these results as a filter before applying the 1NN classifier. For every element in the test set, we rank all the classes according to the distance to the median graphs. After that, the element is classified using the 1NN classifier but using only the elements in the training set of the best k classes according to the previous ranking, instead of using all the classes as in the 1NN classical approach. It is clear that, if k is set to 1, then the results are the same as those obtained with the classification

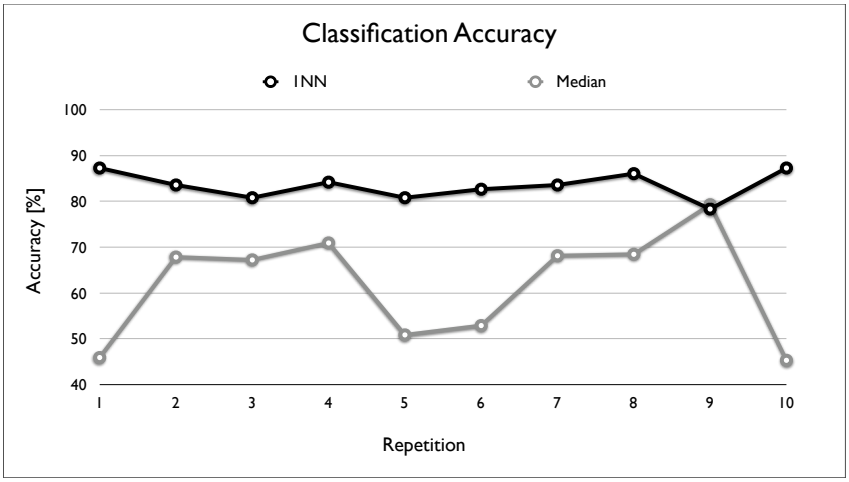


Fig. 4. Classification accuracy for both the 1NN and the median graph approach, for every repetition of the experiment.

using simply the median. Conversely, if k is equal to 6, then the results are the same as in the classical 1NN classifier.

Figure 5 shows, for every value of k, the maximum, the average and the minimum classification accuracy achieved along the 10 repetitions of the experiment. In addition, Figure 6 show the results for two individual repetitions, representing extreme cases of good (i.e. the median approach outperforms the 1NN classifier for $k \leq 6$) and bad (i.e. the median approach does not outperform the 1NN classifier for $k \leq 6$) results, respectively. The first remark to be done is that, while the 1NN classifier ($k = 6$) exhibits a good stability, the results when the classification depends on the median graph ($k = 1, \ldots, 5$) show a high variability. In some cases, as in the Figure 6(a), we can obtain better results using the median graph with a low value of k than with the 1NN classifier. In Figure 5 we can observe that, even for $k = 1$ or $k = 2$, the best results using the median graph can outperform the worse results using the 1NN classifier. That means that, with the median graph, it is possible to achieve similar results to the 1NN, but using a lower number of comparisons. This reduction in the computation time can be very important for some real applications. In order to reinforce this idea we show, in Figure 7, for every value of k the number of repetitions where this value of k permits to obtain the same or better classification results as in the 1NN classifier. We can see that, in almost half of the repetitions, we only need at most 3 classes ($k = 4$) to obtain better results. However, in the worst cases (for example, Figure 6(b)) the results of the median graph approach are very poor. This variability suggests a dependency of the quality of the median graph on the training set and opens the door to find methods for the selection of the best graphs for the computation of the median graph.

Finally, it is important to recall that this is a preliminary experiment with the

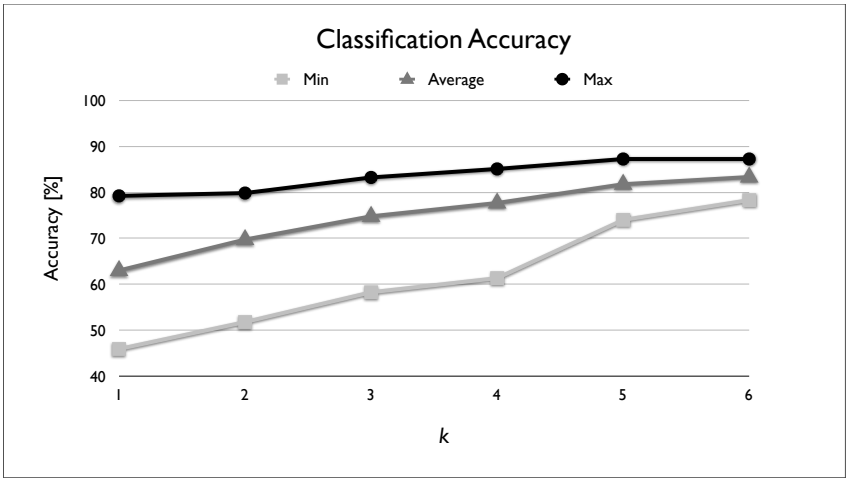


Fig. 5. Maximum, average and minimum classification accuracy for each value of k along the 10 repetitions.

(a)

(b)

Fig. 6. One of the best (a) and worse (b) cases among the 10 repetitions.

aim to show that the median graph can be a good option to represent a set of graphs. For a real application of the median graph to classification problems, further work is required in order to find the best strategy to use the median graph in the existing classification methods.

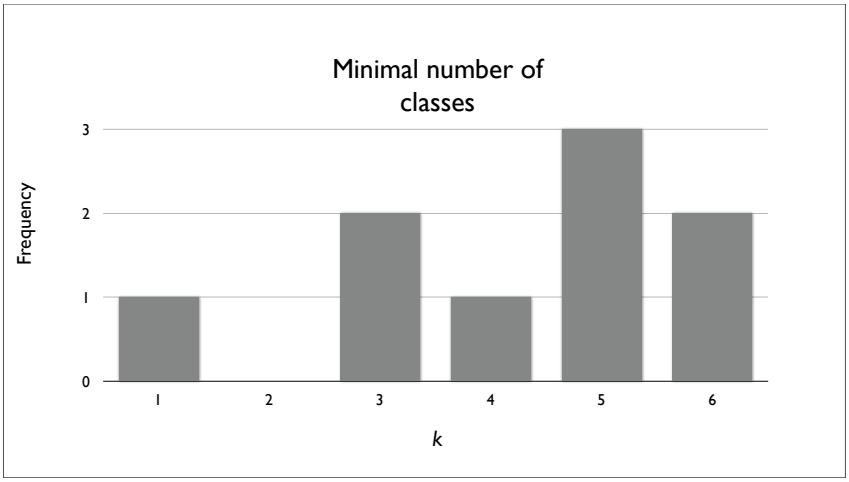


Fig. 7. Histogram of minimum number of classes to achieve the same results as in the 1NN classifier.

7 Conclusions

The median graph has been presented as a good alternative to compute the representative of a set of graphs. Although some theoretical properties and algorithms have been introduced so far related to the median graph, existing methods do not permit to used it in real pattern recognition applications.

In this paper we have derived new theoretical properties of the median graph and we have developed a new algorithm using these properties that permit to extend the computation of the median graph to real datasets with large graphs. The main contributions of this paper are twofold. Firstly, from a theoretical point of view, we have shown that using a particular cost function and a distance measure based on the maximum common subgraph, the original bounds for the median graph related to its size and its sum of distances can be reduced. Such reductions can be used either to obtain a better knowledge of the median graph or can be used to develop more efficient and accurate algorithms. This is precisely, the second contribution of this work. Using the new bounds we present a new approximate algorithm for the median graph computation based on genetic search.

With these new bounds and the new algorithm we performed a set of experiments using webpages extracted from real data. The first conclusion of these experiments is that, with this new algorithm, we are able to obtain accurate approximations of the median graph (in terms of SOD) with a computation time that permits to work with sets of graphs composed of around 200 nodes each. Although the applicability of the median graph to real problems is still limited, these results show that the concept of median graph can be used in real world applications. It demonstrates, for the first time, that the median graph is a feasible alternative to obtain a representative of a set of graphs.

For instance, we have shown in a preliminary experiment that the classification using the median graph can obtain similar results as a nearest-neighbor classifier but with a much lower computation time.

Nevertheless, there are still a number of issues to be investigated in the future. These theoretical results give us the opportunity to increase the knowledge about the median graph and open a door towards a better understanding of its behavior. In this sense, more accurate bounds or properties might be investigated using these new results. Such advances in the theoretical level may lead also to obtain more accurate and efficient approximate solutions of the median graph, either by producing enhanced versions of the existing algorithms or by developing new approaches to compute the median graph. Although the genetic algorithms are a class of optimization techniques widely used to solve high computational problems, they are not the unique alternative. Other optimization techniques such as Tabu search seem also suitable to address the median graph computation under this cost function in order to obtain better accuracy and computation time. Thus, applying other optimization algorithms remains as an open path to be explored. In addition, the preliminary experiments on classification open the possibility of extending the application of the median graph to classification algorithms where a representative of the set of graphs is required.

Acknowledgements

This work has been partially supported by the research Fellowship 401-027 (UAB), the Cicyt project TIN2006-15694-C02-02 (Ministerio Ciencia y Tecnología) and the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018). We would like to thank K. Riesen from the University of Bern for his help with the webpages database and to make it available to us.

References

[1] S. Auwatanamongkol, Inexact graph matching using a genetic algorithm for image recognition, Pattern Recognition Letters 28 (12) (2007) 1428–1437.

[2] E. Balas, C. S. Yu, Finding a maximum clique in an arbitrary graph, SIAM J. Comput. 15 (4) (1986) 1054–1068.

[3] H. Bunke, On a relation between graph edit distance and maximum common subgraph, Pattern Recognition Letters 18 (8) (1997) 689–694.

[4] H. Bunke, G. Allerman, Inexact graph matching for structural pattern recognition, Pattern Recognition Letters 1 (4) (1983) 245–253.

[5] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, M. Vento, A comparison of algorithms for maximum common subgraph on randomly connected graphs, in: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops SSPR 2002 and SPR 2002, Windsor, Ontario, Canada, August 6-9, 2002, Proceedings. Lecture Notes in Computer Science Vol. 2396, 2002.

[6] H. Bunke, P. Foggia, C. Guidobaldi, M. Vento, Graph clustering using the weighted minimum common supergraph, in: Graph Based Representations in Pattern Recognition, 4th IAPR International Workshop, GbRPR 2003, York, UK, June 30 - July 2, 2003, Proceedings. Lecture Notes in Computer Science Vol. 2726, 2003.

[7] H. Bunke, X. Jiang, A. Kandel, On the minimum common supergraph of two graphs, Computing 65 (1) (2000) 13–25.

[8] H. Bunke, A. Münger, X. Jiang, Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem, Pattern Recognition Letters 20 (11-13) (1999) 1271–1277.

[9] H. Bunke, K. Shearer, A graph distance metric based on the maximal common subgraph, Pattern Recognition Letters 19 (3-4) (1998) 255–259.

[10] D. Conte, P. Foggia, M. Vento, Challenging complexity of maximum common subgraph detection algorithms: A performance analysis of three algorithms on a wide database of graphs, Journal of Graph Algorithms and Applications 11 (1) (2007) 99–143.

[11] A. D. J. Cross, R. C. Wilson, E. R. Hancock, Inexact graph matching using genetic search, Pattern Recognition 30 (6) (1997) 953–970.

[12] P. J. Durand, R. Pasari, J. W. Baker, C. che Tsai, An efficient algorithm for similarity analysis of molecules, Internet Journal of Chemistry 2 (17).

[13] M.-L. Fernández, G. Valiente, A graph distance metric combining maximum common subgraph and minimum common supergraph, Pattern Recognition Letters 22 (6/7) (2001) 753–758.

[14] M. Ferrer, F. Serratosa, A. Sanfeliu, Synthesis of median spectral graph, in: Second Iberian Conference of Pattern Recognition and Image Analysis. Volume 3523 LNCS, 2005.

[15] M. Ferrer, F. Serratosa, E. Valveny, On the relation between the median and the maximum common subgraph of a set of graphs, in: 6th IAPR-TC-15 International Workshop, GbRPR 2007, Alicante, Spain, June 11-13, 2007, Proceedings, vol. 4538 of Lecture Notes in Computer Science, Springer, 2007.

[16] M. Ferrer, E. Valveny, F. Serratosa, Spectral median graphs applied to graphical symbol recognition, in: J. F. M. Trinidad, J. A. Carrasco-Ochoa, J. Kittler (eds.), CIARP, vol. 4225 of Lecture Notes in Computer Science, Springer, 2006.

[17] M. Ferrer, E. Valveny, F. Serratosa, Bounding the size of the median graph, in: J. Martí, J.-M. Benedí, A. M. Mendonça, J. Serrat (eds.), IbPRIA (2), vol. 4478 of Lecture Notes in Computer Science, Springer, 2007.

[18] M. Ferrer, E. Valveny, F. Serratosa, Generalized median graph: A new optimal algorithm based on the maximum common subgraph, CVC Technical Report no. 109 (2007).

[19] A. Hlaoui, S. Wang, A new median graph algorithm, in: Graph Based Representations in Pattern Recognition, 4th IAPR International Workshop, GbRPR 2003, York, UK, June 30 - July 2, 2003, Proceedings. Lecture Notes in Computer Science Vol. 2726, 2003.

[20] A. Hlaoui, S. Wang, Median graph computation for graph clustering, Soft Comput. 10 (1) (2006) 47–53.

[21] X. Jiang, A. Münger, H. Bunke, Synthesis of representative graphical symbols by computing generalized median graph, in: A. K. Chhabra, D. Dori (eds.), GREC, vol. 1941 of Lecture Notes in Computer Science, Springer, 1999.

[22] X. Jiang, A. Münger, H. Bunke, On median graphs: Properties, algorithms, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 23 (10) (2001) 1144–1151.

[23] K. A. D. Jong, W. M. Spears, Using genetic algorithms to solve NP-complete problems, in: J. D. Schaffer (ed.), ICGA, Morgan Kaufmann, 1989.

[24] A. Kandel, H. Bunke, M. Last, Applied Graph Theory in Computer Vision and Pattern Recognition (Series in Computational Intelligence), Springer-Verlag, Berlin, 2007.

[25] J. J. McGregor, Backtrack search algorithms and the maximal common subgraph problem, Software - Practice and Experience 12 (1) (1982) 23–24.

[26] M. Mitchel, An Introduction to Genetic Algorithms, MIT Press., 1996.

[27] A. Münger, Synthesis of prototype graphs from sample graphs, in: Diploma Thesis, University of Bern (in German), 1998.

[28] M. Neuhaus, H. Bunke, A quadratic programming approach to the graph edit distance problem, in: 6th IAPR-TC-15 International Workshop, GbRPR 2007, Alicante, Spain, June 11-13, 2007, Proceedings, vol. 4538 of Lecture Notes in Computer Science, Springer, 2007.

[29] M. Neuhaus, K. Riesen, H. Bunke, Fast suboptimal algorithms for the computation of graph edit distance, in: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2006 and SPR 2006, Hong Kong, China, August 17-19, 2006, Proceedings. Lecture Notes in Computer Science 4109, 2006.

[30] K. Riesen, M. Neuhaus, H. Bunke, Bipartite graph matching for computing the edit distance of graphs, in: 6th IAPR-TC-15 International Workshop, GbRPR 2007, Alicante, Spain, June 11-13, 2007, Proceedings, vol. 4538 of Lecture Notes in Computer Science, Springer, 2007.

[31] A. Sanfeliu, K. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Transactions on Systems, Man and Cybernetics 13 (3) (1983) 353–362.

[32] A. Schenker, H. Bunke, M. Last, A. Kandel, Graph-Theoretic Techniques for Web Content Mining (Machine Perception and Artificial Intelligence) (Series in Machine Perception and Artificial Intelligence), World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.

[33] Y. Wang, C. Maple, A novel efficient algorithm for determining maximum common subgraphs, in: 9th International Conference on Information Visualisation, IV 2005, 6-8 July 2005, London, UK, IEEE Computer Society, 2005.

[34] Y.-K. Wang, K.-C. Fan, J.-T. Horng, Genetic-based search for error-correcting graph isomorphism, IEEE Transactions on Systems, Man, and Cybernetics, Part B 27 (4) (1997) 588–597.

[35] D. White, R. C. Wilson, Mixing spectral representations of graphs, in: 18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China, IEEE Computer Society, 2006.