# Heuristics for the generalized median graph problem

Leonardo M. Musmanno and Celso C. Ribeiro

Universidade Federal Fluminense, Institute of Computing, Niterói, RJ 24210-240, Brazil
{lmusmanno,celso}@ic.uff.br

Structural approaches for pattern recognition frequently make use of graphs to represent objects. The concept of object similarity is of great importance in pattern recognition. The graph edit distance is often used to measure the similarity between two graphs. It basically consists in the amount of distortion needed to transform one graph into the other. The median graph of a set $S$ of graphs is a graph of $S$ that minimizes the sum of its distances to all other graphs in $S$. The generalized median graph of $S$ is any graph that minimizes the sum of the distances to all graphs in $S$. It is the graph that best captures the information contained in $S$ and may be regarded as the best representative of the set. Exact methods for solving the generalized median graph problem are capable to handle only a few number of small graphs. We propose two new heuristics for solving the generalized median graph problem: a greedy adaptive algorithm and a GRASP heuristic. Numerical results indicate that both heuristics can be used to obtain good approximate solutions for the generalized median graph problem, significantly improving the initial solutions and the median graphs. Therefore, the generalized median graph can be effectively computed and used as a better representation than the median graph in a number of relevant pattern recognition applications. This conclusion is supported by experiments with a classification problem, illustrating that the approach based on the generalized median graph is competitive with the $k$-NN classifier.

## 1 Introduction and motivation

Two approaches are often used in pattern recognition problems. In the case of statistical approaches, objects are represented by vectors in $\mathbb{R}^n$ representing a set of measures. Structural approaches for pattern recognition frequently make use of graphs to represent objects. They make it possible to represent relationships between different parts of the same object or pattern. The use of graphs also allows the representation of objects with a different number of items or parts by graphs with different numbers of vertices and edges. One important step towards the combination of these two approaches consists in using graphs to represent objects and associating attributes (i.e., feature vectors) to their vertices and edges. Conte et al. (2004) have noticed that graph matching techniques have been successfully applied to many areas, such as biology and bio-medicine. As an example, Fischer et al. (2002) applied a graph matching identification approach for the retrieval of diatoms, a unicellular algae found in water and other places. The retrieval is based on the matching of labeled grid graphs (a regular, rectangular arrangement of nodes overlayed on an image) carrying texture information of the underlying diatom. Each node of the graph is labeled with texture features describing a rectangular sub-region of the object. Based on the grid graph representation, the problem of diatom identification can be formulated as a labeled graph matching task, where the goal is to find an optimal one-to-one correspondence between the nodes of an input graph and the nodes of a graph stored in an image database. Other examples of graph matching applications include 2D and 3D image analysis (Torsello and Hancock, 2003), document processing (Suganthan and Yan, 1998), biometric identification (Fan et al., 1998; Hong et al., 2000; Lades et al., 1993), image databases (Berretti et al., 2001), and video analysis (Shearer et al., 2001).

**Definition 1 (Labeled graph)** *Given a finite attribute set $L$, a labeled graph $G = (V, E, \mu, \upsilon)$ is a quadruple defined by a set $V$ of vertices, a set $E \subseteq V \times V$ of edges, a function $\mu : V \to L$ that associates an attribute value in $L$ to each vertex, and a function $\upsilon : E \to L$ that associates an attribute value in $L$ to each edge.*

Unless otherwise stated, we refer to a labeled graph in the remainder of this text simply by a graph. There is no restriction on the nature of set $L$. In most cases, $L = \mathbb{R}^n$ or $L$ is a discrete set of labels. If

necessary, $L$ may also include a special label $\varepsilon$ to indicate that no specific label is assigned to a node or edge. If all elements of the graph are labeled with $\varepsilon$, the graph is called *unlabeled*. A weighted graph is a special case of a labeled graph, where all nodes are labeled with $\varepsilon$ and each edge is labeled with a real number. The concept of object similarity is of great importance in pattern recognition. The graph edit distance is often used to measure the similarity between two (labeled) graphs. Basically, the graph edit distance $d(G_1, G_2)$ between two graphs $G_1$ and $G_2$ takes into account the costs of the edit operations that are necessary to transform $G_1$ into $G_2$.

Given a set of objects, their median is frequently used to indicate the element that best represents the set, i.e., the object that best captures the information presented in all elements. Jiang et al. (2001) introduced the concepts of median and generalized median graphs, which have a great potential for applications, e.g. in clustering or classification (Mukherjee et al., 2007, 2009), chemistry, molecular biology, and handwriting recognition.

**Definition 2 (Median graph)** *Given a set $S = \{G_1, \ldots, G_n\}$ of labeled graphs defined over an alphabet $L$, its median graph is any graph $\hat{G} = argmin\{\sum_{i=1}^{n} d(G, G_i) : G \in S\}$ that minimizes the sum of its distances to all graphs in $S$.*

The median graph $\hat{G}$ of $S$ is necessarily one of its elements. If one seeks a representative graph that is not restricted to belonging to $S$, the following definition applies:

**Definition 3 (Generalized median graph)** *Given a set $S = \{G_1, \ldots, G_n\}$ of labeled graphs defined over an alphabet $L$, its generalized median graph is any graph $\overline{G} = argmin\{\sum_{i=1}^{n} d(G, G_i) : G \in U\}$ that minimizes the sum of its distances to all graphs in $S$, where $U$ is the set of all labeled graphs defined over the alphabet $L$.*

Therefore, the generalized median graph $\overline{G}$ of $S$ is any graph that minimizes the sum of its distances to the graphs in $S$, regardless of belonging to $S$ or not.

The median graph can be computed in $O(n^2\delta)$, where $n$ is the number of graphs in $S$ and $\delta$ is the complexity of the computation of the distance function $d(.)$. Since the computation of the graph edit distance between two graphs is ***NP***-hard (Voigt, 2010; Zeng et al., 2009), the median graph problem considering the graph edit distance cannot be solved in polynomial time, unless ***P=NP***. Finding the median graph is ***NP***-complete even for strings (de la Higuera and Casacuberta, 2000).

The computation of the generalized median graph is even more time consuming, because it does not depend only on the complexity of the distance function $d(.)$, but also on the size of the search space $U$. Exact algorithms cannot deal with large graphs, with their application being restricted to small problems involving sets of graphs with no more than 25 vertices altogether. Earlier approximate approaches to find the generalized median graph include greedy (Hlaoui and Wang, 2003) and genetic (Jiang et al., 2001) algorithms. Ferrer (2008) proposed a new genetic algorithm (Ferrer et al., 2009) and a heuristic based on graph embedding (Ferrer et al., 2008, 2010). Other approaches use the relationship between common-labeling and the median graph to compute and find bounds on the cost of the median graph (Rebagliati et al., 2012).

In this work, we propose two new heuristics for solving the generalized median graph problem: a greedy adaptive algorithm and a GRASP heuristic. In the next section, we present the preliminaries for the new heuristics, including the definitions of the graph edit instance, of the maximum common subgraph, and of the minimum common supergraph of a set of graphs. An adaptive greedy algorithm is presented in Section 3, while a GRASP heuristic is described in Section 4. Computational experiments reporting experimental results indicating that the proposed heuristics can be used to obtain good approximate solutions for the generalized median graph problem in reasonable computation times are presented in Section 5. These results show that good approximations to the generalized median graph can be effectively computed by the heuristics proposed in this paper, making it a better representation than the median graph to be used in a number of relevant pattern recognition applications. Concluding remarks are drawn in the last section, where we also comment on additional computational experiments showing

that not only the generalized median graph is a good representative for graph sets, but also that classification approaches based on the generalized median graph are competitive with the $k$-NN classifier, in which an object to be classified is assigned to the most common class among its $k$ nearest neighbors.

## 2 Preliminaries

**Definition 4 (Subgraph)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs. $G_1$ is said to be a subgraph of $G_2$ if and only if $V_1 \subseteq V_2$, $E_1 \subseteq E_2$, $\mu_1(u) = \mu_2(u)$, $\forall u \in V_1$, and $\upsilon_1(e) = \upsilon_2(e)$, $\forall e \in E_1$. In this case, we say that $G_1 \subseteq G_2$.*

**Definition 5 (Induced subgraph)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs such that $G_1 \subseteq G_2$. $G_1$ is the subgraph induced in $G_2$ by $V_1$ if and only if $E_1 = E_2 \cap (V_1 \times V_1)$.*

**Definition 6 (Graph isomorphism)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs. $G_1$ and $G_2$ are said to be isomorphic if there is a bijection $f : V_1 \to V_2$ such that: (i) $\mu_1(u) = \mu_2(f(u))$, $\forall u \in V_1$; (ii) for each edge $e_1 = (u, v) \in E_1$, there exists an edge $e_2 = (f(u), f(v)) \in E_2$ with $\upsilon_1(e_1) = \upsilon_2(e_2)$; and (iii) for each edge $e_2 = (u, v) \in E_2$, there exists an edge $e_1 = (f^{-1}(u), f^{-1}(v)) \in E_1$ with $\upsilon_1(e_1) = \upsilon_2(e_2)$.*

**Definition 7 (Subgraph isomorphism)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs. There exists a subgraph isomorphism from $G_1$ to $G_2$ if there exists an induced subgraph $G \subseteq G_2$ and an injection $f : V_1 \to V_2$ defining a graph isomorphism from $G_1$ to $G$.*

**Definition 8 (Common subgraph)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs. A graph $G$ is a common subgraph of $G_1$ and $G_2$ if and only if there is a subgraph isomorphism from $G$ to $G_1$ and another from $G$ to $G_2$.*

In the following, we denote by $\#(G)$ the number of vertices of a graph $G$.

**Definition 9 (Maximum common induced subgraph)** *A common induced subgraph $G$ of $G_1$ and $G_2$ is maximum if all other common subgraphs of $G_1$ and $G_2$ have at most $\#(G)$ vertices.*

The decision version of the problem of finding a maximum common subgraph is *NP*-complete (Garey and Johnson, 1979). The maximum common subgraph of two graphs may be computed by a backtrack strategy (McGregor, 1982). Alternatively, one may seek a maximum clique in an auxiliary graph built from $G_1$ and $G_2$, which is then transformed into the maximum common subgraph (Balas and Yu, 1986; Durand et al., 1999). These three exact algorithms have the same time complexity $O((|V_2|+1)!/(|V_2|-|V_1|+1)!)$. They have been compared in (Bunke et al., 2002; Conte et al., 2007), with the numerical results indicating no clear advantage of one algorithm over the others.

In the following, we use the algorithm of Durand-Pasari (Durand et al., 1999) to compute the maximum common subgraph of two graphs $G_1$ and $G_2$ and we denote its output by $\mathrm{MaxSub}(G_1, G_2)$. Given the two graphs $G_1$ and $G_2$, the first step of the Durand-Pasari algorithm is the construction of an auxiliary (undirected) graph whose nodes correspond to pairs $(n_1, n_2)$ of nodes of the two original graphs, where $n_1 \in V_1$, $n_2 \in V_2$, and $\mu_1(n_1) = \mu_2(n_2)$. Edges of the auxiliary graph represent the compatibility between pairs of nodes: the node corresponding to the pair $(n_1, n_2)$ is connected to the node corresponding to the pair $(m_1, m_2)$ if and only if edge $(n_1, m_1)$ of $G_1$ has the same label of edge $(n_2, m_2)$ of $G_2$. Each clique in the auxiliary graph corresponds to a common subgraph of $G_1$ and $G_2$ and vice versa. Therefore, the maximum common subgraph of $G_1$ and $G_2$ can be obtained by finding the maximum clique in the auxiliary graph. A common subgraph of a set $S = \{G_1, \ldots, G_n\}$ of graphs may be obtained by the pairwise repeated application of the algorithm of Durand-Pasari to the graphs in $S$. In this case, we denote by $\mathrm{MaxSub}(S)$ the common subgraph so obtained.

**Definition 10 (Common supergraph)** *Let $G_1 = (V_1, E_1, \mu_1, \upsilon_1)$ and $G_2 = (V_2, E_2, \mu_2, \upsilon_2)$ be two labeled graphs. A graph $G$ is a common supergraph of $G_1$ and $G_2$ if and only if there is a subgraph isomorphism from $G_1$ to $G$ and another from $G_2$ to $G$.*

**Definition 11 (Minimum common supergraph)** *A common supergraph $G$ of $G_1$ and $G_2$ is minimum if all other common supergraphs of $G_1$ and $G_2$ have at least $\#(G)$ vertices.*

Bunke et al. (2000) proposed an exact algorithm to compute the minimum common supergraph of two graphs $G_1$ and $G_2$, whose output is denoted by $\text{MinSup}(G_1, G_2)$. In fact, they showed that the minimum common supergraph problem can be solved by maximum common subgraph computations. A common (but not necessarily minimum) supergraph of a set $S = \{G_1, \ldots, G_n\}$ of graphs may be obtained by the pairwise repeated application of the algorithm of Bunke to the graphs in $S$. In this case, we denote by $\text{MinSup}(S)$ the common supergraph so obtained.

In this work, we use the *graph edit distance* to measure the similarity between two graphs, as proposed by Bunke (1997) and used by Ferrer (2008). Vertex insertions or deletions have a unit cost, while edge insertions or deletions have a null cost. The substitution of a vertex by another has a null cost if both have the same attribute, otherwise the substitution cost is assumed to be arbitrarily large. The same applies to the substitution cost of an edge by another. Bunke et al. (2000) have shown that, under this cost function, there is always an optimal path between two given graphs that involves only vertex and edge deletions, insertions, or substitutions with identical attributes.

Ferrer (2008) showed that the graph edit distance between two graphs $G_1$ and $G_2$ under the above operation costs can be computed as

$$d(G_1, G_2) = \#(G_1) + \#(G_2) - 2 \cdot \#(\text{MaxSub}(G_1, G_2)),$$

where $\#(\text{MaxSub}(G_1, G_2))$ denotes the number of vertices in the maximum common subgraph of $G_1$ and $G_2$ obtained by the algorithm of Durand-Pasari.

Let $S = \{G_1, \ldots, G_n\}$ be a set of labeled graphs defined over an alphabet $L$. The sum of distances from a graph $G$ to $S$ is given by

$$\text{SOD}(G, S) = \sum_{i=1}^{n} d(G, G_i),$$

where $d(G, G_i)$ is the graph edit distance between $G$ and $G_i$, for $i = 1, \ldots, n$. Therefore, a generalized median graph of $S$ is given by

$$\overline{G} = \text{argmin}\{\text{SOD}(G, S) : G \in U\},$$

where $U$ is the set of all labeled graphs defined over the alphabet $L$. The generalized median graph $\overline{G}$ of $S$ satisfies

$$\#(\text{MaxSub}(S)) \leq \#(\overline{G}) \leq \#(\text{MinSup}(S)).$$

Furthermore, let $G'$ be the graph induced in $\text{MinSup}(S)$ by a subset $V'$ of its vertices. Let $G''$ be any subgraph of $G'$ with the same vertex set $V'$. Ferrer (2008) showed that $\text{SOD}(G', S) \leq \text{SOD}(G'', S)$. This result naturally suggests to consider the search space for the generalized median graph of $S$ to be formed by all subgraphs that can be induced in $\text{MinSup}(S)$ having no fewer vertices than $\text{MaxSub}(S)$. Both the adaptive greedy algorithm and the GRASP heuristic proposed in the next sections will begin by computing $\text{MinSup}(S)$ as their initial solution, followed by the evaluation of candidate solutions that will be subgraphs that can be induced in $\text{MinSup}(S)$.

# 3 Greedy adaptive algorithm

The computation of the edit distance $d(G, H) = \#(G) + \#(H) - 2 \cdot \#(\text{MaxSub}(G, H))$ between two graphs $G$ and $H$ is expensive, since it depends on the computation of their maximum common subgraph. Therefore, in the computation of the generalized median graph one is interested in avoiding maximum common subgraph computations whenever possible.

For the sake of explaining the evaluation of neighbors in the heuristics to be presented, we assume that the distance $d(G,H) = \#(G) + \#(H) - 2 \cdot \#(\text{MaxSub}(G,H))$ between two graphs $G$ and $H$ has been already computed. Let $G^{(-v)}$ be the graph obtained from $G$ by removing any of its vertices $v$. Musmanno (2013) has shown that if $v$ is not a vertex of $\text{MaxSub}(G,H)$, then $d(G^{(-v)},H) = d(G,H) - 1$. Otherwise, $d(G^{(-v)},H) = d(G,H) \pm 1$. In the last case, $d(G^{(-v)},H) = d(G,H) - 1$ if and only if $G$ and $H$ have another maximum common subgraph with the same number of vertices of $\text{MaxSub}(G,H)$ that does not contain $v$.

Let $\text{MinSup}(S)$ be the minimum common supergraph of the set of graphs $S = \{G_1, \ldots, G_n\}$. Since $\text{MinSup}(S)$ is a supergraph of each $G_i$, then

$$\text{MaxSub}(\text{MinSup}(S), G_i) = G_i$$

for each $i = 1, \ldots, n$. Furthermore,

$$\text{SOD}(\text{MinSup}(S), S) = \sum_{i=1}^{n} d(\text{MinSup}(S), G_i).$$

Let $\text{MinSup}(S)^{(-v)}$ be the graph obtained by removing a vertex $v$ from $\text{MinSup}(S)$ and $L^v = \{i = 1, \ldots, n : v \in \text{MaxSub}(G_i, \text{MinSup}(S))\}$. Assuming that the removal of vertex $v$ increases the edit distance by 1 for each $G_i : i \in L^v$, we obtain the following estimate for $\text{SOD}(\text{MinSup}(S)^{(-v)}, S)$:

$$\overline{\text{SOD}}(\text{MinSup}(S)^{(-v)}, S) =$$

$$= \text{SOD}(\text{MinSup}(S), S) - (n - |L^v|) + |L^v| =$$

$$= \text{SOD}(\text{MinSup}(S), S) - n + 2 \cdot |L^v|.$$

A greedy algorithm for the generalized median graph problem is obtained by removing the vertices $v$ of $\text{MinSup}(S)$ one by one in the decreasing order of the estimates $\overline{\text{SOD}}(\text{MinSup}(S)^{(-v)}, S)$. The best solution is updated whenever the removal of a vertex improves the incumbent.

A greedy adaptive algorithm can be derived from this greedy algorithm. In this case, every time a vertex is removed from the current solution, all estimates are recomputed from the new solution and the next vertex to be removed will be the one with the smallest updated estimate. The algorithm stops when all candidate vertices have been considered and examined for elimination. The pseudo-code for this algorithm is presented in Figure 1.

# 4 GRASP heuristic

GRASP (Greedy Randomized Adaptive Search Procedure) (Feo and Resende, 1989, 1995) is a multi-start metaheuristic, in which each iteration consists of two phases: construction and local search. The construction phase builds a solution. We assume that if this solution is not feasible, then either a repair procedure is applied to achieve feasibility or a new attempt to build a feasible solution is made. Once a feasible solution is obtained, its neighborhood is investigated until a local minimum is found during the local search phase. The best overall solution is kept as the result. Literature surveys are presented by Festa and Resende (2009a); Resende and Ribeiro (2002, 2005, 2010, 2014). Extensive accounts of successful applications of GRASP are reported by Festa and Resende (2002, 2009b).

The greedy randomized adaptive algorithm used in the construction phase of the GRASP heuristic is an extension of the greedy adaptive algorithm in Figure 1, in which the node selected for elimination at each iteration is randomly chosen among those with the best SOD estimates, but not necessarily the best one. Let *RCL* be a restricted candidate list formed by all vertices $v$ in the current solution for which $\overline{\text{SOD}}(\text{CurrentSolution}^{(-v)}, S) \in [\text{S}_{\min}, \text{S}_{\min} + \alpha \cdot (\text{S}^{max} - \text{S}_{min})]$, where

$$\text{S}_{\min} = \min\{\overline{\text{SOD}}(\text{CurrentSolution}^{(-v)}, S) : v \in \text{Candidates}\},$$

```
begin GREEDY-ADAPTIVE
1   Set CurrentSolution ← MinSup(S);
2   Set CurrentSOD ← SOD(MinSup(S),S);
3   Let Candidates be the vertex set of CurrentSolution;
4   Compute the estimate SOD(MinSup(S)^(−v),S) for each vertex v of MinSup(S);
5   while Candidates ≠ ∅ do;
6       u ← argmin{SOD(CurrentSolution^(−v),S) : v ∈ Candidates};
7       Let Solution be the graph obtained by removing vertex u from CurrentSolution;
8       if SOD(Solution,S) < SOD(CurrentSolution,S) then
9           CurrentSOD ← SOD(Solution,S);
10          CurrentSolution ← Solution;
11      end-if;
12      Candidates ← Candidates \ {u};
13      Update SOD(CurrentSolution^(−v),S) for each vertex v of CurrentSolution;
14  end-while;
15  return CurrentSolution;
end GREEDY-ADAPTIVE.
```

Figure 1: Pseudo-code of the greedy adaptive algorithm.

$$S_{\max} = \max\{\overline{SOD}(\text{CurrentSolution}^{(-v)}, S) : v \in \text{Candidates}\},$$

and $\alpha \in [0,1]$ is a threshold parameter that controls the amounts of greediness and randomness in the algorithm. The case $\alpha = 0$ corresponds to a pure greedy algorithm, while $\alpha = 1$ is equivalent to a random construction. Experiments for tuning parameter $\alpha$ have been performed, using different values $\alpha = 0.05$, 0.1, 0.2, 0.3, 0.4, and 0.5. Since no significant difference was observed in the numerical results, in the next section we report numerical experiments performed with $\alpha = 0.1$.

The local search phase acts to iteratively improve the constructed solution, by considering the insertion of nodes belonging to MinSup($S$) but not to the solution constructed in the first phase.

## 5 Experimental results

The greedy adaptive algorithm and the GRASP heuristic have been implemented in C++ and compiled with the compiler gcc (TDM-2 mingw31) 4.1.1. All computational experiments have been carried out on an Intel i5 2.8 GHz quadcore processor with 4 GB of RAM memory running under Windows 7 Home.

Test problems have been extracted from the AIDS group of graphs from the IAM Graph Database Repository (Bunke, 2011; Research Group on Computer Vision and Artificial Intelligence, 2011). The algorithms were tested on 100 randomly chosen instances, divided into ten test sets. Each test set contains ten instances of the same size, where the size of an instance is characterized by the total number of vertices in the graphs within its input graph set $S$. We considered test sizes of 20, 40, 60, 80, 100, 120, 140, 160, 180 and 200 vertices all together.

The GRASP heuristic was run for 100 iterations for each instance. The solutions obtained by GRASP were at least as good as those found by the adaptive greedy algorithm for all test problems. Furthermore, GRASP found strictly better solutions than the adaptive greedy algorithm for 57 instances over all 100 test problems. The advantage of GRASP in terms of solution quality increases with the problem size: GRASP found strictly better solutions than the adaptive greedy algorithm for 35 out of the largest 50 instances with 120 to 200 vertices.

Table 1 displays, for each test set (formed by ten instances each), the average reductions between the solution values $\text{SOD}(\overline{G}, S)$ obtained by the adaptive greedy algorithm and by the GRASP heuristic with respect to the sum of distances $\text{SOD}(\text{MinSup}(S), S)$ from MinSup($S$) to $S$ and to the sum of distances

$\text{SOD}(\hat{G},S)$ from the median graph $\hat{G}$ to $S$, showing by how much the proposed heuristics are able to improve, respectively, the initial solution and the upper bound given by the median graph.

Table 1: Average improvement in the sum of distances of the best solution found by each algorithm, with respect to the sum of distances from $\text{MinSup}(S)$ and from the median graph $\hat{G}$ to $S$.

| | Adaptive greedy | | GRASP | |
|---|---|---|---|---|
| Vertices (total) | $\text{SOD}(\overline{G},S)/\text{SOD}(\text{MinSup}(S),S)$ (%) | $\text{SOD}(\overline{G},S)/\text{SOD}(\hat{G},S)$ (%) | $\text{SOD}(\overline{G},S)/\text{SOD}(\text{MinSup}(S),S)$ (%) | $\text{SOD}(\overline{G},S)/\text{SOD}(\hat{G},S)$ (%) |
| 20 | 55.02 | 8.15 | 55.02 | 8.15 |
| 40 | 66.39 | 10.82 | 67.22 | 12.99 |
| 60 | 73.18 | 11.31 | 73.77 | 13.25 |
| 80 | 77.04 | 11.29 | 78.48 | 13.65 |
| 100 | 79.63 | 13.60 | 80.13 | 15.66 |
| 120 | 80.71 | 12.43 | 81.03 | 13.88 |
| 140 | 84.36 | 11.73 | 84.80 | 14.09 |
| 160 | 83.24 | 13.93 | 83.62 | 15.39 |
| 180 | 85.20 | 13.16 | 85.61 | 15.50 |
| 200 | 85.19 | 15.36 | 85.70 | 18.31 |
| Average | 77.00 | 12.18 | 77.54 | 14.09 |

Figure 2 depicts the evolution of the best solution found along 100 GRASP iterations for instance i05.140 with a total of 140 vertices. It also illustrates by how much the local search is able to improve the solution built by the greedy adaptive algorithm in the GRASP construction phase at each iteration.

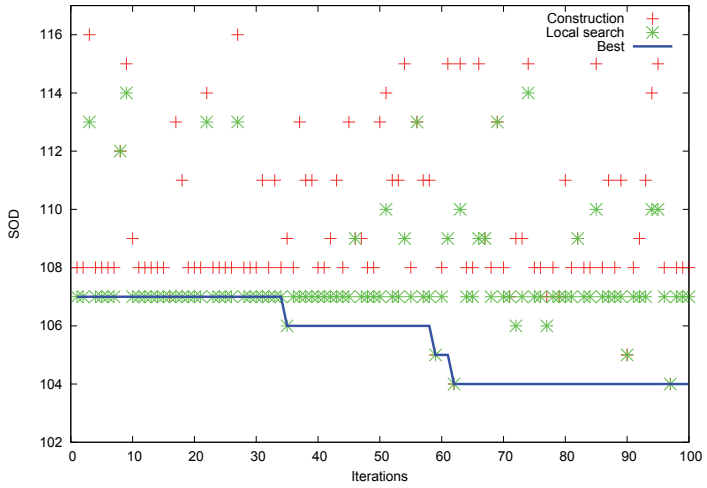


Figure 2: Evolution of the best solution found along 100 GRASP iterations for instance i05.140 with a total of 140 vertices.

## 6 Concluding remarks

Structural approaches for pattern recognition frequently make use of graphs to represent objects. The graph edit distance is often used to measure the similarity between two graphs. The generalized median graph is a good representative of a set of labeled graphs under the graph edit distance.

We proposed two heuristics for computing generalized median graphs: an adaptive greedy algorithm and its extension to a GRASP heuristic. The GRASP heuristic obtained generalized median graphs that

significantly improved the initial solutions and those provided by the median graphs. On average, the GRASP heuristic improved the sum of distances from the minimum common supergraph by 77.54% and the sum of distances from the median graph by 14.09%.

Nearest-neighbor classifiers are supervised learning tasks based on a training set of patterns. In this training set, every pattern has a category label assigned to it. Given a test set composed of all patterns to be classified, each of its patterns is compared to all elements of the training set and labeled with the class of the most similar element. The 1-nearest-neighbor classifier (1-NN) is defined by assigning a test pattern to the class of its most similar training pattern. The 1-NN classifier can be extended to consider not only the most similar pattern in the training set but, instead, the $k$ closest patterns: in a $k$-NN classifier, the test pattern is assigned to the class that occurs most frequently among its $k$ nearest or closest training patterns (Ferrer, 2008; Fukunaga, 1990).

We performed classification experiments classifying some queries using two approaches: the $k$-NN classifier and generalized median graphs. The latter amounts to computing an approximate generalized median graph of each class and comparing each query to these approximate generalized median graphs. The median graph approach presents the advantage that the number of comparisons is greatly reduced, since each query is compared only to a small number of graphs, while with $k$-NN the query is compared with every element of all classes. The instances used in this experiment were taken from the IAM Graph Database Repository (Research Group on Computer Vision and Artificial Intelligence, 2011). Numerical results that will be reported in the final version of this paper show that the approach based on the generalized median graph is competitive with the $k$-NN classifier, leading to smaller computation times (in particular, when the size of the training set increases) and similar accuracy levels.

These results lead to the main conclusions of this work. Good approximations to the generalized median graph can be effectively computed by the heuristics proposed in this paper, making it a better representation than the median graph to be used in a number of relevant pattern recognition or machine learning applications.

## References

Balas, E. and Yu, C. S. (1986). Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15:1054–1068.

Berretti, S., Bimbo, A. D., and Vicario, E. (2001). Efficient matching and indexing of graph models in content-basedretrieval. *IEEE Transactions Pattern Analysis Machine Intelligence*, 23:1089–1105.

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

Bunke, H. (2011). Graph representation for intelligent information processing - Fundamentals and algorithms for classification and clustering. Online reference available at http://cvpr-ss-2010.cecs.anu.edu.au/pdfs/HorstBunke.pdf, last visited on February 2, 2015.

Bunke, H., Foggia, P., Guidobaldi, C., Sansone, C., and Vento, M. (2002). A comparison of algorithms for maximum common subgraph on randomly connected graphs. *Lecture Notes in Computer Science*, 2396:123–132.

Bunke, H., Jiang, X., and Kandel, A. (2000). On the minimum common supergraph of two graphs. *Computing*, 65:13–25.

Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18:265–298.

Conte, D., Foggia, P., and Vento, M. (2007). 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*, 11:99–143.

de la Higuera, C. and Casacuberta, F. (2000). Topology of strings: Median string is NP-complete. *Theoretical Computer Science*, 230:39–48.

Durand, P., Pasari, R., Baker, J. W., and Tsai, C.-C. (1999). An efficient algorithm for similarity analysis of molecules. *Internet Journal of Chemistry*, 2(17). Online reference available at http://www.cs.kent.edu/∼jbaker/paper, last visited on February 2, 2015.

Fan, K. C., Liu, C. W., and Wang, Y. K. (1998). A fuzzy bipartite weighted graph matching approach to fingerprint verification. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4363–4368, San Diego. IEEE.

Feo, T. and Resende, M. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71.

Feo, T. and Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.

Ferrer, M. (2008). *Theory and Algorithms on the Median Graph - Application to Graph-based Classification and Clustering*. PhD thesis, Universitat Autonoma de Barcelona, Belaterra.

Ferrer, M., Valveny, E., and Serratosa, F. (2009). Median graphs: A genetic approach based on new theoretical properties. *Pattern Recognition*, 42:2003–2012.

Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., and Bunke, H. (2008). An approximate algorithm for median graph computation using graph embedding. In *19th International Conference on Pattern Recognition*, volume 2, pages 1–4, Tampa.

Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., and Bunke, H. (2010). Generalized median graph computation by means of graph embedding in vector spaces. *Pattern Recognition*, 43:1642–1655.

Festa, P. and Resende, M. (2002). GRASP: An annotated bibliography. In Ribeiro, C. and Hansen, P., editors, *Essays and Surveys in Metaheuristics*, pages 325–367. Kluwer.

Festa, P. and Resende, M. (2009a). An annotated bibliography of GRASP, Part I: Algorithms. *International Transactions in Operational Research*, 16:1–24.

Festa, P. and Resende, M. (2009b). An annotated bibliography of GRASP, Part II: Applications. *International Transactions in Operational Research*, 16:131–172.

Fischer, S., Gilomen, K., and Bunke, H. (2002). Identification of diatoms by grid graph matching. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 94–103, London. Springer.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York.

Hlaoui, A. and Wang, S. (2003). A new median graph algorithm. *Lecture Notes in Computer Science*, 2726:225–234.

Hong, P., Wang, R., and Huang, T. (2000). Learning patterns from images by combining soft decisions and hard decisions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 79–83, Hilton Head. IEEE Computer Society.

Jiang, X., Munger, A., and Bunke, H. (2001). On median graphs: Properties, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1144–1151.

Lades, M., Vorbruggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., P., R., Wurz, and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311.

McGregor, J. J. (1982). Backtrack search algorithms and the maximal common subgraph problem. *Software Practice and Experience*, 12:23–34.

Mukherjee, L., Singh, V., Peng, J., Xu, J., Zeitz, M., and Berezney, R. (2007). Generalized median graphs: Theory and applications. In *Proceedings of the IEEE 11th International Conference on Computer Vision*, Rio de Janeiro. IEEE. online reference available at http://www.biostat.wisc.edu/∼vsingh/mediang.pdf, last visited on February 2, 2015.

Mukherjee, L., Singh, V., Peng, J., Xu, J., Zeitz, M. J., and Berezney, R. (2009). Generalized median graphs and applications. *Journal of Combinatorial Optimization*, 17:21–44.

Musmanno, L. (2013). Approximate algorithms for the generalized median graph problem (in Portuguese). Master's thesis, Universidade Federal Fluminense, Niterói.

Rebagliati, N., Solé-Ribalta, A., Pelillo, M., and Serratosa, F. (2012). On the relation between the common labelling and the median graph. In *Proceedings of the 2012 Joint IAPR International Conference on Structural, Syntactic, and Statistical Pattern Recognition*, pages 107–115. Springer.

Research Group on Computer Vision and Artificial Intelligence (2011). IAM graph database repository. online reference at http://www.iam.unibe.ch/fki/databases/iam-graph-database, last visited on February 2, 2015.

Resende, M. and Ribeiro, C. (2002). Greedy randomized adaptive search procedures. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer.

Resende, M. and Ribeiro, C. (2005). GRASP with path-relinking: Recent advances and applications. In Ibaraki, T., Nonobe, K., and Yagiura, M., editors, *Metaheuristics: Progress as Real Problem Solvers*, pages 29–63. Springer.

Resende, M. and Ribeiro, C. (2010). Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, pages 283–319. Springer, 2nd edition.

Resende, M. and Ribeiro, C. (2014). GRASP: Greedy Randomized Adaptive Search Procedures. In Burke, E. and Kendall, G., editors, *Search Methodologies*, chapter 11, pages 285–310. Springer, 2nd edition.

Shearer, K., Bunke, H., and Venkatesh, S. (2001). Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34:1075–1091.

Suganthan, P. N. and Yan, H. (1998). Recognition of handprinted chinese characters by constrained graph matching. *Image and Vision Computing*, 16:191 – 201.

Torsello, A. and Hancock, E. R. (2003). Computing approximate tree edit-distance using relaxation labeling. *Pattern Recognition Letters*, 24:1089–1097.

Voigt, K. (2010). Semi-automatic matching of heterogeneous model-based specifications. *Lecture Notes in Informatics*, P-160:537–542. Online reference at http://subs.emis.de/LNI/Proceedings/Proceedings160/article5479.html, last visited on February 2, 2015.

Zeng, Z., Tung, A. K. H., Wang, J., Feng, J., and Zhou, L. (2009). Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2:25–36.