

# Step-by-Step Calculation of All Maximum Common Substructures through a Constraint Satisfaction Based Algorithm

Gonzalo Cerruela García, Irene Luque Ruiz,\* and Miguel Ángel Gómez-Nieto

Department of Computing and Numerical Analysis, University of Córdoba, Campus Universitario de Rabanales, Building C2, Plant 3, E-14071 Córdoba, Spain

Received August 2, 2003

In this paper we propose a new algorithm for subgraph isomorphism based on the representation of molecular structures as colored graphs and the representation of these graphs as vectors in  $n$ -dimensional spaces. The presented process that obtains all maximum common substructures is based on the solution of a constraint satisfaction problem defined as the common  $m$ -dimensional space ( $m \leq n$ ) in which the vectors representing the matched graphs can be defined.

## 1. BACKGROUND

Graphs are data structures universally used for the representation of many and varied problems in different areas of the sciences, engineering and humanities. If objects or elements of a certain domain of a given problem are represented by means of graphs, one has the possibility of comparing these graphs in order to detect the set of common properties of these elements. This process is known as graph isomorphism or graph homomorphism.<sup>1</sup>

For the last fifty years many algorithms for finding if two graphs  $G_1$  and  $G_2$  are equal have been proposed. These algorithms, as proposed by Ullmann,<sup>2</sup> are efficient when the graphs are highly different, diminishing in efficiency as the graph similarity increases.

The problem becomes more complicated when the purpose is to find what subgraph of the  $G_1$  graph exists in the  $G_2$  graph, known as subgraph isomorphism.<sup>3,4</sup> The detection of subgraph isomorphism, and especially the detection of the maximum common substructures (MCS), has been applied to a great variety of fields. Chemistry has paid more attention to the solution of similarity problems,<sup>5</sup> design and synthesis of products,<sup>6</sup> studies of NMR and of mass spectra,<sup>7</sup> QSPR and QSAR,<sup>8</sup> etc.

A graph  $G$  is called a maximum common subgraph (MCS) of two graphs  $G_1$  and  $G_2$ , if there is no other common subgraph of  $G_1$  and  $G_2$  that has more nodes than  $G$  (*node induced subgraph*).<sup>9</sup> This is in contrast with other definitions<sup>10</sup> where a MCS of two given graphs is defined as the common subgraph which contains the maximum number of edges (*edge induced subgraph*). Notice that according to this definition MCS is not necessarily unique for two given graphs. Although there are other considerations, this paper considers that MCS is a connected graph, and therefore MCS is also called the maximum clique in  $G$ .

The main problem in the detection of subgraph isomorphism is the fact that it is an NP-complete problem.<sup>11</sup> The reduction of this computational cost is still a challenge for

researchers for two main reasons: (a) the applicability of these algorithms to any research area and (b) the huge increase of information required for any problem (for example, the size of the databases of chemical compounds)—and needs the development of efficient algorithms that allows the comparison of thousands or millions of graphs within an acceptable time.

Different techniques have been proposed to reduce the computational cost corresponding to subgraph isomorphism: decomposition of the graph into trees and use of backtracking algorithms, forward-checking, looking-ahead, etc.,<sup>12</sup> optimization techniques involving the use of neural networks, genetic algorithms,<sup>13</sup> simulated annealing,<sup>14</sup> and so on.<sup>15–18</sup>

TopSim<sup>6</sup> implements an algorithm based on the use of relational line graphs representing the graphs to be compared. From the line graph representation, the algorithm is based on the calculation of a compatibility graph that represents all the possible relationships among the nodes of  $LG_1$  and  $LG_2$  graphs. Later on, the algorithm generates a hierarchical structure representing all the possible cliques. Finally, a depth-wide search algorithm is used, and the maximum clique among the graphs is obtained.

Recently, a new algorithm—named *Rascal*—concerned with the calculation of the maximum common edge subgraph has been proposed.<sup>18</sup> This algorithm is based on both the representation of the  $G_1$  and  $G_2$  graphs, as line graphs  $LG_1$  and  $LG_2$ , and on carrying out the modular product between these line graphs in order to obtain the maximum common edge subgraph (MCES) or the maximum overlapping set (MOS). This efficient algorithm is applied to the calculation of similarities among molecular graphs by the introduction of some heuristics based on the composition of these graphs (linear chains, aromatic rings, etc.) to reduce the computational cost of the algorithm.

These heuristics are guided to obtain (in preprocessing time) information on the existence of cycles and chains with the objective of diminishing the size of the modular product. The authors demonstrate that the use of these heuristics diminishes the cost of the modular product.<sup>18</sup> However, the

\* Corresponding author phone: +34-957-21-2082; fax: +34-957-21-8630; e-mail: mailurui@uco.es.

preprocessing cost should be high since it is necessary to obtain all the cycles, chains, paths up to a certain size, numeration of the path and cycles, etc.

In principle, the algorithms described in the literature are based on two different principles:<sup>9</sup> (a) algorithms based on searching for the *MCS* by finding all common subgraphs of the two given graphs and choosing the largest and (b) algorithms based on building the association graph between the two given graphs and then searching for the maximum clique of the latter graph.

Based on this last principle, in this paper we describe an algorithm that deals with the calculation of similarity measures based on all maximum common substructures. The algorithm is based on the fulfillment of a set of constraints and obtaining the compatibility graph between two given graphs, extracting all maximum common substructures that satisfy the imposed restrictions.

Algorithms work without heuristic considerations in the preprocessing stage thanks to the use of the constraints satisfaction model on both the vectorial representation of the molecular graphs and the vectorial operations among the vectors representing the graphs to be compared. The proposed algorithm is conceptually simple, and it finds all maximum common substructures (*AMCS*) with low memory requirements and with a computation lineally dependent on  $O(nm)$ .

This paper is organized in the following way: in Section 2 definitions of the used terminology and examples are presented in order to clarify the manuscript content. In Section 3 the theoretical foundation of the proposed algorithm and the model of representation of the molecular graph are described. Section 4 shows the characteristics and algorithm functionality and an example of its application. Last, section 5 is dedicated to the description of the performed tests with the proposed algorithm on a wide collection of chemical compounds, with a final discussion on obtained results.

## 2. INTRODUCTION AND DEFINITIONS

All maximum common subgraphs between two graphs  $G_1$  and  $G_2$  are a set of graphs which are maximum and common to  $G_1$  and  $G_2$ , obtained as follows:

1. *MCS* between the graphs  $G_1$  and  $G_2$  is obtained—*MCS*<sup>*l*</sup>—( $G_1, G_2$ )—

2. From the graphs  $G_1$  and  $G_2$  the *MCS*<sup>*l*</sup> previously obtained is erased.

3. Steps (1) and (2) are repeated over the resulting graphs of step 2 until any new resulting graph is null or a new *MCS* does not exist.

Thus, *AMCS* is a set of maximum and common subgraphs to  $G_1$  and  $G_2$  and that

1. All the graphs of the *AMCS* set are connected graphs.  
2. Elements of the *AMCS* set are *MCS*<sup>*k*</sup> that maximize the *MCS*<sup>*k+1*</sup>.

3. As the *MCS* between two graphs  $G_1$  and  $G_2$  is not necessarily unique,<sup>9</sup> neither is the set of subgraphs common to  $G_1$  and  $G_2$  that maximize the matching (*MOS*).

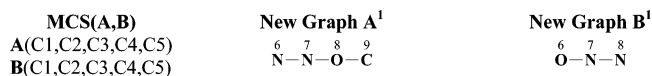
4. The graph composed of all the subgraphs of the *AMCS* set does not necessarily correspond to the *MOS* between the  $G_1$  y  $G_2$  graphs.

The following example in which the *A* and *B* graphs are compared clarifies this definition.



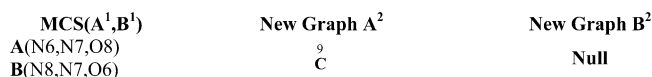
The subgraph of higher size common to the *A* and *B* graphs has five nodes. Chart 1 shows the *MCS* obtained and the new graphs *A*<sup>*l*</sup> and *B*<sup>*l*</sup> (after it is eliminated from the graphs *A* and *B* the *MCS* calculated).

Chart 1



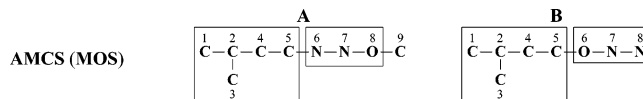
On the resulting graphs a new maximum common subgraph of size 3 can be obtained, as is shown in Chart 2 together with the new graphs *A*<sup>*2*</sup> and *B*<sup>*2*</sup> resulting from eliminating the new *MCS*(*A*<sup>*l*</sup>, *B*<sup>*l*</sup>) obtained.

Chart 2



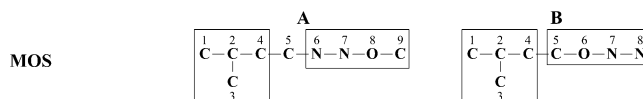
Between graphs *A*<sup>*2*</sup> and *B*<sup>*2*</sup> there is no new common subgraph. So, the *AMCS* is composed of the *MCS*(*A*, *B*) and *MCS*(*A*<sup>*l*</sup>, *B*<sup>*l*</sup>) subgraphs previously obtained and corresponding to the *MOS* between the graphs *A* and *B*, as shown in Chart 3.

Chart 3



However, although between graphs *A* and *B* there is no *MCS* of a greater size than 5, it is possible to obtain a different *MOS*, as shown in Chart 4. This *MOS* is composed of two subgraphs of size 4 that are common to graphs *A* and *B*.

Chart 4



The classic measurements of similarity obtained based on the *MOS* would give the same result independently of the size of the considered common subgraphs between *A* and *B*. However, for certain applications of screening/clustering in chemical databases, as described at the end of the manuscript, the size of these subgraphs is important.

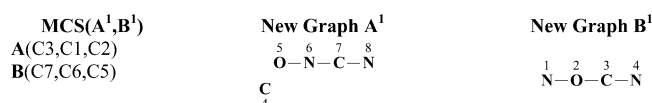
The following example presents a different case:



The chain formed by (C5,C6,C7) is the *MCS* corresponding to graph *B*, which can be matched in two different ways with graph *A*: (a) considering the A(C2) node and (b) without considering the A(C2) node, that will give different results.

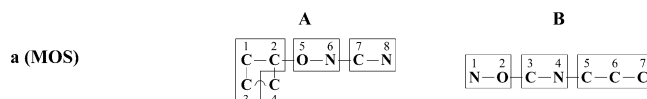
(a) **Considering the A(C2) Node.** Between the resulting graphs *A*<sup>*l*</sup> and *A*<sup>*2*</sup> two possible matchings of the same size

Chart 5



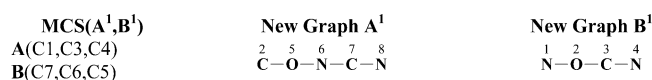
can be obtained, both giving the same result: (a) initially matching A(N6,O5) with B(N1,O2) and after A(C7,N8) with B(N4,C3) and (b) and vice versa, initially matching A(C7,N8) with B(N4,C3) and after A(N6,O5) with B(N1,O2), (other possible matching between A(N6,C7) and B(N4,C3) is possible, but as we will describe below this matching must be not considered). Chart 6 shows the result with an *MOS* of 7 nodes.

Chart 6



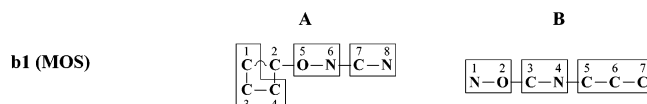
(b) **Without Considering the A(C2) Node.** Now, among the resulting graphs A<sup>1</sup> and B<sup>1</sup> it is possible to carry out three types of different matching:

Chart 7



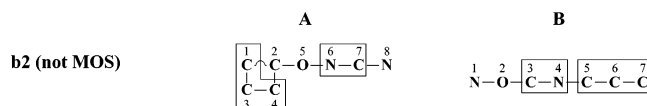
(b1) A matching between A(N8,C7) and B(N4,C3) and therefore the following matching A(O5,N6) and B(O2,N1): Chart 8 shows the result, obtaining again an *MOS* of 7 nodes.

Chart 8



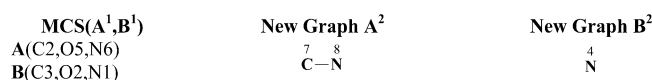
(b2) A matching between A(N6,C7) and B(N4,C3), which does not allow for carrying out new matching among the resulting graphs: Chart 9 shows the result, obtaining an overlapping of 5 nodes. Of course, this option must be not considered.

Chart 9



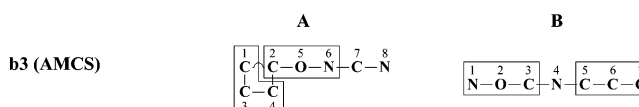
(b3) As the algorithm proposed in this paper works—selecting the *MCS*(A<sup>1</sup>,B<sup>1</sup>)—, that is, a matching between A(C2,O5,N6) and B(C3,O2,N1), there is no existing new matching between the resulting graphs A<sup>2</sup> and B<sup>2</sup>, as shown in the Chart 10.

Chart 10



So, the obtained *AMCS* of comparing graphs A and B is composed of two subgraphs of 3 nodes each, as shown in Chart 11. In this example the *AMCS* is composed of a set of maximum and common subgraphs to A and B, but this set is smaller than the *MOS* previously obtained in the other cases.

Chart 11



The exposed examples allow one to observe the following:

- The *MCS* between two graphs A and B is not unique.<sup>9</sup>
- Therefore neither is the *MOS* between two graphs A and B unique.
- The *AMCS* is composed of a set of common and maximum subgraphs to A and B. This set is not necessarily equal to the *MOS*.

Therefore, different measures of similarity can be obtained in the function of the used approach (*MCS*, *MOS*, *AMCS*). The use of these different approaches can be applied adequately for the development of applications in the organization and recovery of information in chemical databases.

In the following sections an algorithm is described with the aim of obtaining the set or *AMCS* step-by-step, so that all the *MCS* between two given graphs can be obtained.

### 3. FOUNDATIONS OF THE ISOMORPHISM MODEL USED

The developed algorithm is based on the representation of the molecular structures as colored graphs and the representation of these graphs as vectors in multidimensional spaces. The process to obtain all maximum common substructures is based in the solution of a constraints satisfaction problem<sup>12,19,20</sup> defined as the common *k*-dimensional space in which the corresponding vectors can be plotted.

**3.1. Characteristic of the Molecular Graph.** A molecular graph can be represented using a characteristics vector  $C_G$  whose elements or characteristics represent the different relationships existing among the molecular graph elements, that is to say, to each one of the one-depth subgraphs existing in the molecular graph.

Thus, given a molecular graph  $G = (V, E, F)$ , where  $V$  represents the vertexes of the graph,  $E$ , the edges and  $F$  a function that assigns the relationship type (edge) among each one of the vertexes  $v_i$  and  $v_j$ , this graph can be represented by means of its associated characteristics vector  $C_G$ , where each element represents a subgraph in  $G$  formed by two vertexes  $v_i$  and  $v_j$  and an edge  $e_{ij}$  characterized by  $t(v_i, v_j)$  function.

**3.2. Characteristic Relationships.** Two characteristics that represent the  $G$  graph are related if both share at least a vertex in  $G$ . The relationships set existing among the characteristics that represent the molecular graph can be written as:

$$CR_G = C_G \Psi \quad (1)$$

where  $CR_G$  is the characteristics relationships array (matrix),  $C_G$  is the array of characteristics that represents the  $G$  graph, and  $\Psi$  is a matrix of  $x_{ij}$  coefficients that describe the existence of a relationship between two characteristics  $cg_i$  and  $cg_j$ , taking value 1 if:

1. The characteristic  $cg_i$  has a vertex common to the characteristic  $cg_j$ ,

2. The characteristic  $cg_i = cg_j$ , and this characteristic is only related to one characteristic in the  $G$  graph, and 0 otherwise.

**3.3. Foundations of the Solution.** Given two chemical compounds  $A$  and  $B$  whose molecular topological structures are represented in  $n$  and  $m$  dimensional spaces through the characteristics and relationships vectors  $C_A$ ,  $CR_A$  and  $C_B$ ,  $CR_B$ , the common topological structure to both molecules can be obtained by finding a common  $k$ -dimensional space, with  $k \leq \min(n, m)$ , in which the common topological structures can be plotted by means of the representation in this space of each one of the one-depth molecular subgraphs present in the molecular graphs that represent molecules.

Representing the molecular structure under this model, the problem of finding the maximum common substructure between two molecular structures decreases to a simple vectorial problem and consists of finding a matching vector  $M$  defined in a space  $k$  that is maximum and common to  $n$  and  $m$ .

Also the proposed model makes it possible to find a set of vectors  $M_i$  representing a set of  $k_i$  disjoint spaces common to the vectors  $C_A$  and  $C_B$ , that allow knowing the maximum common substructure between two molecules, and also the set of all maximum common substructures.

Knowing the set of vectors  $M_i$ , it is possible to reduce the matching problem among the molecular structures to a simple chains matching problem and even to use a distance metrics of chains to propose structural similarity measures among the matched molecules.

#### 4. MATCHING OF CHARACTERISTICS. AN ALGORITHM BASED ON CONSTRAINT SATISFACTION

Given two chemical compounds  $A$  and  $B$ , represented in their  $n$  and  $m$  dimensional spaces through their corresponding characteristics and relationships vectors, as follows:

$$C_A = ac_1, ac_2, ac_3, \dots, ac_n \quad (2)$$

$$CR_A = C_A \Psi_A \quad (3)$$

$$C_B = bc_1, bc_2, bc_3, \dots, bc_m \quad (4)$$

$$CR_B = C_B \Psi_B \quad (5)$$

Graphs overlapping constraints are established, by which the graph isomorphism process will be based. This overlapping constraint can be enunciated as follows: “the highest  $k$ -dimensional space common to two multidimensional vectors  $A$  and  $B$  is that in which the highest number of characteristics and relationships representing the  $A$  and  $B$  graphs can be represented”. These constraints can be formulated in the following way:

**Constraint 1:** Two characteristics  $a_i \in C_A$  and  $b_j \in C_B$  represent the same space (dimension) if both are constituted by the same type of two colored nodes related by the same type of edge (the same one-depth subgraph).

**Constraint 2:** The common components of  $A$  and  $B$  characteristic arrays represented in the isomorphic  $k$ -dimensional space satisfy the previous constraint and preserve the relationships present in the corresponding  $CR_A$  and  $CR_B$  arrays.

**Constraint 3:** The isomorphic  $k$ -dimensional space must be maximum.

Thus, we have developed a constraint satisfaction algorithm that only makes use of binary array structures and vectorial arithmetic, with the following structure:

**Step 1:** A  $k$ -dimensional space common to  $C_A$  and  $C_B$  is obtained. In this space we can represent the characteristics of chemical compounds  $A$  and  $B$  through the cross product of corresponding characteristics vectors.

We can define a vector  $V = C_A \times C_B$  as an array of  $n \times m$  size as follows:

$$\begin{aligned} V &= x_1 ac_1 bc_1, x_2 ac_1 bc_2, \dots, x_m ac_1 bc_m \\ &\quad x_{m+1} ac_2 bc_1, \dots, x_{2m} ac_2 bc_m, \dots \\ &\quad x_{(n-1)m} ac_n bc_1, x_{(n-1)m+1} ac_n bc_2, \dots, x_{nm} ac_n bc_m \\ V &= C_A \times C_B \Phi \end{aligned} \quad (6)$$

where  $\Phi$  is a matrix of  $x_{ij}$  coefficients that will take the value of 1 if  $ac_i$  and  $bc_j$  characteristics satisfy the imposed constraint 1, and 0 otherwise.

$V$  is an array of dimension equal to  $k$ , where  $k \leq n \times m$ , and defining the possible total of characteristic combinations corresponding to the  $A$  and  $B$  graphs that satisfies the imposed constraint 1.

**Step 2:** Of the whole possible set of characteristics common to  $A$  and  $B$  graphs, it is necessary to find those that also have a maximum set of relationships with the remaining characteristics, that is, those characteristics defining a maximum hyperplane in the  $k$  common space.

Thus, for each one of the  $k$  elements  $(a_i b_j)$  of the  $V$  array—those characteristics of  $A$  and  $B$  graphs that satisfy the imposed constraint 1—the cross product of its components in the relationship arrays  $CR_A(ac_i)$  and  $CR_B(bc_j)$  is obtained. Thus, given a set of elements  $k$  in the  $V$  array:

$$\forall k \in V | k = (a_i b_j), VR = CR_A(a_i, a_p) \times CR_B(b_j, b_q) \quad (7)$$

In a general way, we can formulate as

$$VR = C_A \Psi_A \times C_B \Psi_B = C_A \times C_B \Omega \quad (8)$$

where  $VR$  represents the set of all possible relationships among the common characteristics to  $A$  and  $B$ , and  $\Omega$  is a matrix obtained as  $\Omega = \Psi_A \times \Psi_B$ , where each element  $\Omega(i, j)$  will take the value 1 if also the  $\Psi_A(i, j)$  and  $\Psi_B(k, l)$  elements are equal to 1, and 0 otherwise.

**Step 3:** We can find the common hyperplane (fulfillment of the constraint 2) in which we can represent the  $A$  and  $B$  graphs, by simply carrying out the dot product to the  $V$  and  $VR$  array, as follows:

$$H = V \cdot VR \quad (9)$$

Decreasing the problem to find those  $H$  subspaces that maximize the plotting of the  $A$  and  $B$  characteristics, the  $H$  array will have a  $k \times n \times m$  size, representing each one of the  $k_i$  dimensions, a space in which there exists an isomorphism between  $A$  and  $B$  graphs.

**Step 4:** The following processes are performed:

*Step 4.1:* The first  $k_i$  dimension of the  $H$  array is located, and a matching vector  $M_1$  is created with the elements of the  $H$  array.



Step 4.2: The following  $k_i$  dimension of the  $H$  array is located, and

1. If some of the  $C_A$  or  $C_B$  characteristics of the new elements are present in the  $M_i$  array, these elements are added to  $M_i$ .

2. Otherwise a new matching array  $M_2$  is created with the new element of the  $H$  array.

Step 4.3: Step 4.2 is repeated until all the elements of the  $H$  array are analyzed. Each new analyzed element of  $H$  is added to the matching arrays  $M_i$  if some of their  $C_A$  or  $C_B$  characteristics are present, otherwise a new matching array is created and the  $H$  element is added to the new matching array generated.

#### 4.1. Obtaining the Maximum Common Substructure.

Once this step has been taken, there will have been generated a series of  $M_i$  matching arrays containing all the possible common subspaces among  $A$  and  $B$  graphs. Although, when more than one common characteristic among  $A$  and  $B$  present is equal—representing the same one-depth subgraph—then in the  $M_i$  arrays there appears more than one possible solution to the isomorphism. Then:

**Step 5:** The following step consists of eliminating the redundancies in the  $M_i$  matching arrays:

Step 5.1: Each one of the  $M_i$  arrays is analyzed checking the redundant  $C_A$  or  $C_B$  characteristics (appearing more than once), proceeding with Step 5.2 in an affirmative case.

Step 5.2: For each  $C_A$  or  $C_B$  redundant characteristic, the number of occurrences of this characteristic is analyzed in  $H$ .

Step 5.3: Matching array element whose characteristics  $C_A$  or  $C_B$  have a smaller number of appearances in  $H$  is eliminated.

Once this process has ended, the matching arrays  $M_i$  contain all the possible matching among the characteristics that represent  $A$  and  $B$  graphs.

**Step 6:** Last, the number of elements of the matching arrays  $M_i$  are analyzed, to find the arrays  $\overline{M_i}$  with the highest number of elements representing the maximum common substructure (MCS) among  $A$  and  $B$  compounds.

#### 4.2. Obtaining All Maximum Common Substructures.

Although  $\overline{M_i}$  vector with the highest number of elements contains the maximum common substructure among  $A$  and  $B$  compounds, it is evident that the rest of the matching vectors contain information of other common substructures (not maximum).

Thus, it is convenient to extract this information with the purpose of being able to obtain “finer” measures of structural similarity among the  $A$  and  $B$  topological structures. For this purpose we carry out the following:

**Step 7:** Once selected the  $\overline{M_i}$  matching arrays containing the maximum common substructure, the following processes are carried out:

Step 7.1: All elements from the  $M_i$  arrays containing  $C_A$  or  $C_B$  characteristics present in  $\overline{M_i}$  are deleted.

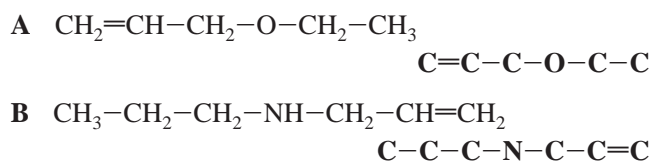
Step 7.2: Step 6 is performed in order to obtain a new  $\overline{M_i}$  with the next maximum common substructure.

Step 7.3: Steps 7.1 and 7.2 are repeated until all the  $M_i$  arrays are analyzed, or the  $M_i$  arrays are empty.

Once the process is concluded, a set of  $\overline{M_i}$  arrays is generated, containing the information corresponding to all maximum common substructures between the  $A$  and  $B$  compounds.

#### 4.3. Example of Application of the Proposed Algorithm.

To clarify the algorithm proposed for the obtaining of all maximum common substructures between two  $A$  and  $B$  graphs, we describe an example that, although simple for better clarity of the explanation, shows the simplicity and potential of the proposed algorithm.



These are represented through their characteristic and relationships vectors as follows:

$$C_A = \text{C}=\text{C}, \text{C}-\text{C}, \text{C}-\text{O}, \text{O}-\text{C}, \text{C}-\text{C}$$

$$CR_A = (\text{C}=\text{C})(\text{C}=\text{C}), (\text{C}=\text{C})(\text{C}-\text{C}), (\text{C}-\text{C})(\text{C}-\text{O}), \\ (\text{C}-\text{O})(\text{O}-\text{C}), (\text{O}-\text{C})(\text{C}-\text{C}), (\text{C}-\text{C})(\text{C}-\text{C})$$

$$C_B = \text{C}-\text{C}, \text{C}-\text{C}, \text{C}-\text{N}, \text{N}-\text{C}, \text{C}-\text{C}, \text{C}=\text{C}$$

$$CR_B = (\text{C}-\text{C})(\text{C}-\text{C}), (\text{C}-\text{C})(\text{C}-\text{C}), (\text{C}-\text{C})(\text{C}-\text{N}), \\ (\text{C}-\text{N})(\text{N}-\text{C}), (\text{N}-\text{C})(\text{C}-\text{C}), (\text{C}-\text{C})(\text{C}=\text{C}), \\ (\text{C}=\text{C})(\text{C}=\text{C})$$

For better clarity of the process we assign to each different characteristic a letter. So, the characteristic and relationships arrays are represented as follows:

$$C_A = a_1, a_2, a_3, a_4, a_5$$

$$C_B = b_1, b_2, b_3, b_4, b_5, b_6$$

$$CR_A = C_A (1100010100010100010100011)$$

$$CR_B = C_B (11000010100001010000101000011)$$

**Step 1:** Common  $k$ -dimensional space that satisfies the imposed constraint 1 is obtained:

$$V = C_A \times C_B \Phi = C_A C_B (000001110010000000000000110010)$$

**Step 2:** The common  $k$ -dimensional space of the relationships among the characteristics is obtained

$$VR = CR_A \times CR_B \Omega = C_A C_B \begin{pmatrix} 000011000011000000000000000000 \\ 110000000000110000000000000000 \\ 101000000000101000000000000000 \\ 000101000000001010000000000000 \\ 000000000000000000110000110000 \\ 000000000000000000101000101000 \\ 00000000000000000000101000101 \end{pmatrix}$$

**Step 3:** Common hyperspace to both representations is obtained:

$$\mathbf{H} = \mathbf{V} \bullet \mathbf{V}\mathbf{R} =$$

$$C_A C_B \begin{pmatrix} 000001000010000000000000000000 \\ 000000000000000000000000000000 \\ 000000000000000000000000000000 \\ 000001000000000000000000000000 \\ 0000000000000000000000000110000 \\ 0000000000000000000000000100000 \\ 000000000000000000000000000000 \end{pmatrix}$$

**Step 4:** Matching vectors are built:

*Step 4.1:*  $H$  array is visited and a matching array  $M_1$  is built with the first element of  $H$ , that is, with the first dimension  $k$  of the  $H$  array.

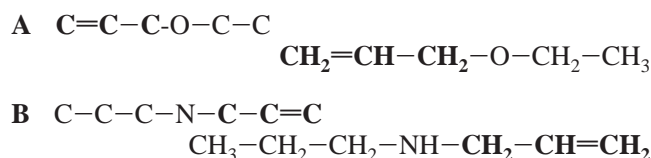
$$\mathbf{M}_1 = a_1 b_6, a_2 b_5$$

*Steps 4.2–4.3:* The remaining dimensions of  $H$  array are analyzed, and a new matching array is built:

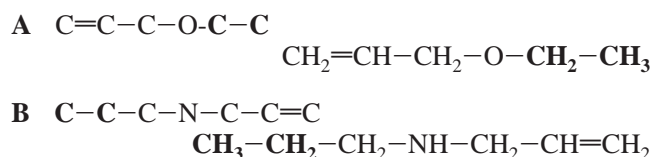
$$\mathbf{M}_2 = a_5 b_1, a_5 b_2$$

**Step 5:** Existence of redundancy is checked in the matching vectors. As is observed in  $M_2$  array,  $a_5$  characteristic appears duplicated in  $a_5 b_1$  and  $a_5 b_2$  elements. Components ( $a_5 b_1$ ) and ( $a_5 b_2$ ) are examined in  $H$  array, and the element ( $a_5 b_2$ ) is eliminated from  $M_2$  since this component occurs less in  $H$  than the ( $a_5 b_1$ ) component.

**Step 6:** The maximum common substructure corresponding to the  $M_1$  vector is obtained, as is observed in boldface characters:



**Step 7:** Remaining common substructures are obtained. In this case, only one new substructure exists, corresponding to the  $M_2$  matching vector, shown in boldface character:



**4.4. Algorithm Requirements.** The proposed algorithm is based on the realization of vectorial operations among the characteristic and relationships vectors representing  $A$  and  $B$  compounds.

Memory requirements are reduced to:

1. Two arrays  $C_A$  and  $C_B$ , to store the  $A$  and  $B$  characteristics, of size  $n$  and  $m$  respectively.
2. Two arrays  $CR_A$  and  $CR_B$ , to store the relationships among the  $A$  and  $B$  characteristics, of  $(n \times m)$  size.
3. An array of  $(n \times m)$  size to store  $V$ .

**Table 1.** Some of the Most Common Similarity Indices<sup>a</sup>

similarity coefficients ( $S_{A,B}$ )			
Tanimoto = $(c/a + b - c)$	Dice = $(2 \cdot c/a + b)$	Cosine (Ochiai) = $(c/\sqrt{a \cdot b})$	Rascal = $(c^2/a \cdot b)$

<sup>a</sup> Extracted from ref 21.  $a$ : number of nodes and edges in  $A$  graph,  $b$ : number of nodes and edges in  $B$  graph,  $c$ : common nodes and edges between  $A$  and  $B$  graphs.

4. An array of  $k \times (n \times m)$  size,  $k$  being the number of elements that satisfy the imposed constraint 1, to store  $\Omega$ . This space is used by the  $H$  array.

5. A set of  $M_i$  arrays of total size equal or smaller to  $(n \times m)$ .

As is observed, the memory space required by the algorithm is low, since the elements of the arrays only store 1 or 0. Furthermore, this size decreases considering only the elements up to the main diagonal of the  $CR$ ,  $V$  and  $H$  arrays, when these vectors are represented as matrices.

On the other hand the computational cost of the algorithm is directly dependent on the number of present characteristics in  $A$  and  $B$ .

## 5. RESULTS AND ALGORITHM BEHAVIOR

The new algorithm was implemented in C language, and the tests were performed on a PC with a Pentium IV processor and 128 Mb Ram memory and a Sun Enterprise 4000 with 8 processors UltraSparc II 250 MHz and 1.5 Gb of memory. In the test we have calculated different similarity indexes<sup>21</sup> (see Table 1) to (a) examine the similarity coefficients behavior and (b) test the algorithm results.

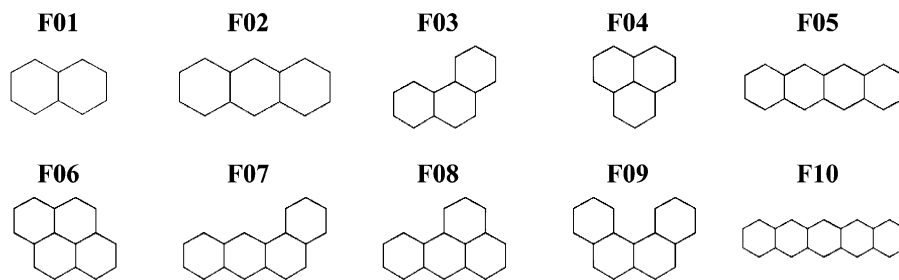
Initially, we checked the algorithm behavior with a well-known set of molecular structures, on which it is easy to validate the obtained results (see Figure 1). Some of the results corresponding to the all-maximum common substructures are shown in Tables 2 and 3. This information is obtained through adding all matching arrays obtained by the algorithm for each matching performed.

But one of the advantages of the developed algorithm is that each one of the maximum common substructures is obtained. The importance of this information is shown in the experiment carried out on the molecular structures in Figure 2.

In Figure 2, three series of molecular structures were considered. Each series corresponding to graphs which have (a) the same maximum common substructure, (b) a second maximum common substructure, (c) but the size of the not common substructure increases with the graphs in the series. Furthermore, as shown in Figure 2, the size of the redundant common substructures increases from series A to C.

In the tests,  $GI$  graphs are taken as patterns for each series, and the matching is performed with all the molecular structures of its corresponding series, calculating the following similarity measures for all the coefficients plotted in Table 1: (a) similarity considering only the maximum common substructure— $MCS$ — and (b) similarity considering all maximum common substructures— $AMCS$ —.

As shown on the left-hand side of Figure 3 the measure of similarity based on the  $MCS$  is very little affected by the noncommon subgraph size, and no existing remarkable differences between the coefficients are used. Thus, for a similar series, when increasing the size of the graph, very few changes are observed in the similarity measures if only



**Figure 1.** Some fused compounds used in the algorithm tests.

**Table 2.** Tanimoto and Dice Similarity Values for the Series of Figure 1

T/D	F01	F02	F03	F04	F05	F06	F07	F08	F09	F10
F01	<b>1.000</b>	0.700	0.700	0.750	0.538	0.600	0.538	0.568	0.538	0.438
F02	0.824	<b>1.000</b>	0.935	0.813	0.769	0.806	0.769	0.811	0.725	0.625
F03	0.824	0.967	<b>1.000</b>	0.871	0.725	0.857	0.769	0.763	0.769	0.592
F04	0.857	0.897	0.931	<b>1.000</b>	0.634	0.800	0.675	0.757	0.675	0.520
F05	0.700	0.870	0.841	0.776	<b>1.000</b>	0.762	0.950	0.810	0.902	0.776
F06	0.750	0.862	0.923	0.857	0.865	<b>1.000</b>	0.805	0.846	0.850	0.627
F07	0.700	0.870	0.841	0.806	0.974	0.892	<b>1.000</b>	0.900	0.902	0.776
F08	0.724	0.896	0.866	0.862	0.921	0.917	0.947	<b>1.000</b>	0.854	0.700
F09	0.700	0.841	0.870	0.806	0.949	0.919	0.974	0.921	<b>1.000</b>	0.740
F10	0.609	0.769	0.744	0.684	0.874	0.795	0.874	0.824	0.851	<b>1.000</b>

**Table 3.** Cosine and Rascal Similarity Values for the Series of Figure 1

C/R	F01	F02	F03	F04	F05	F06	F07	F08	F09	F10
F01	<b>1.000</b>	0.837	0.837	0.866	0.734	0.775	0.734	0.753	0.734	0.661
F02	0.700	<b>1.000</b>	0.967	0.897	0.877	0.895	0.877	0.900	0.848	0.791
F03	0.700	0.934	<b>1.000</b>	0.932	0.848	0.926	0.877	0.870	0.877	0.764
F04	0.750	0.805	0.868	<b>1.000</b>	0.787	0.894	0.817	0.870	0.817	0.709
F05	0.538	0.769	0.719	0.619	<b>1.000</b>	0.866	0.974	0.895	0.949	0.878
F06	0.600	0.747	0.857	0.744	0.750	<b>1.000</b>	0.893	0.917	0.920	0.781
F07	0.538	0.769	0.719	0.668	0.949	0.798	<b>1.000</b>	0.948	0.949	0.878
F08	0.568	0.811	0.758	0.757	0.849	0.841	0.898	<b>1.000</b>	0.921	0.831
F09	0.538	0.719	0.769	0.668	0.900	0.847	0.949	0.849	<b>1.000</b>	0.855
F10	0.438	0.625	0.584	0.503	0.771	0.648	0.771	0.690	0.731	<b>1.000</b>

the maximum common substructure (*MCS*) in this calculation is considered. Also, few differences are observed when the molecular structure size is changed (see the left-hand of Figure 3).

However, when we consider the *AMCS* in the similarity calculation, substantial changes are observed for the same and different series when the size of these parameters increases (see the right-hand side of Figure 3).

This behavior of the similarity measures taking into account the *MCS* or *AMCS* is shown more clearly in Figure 4, where the difference between the Tanimoto similarity measures obtained for the respective graphs of each series is represented.

As Figure 4 shows, maintaining a constant size of the maximum common substructure, the similarity measures using *MCS* are not very sensitive to the increment of the structure size (common or not common). However, when the *AMCS* is used, the similarity measures are very sensitive to the size increment of the noncommon substructures, as is appreciated in the slope of the curves.

This behavior can be conveniently utilized in computational chemistry, in certain processes, for instance, screening large databases, in those where the measure of similarity is used as selection criterion, and it is sometimes necessary

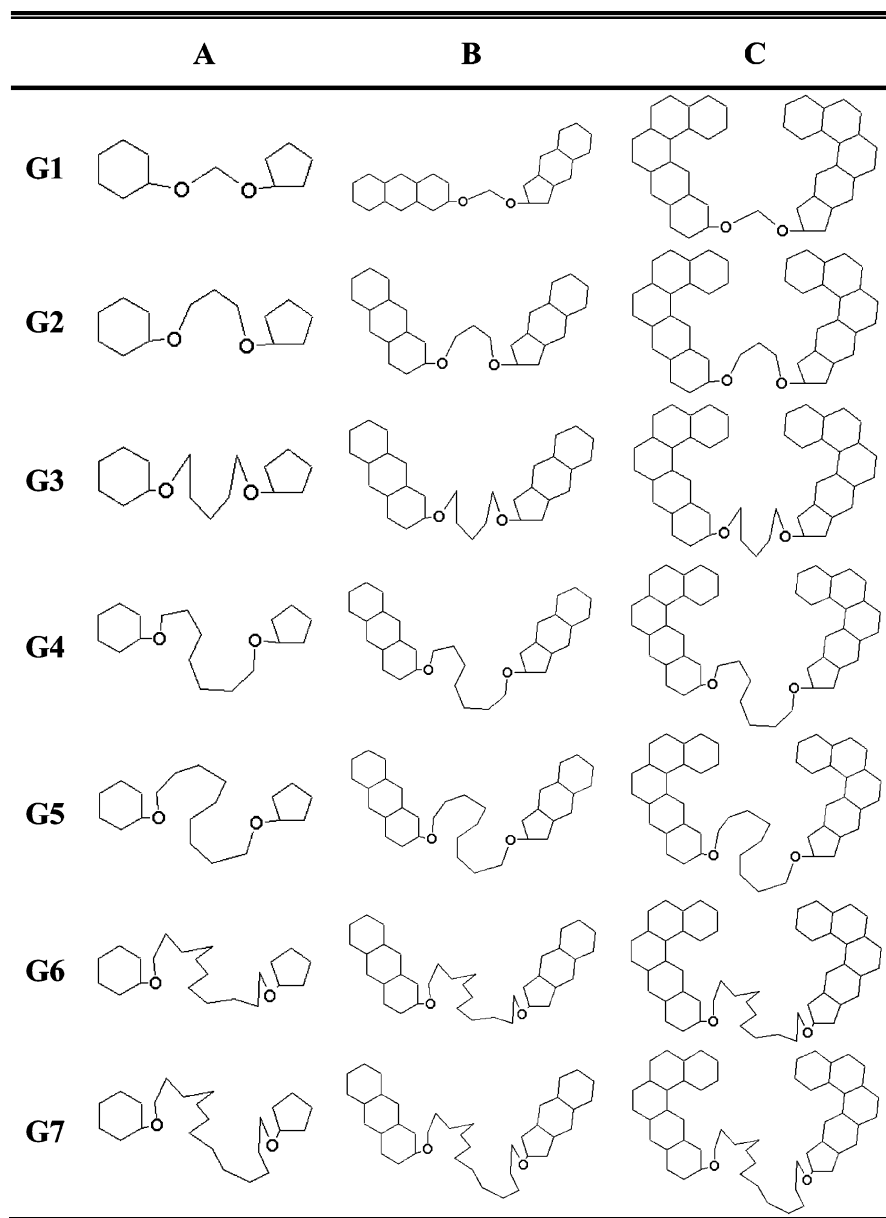
that these measures are not very sensitive to changes in the size of the graphs with regard to the substructure object of the search. In this context it would be possible to extract an acceptable set of molecular structures in which it is desirable to find a substructure (the maximum) involved in a graph problem.

On the other hand, similarity measures based on *AMCS* are convenient when trying to obtain a set of molecular structures in which most of the substructures object of the search are present.

The proposed algorithm allows for obtaining this information through the matching vectors  $M_i$  calculated in the steps 4 to 7. Thus, it is possible to obtain all the maximum common structures between two graphs, calculating different measures of similarity that can be suitable in search processes for substructure selection in chemical databases.

As the results represented in Figures 3 and 4 show, knowing the *AMCS* it is possible to obtain different measures of similarity making use of:

- Only the *MCS*.
- Using the *AMCS* set.
- Using only a subset of the *AMCS*, that is, only a subset of *MCS* existing in the *AMCS* (for example, until a certain size).



**Figure 2.** Three graphs series with the same MCS and different size.

Depending on the used approach the obtained values of similarity will be affected in greater or lesser measure by the size of the compared graphs and, mainly, by the size of the noncommon subgraphs.

And, depending on the objective, in the screening processes in chemical databases the use of one or the other approach will give better results in the recovery of molecules with common substructures (only one, several, of a given minimum size, etc.).

At the moment, we are carrying out a new clustering method of chemical databases using similarity measures calculated on the base of each one of the common substructures among graphs. These measures are used to represent each graph in an  $n$  dimensional array (a bin) with information of the compounds number in which substructures in a given similarity interval<sup>22</sup> are present.

An example of this application, for a reduced number of chemical compounds for simplicity and clarity reasons of the explanation, is shown in Table 4. For each matching process an array of similarity values is obtained. This array

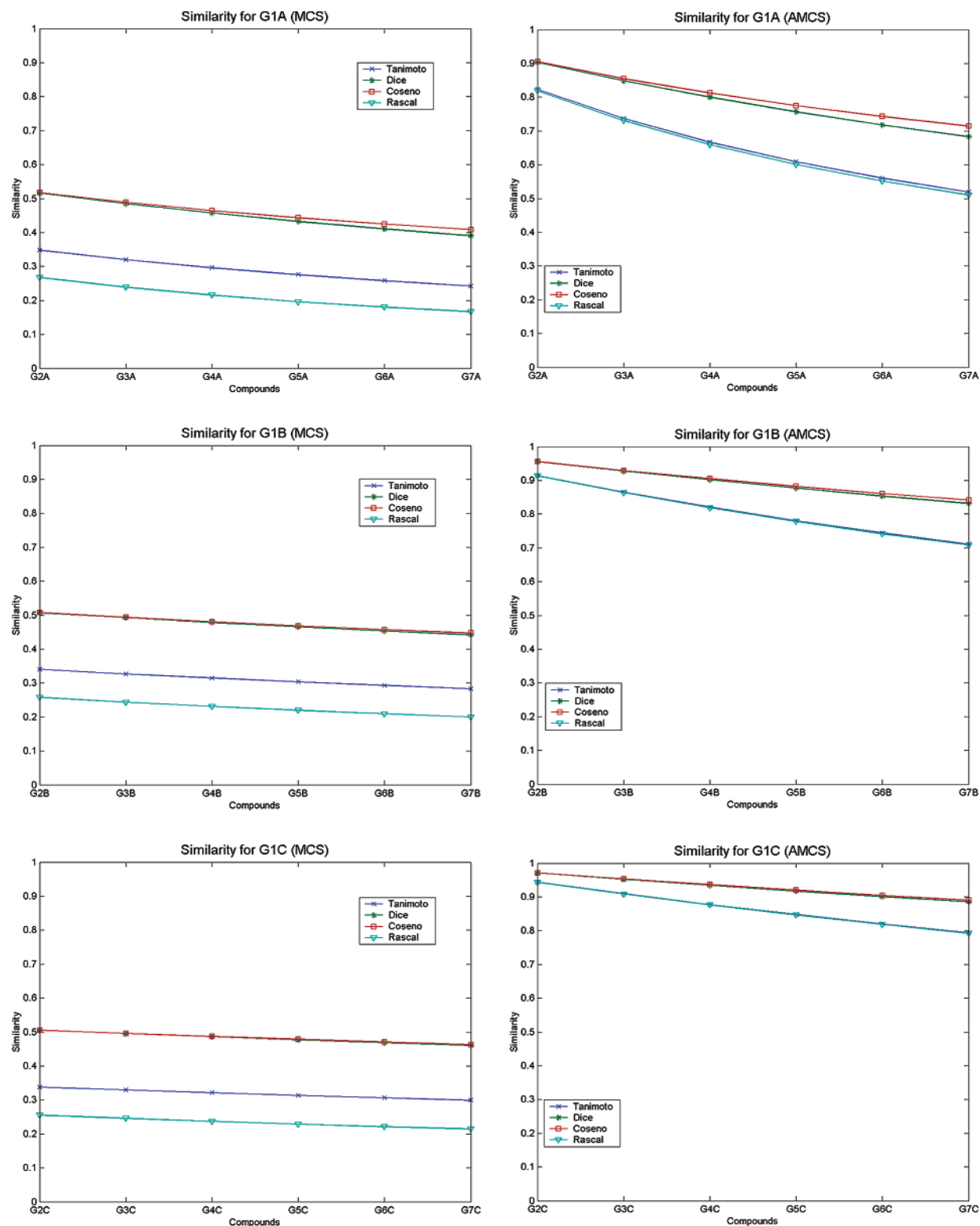
contains the similarity values obtained for each one of the maximum common substructures existing for the two matched compounds.

In the example shown in Table 4 the cosine index has been used for the calculation of the similarity. Thus, the total similarity (represented in boldface) corresponding to the *AMCS* can be obtained through the addition of the partial similarities.

Based on the information provided by the similarity bin it is possible to define a dynamic clustering model of database elements that assigns to the same cell compounds structurally similar (value of *AMCS*) and with a "similar" set of common substructures.

This mechanism avoids inconsistencies such as assigning the same class compound as the DBz0008, DBz0106, and DBz0178 that, although the similarity among these compounds is very close (see Table 4) in the case of the DBz0008 it is due to the presence of many substructures of small size, while in the case of the DBz0106 and the DBz0178 it is due to two common substructures.





**Figure 3.** Behaviors of similarity measures for the graphs series of Figure 2.

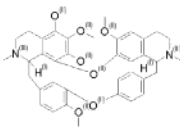
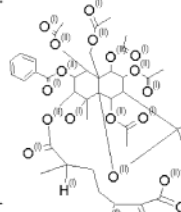
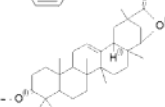
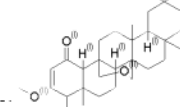
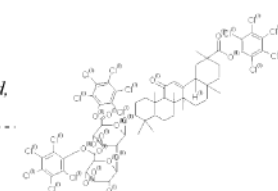
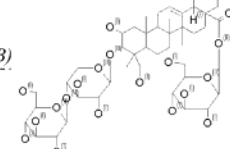
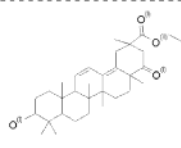
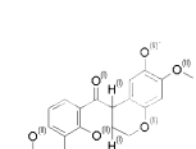
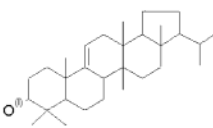
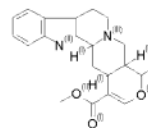
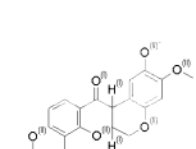
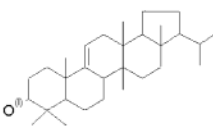
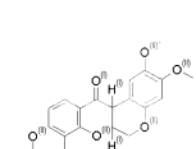
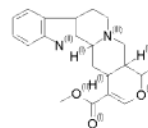
## 6. DISCUSSION

The calculation of the structural similarity is broadly used in computational chemistry. It is a complex problem that requires high computational costs when the size of the problem (graphs or molecular structures) increases.

Recently different algorithms have been proposed in order to reduce the computational cost of this calculation. Some of these algorithms make use of heuristics that allow for

reduction of this computational cost, although they introduce a new calculation cost to consider these heuristics, which is not mentioned in publications. These algorithms determine if two molecular structures are equal, the presence of a structure as part of the other, or all the common substructures between two graphs, that allows for obtaining similarity measures later used in computational chemistry and in the management of chemical databases.

**Table 4.** An Example of Performed Test over Chemical Databases<sup>a</sup>

	DBz0008	DBz0031	DBz0063	DBz0085	DBz0106	DBz0140	DBz0155	DBz0178	DBz0211	DBz0243
 <b>DBz0008</b> (Thaliximine)	0.178, 11, 10 0.161, 10, 9 0.093, 6, 5 0.059, 4, 3 0.042, 3, 2 0.025, 2, 1 0.025, 2, 1 0.025, 2, 1	0.239, 11, 10 0.125, 6, 5 0.080, 4, 3	0.229, 11, 10 0.142, 7, 6 0.076, 4, 3 0.055, 3, 2 0.033, 2, 1	0.154, 11, 11 0.133, 10, 9 0.126, 9, 9 0.021, 2, 1 0.021, 2, 1 0.021, 2, 1 0.021, 2, 1 0.021, 2, 1	0.168, 11, 10 0.088, 6, 5 0.088, 6, 5 0.072, 5, 4 0.056, 4, 3 0.056, 4, 3 0.056, 4, 3 0.024, 2, 1	0.586, 24, 25 0.036, 2, 1	0.253, 12, 11 0.099, 5, 4 0.099, 5, 4 0.033, 2, 1 0.033, 2, 1	0.227, 10, 9 0.132, 6, 5 0.060, 3, 2 0.036, 2, 1 0.036, 2, 1	0.407, 17, 16 0.136, 6, 5	
 <b>DBz0031</b> (Wilforin)	1.000, 50, 56	0.611, 40, 32	0.443, 21, 18	0.535, 27, 22	0.517, 40, 34	0.609, 42, 34	0.622, 26, 26	0.517, 26, 21	0.490, 23, 18	0.543, 23, 21
 <b>DBz0063</b> (Wilforlide)		0.378, 19, 18 0.112, 6, 5	0.305, 16, 15 0.167, 9, 8 0.029, 2, 1	0.270, 22, 21 0.107, 9, 8 0.057, 5, 4 0.044, 4, 3 0.031, 3, 2 0.031, 3, 2 0.019, 2, 1	0.281, 20, 19 0.166, 12, 11 0.065, 5, 4 0.036, 3, 2 0.022, 2, 1 0.022, 2, 1	0.355, 17, 16 0.075, 4, 3 0.075, 4, 3	0.346, 18, 17 0.168, 9, 8	0.312, 15, 14 0.118, 6, 5 0.032, 2, 1 0.032, 2, 1	0.344, 16, 15 0.100, 5, 4	
 <b>DBz0085</b> (Triterpene "R", derivative of)			0.593, 23, 22 0.066, 3, 2	0.501, 27, 24 0.396, 24, 23 0.042, 3, 2 0.025, 2, 1	0.560, 48, 41 0.370, 30, 29 0.048, 3, 2 0.025, 2, 1	0.613, 46, 39 0.570, 30, 29 0.101, 4, 3 0.043, 2, 1	0.505, 25, 22 0.331, 12, 11 0.066, 3, 2 0.040, 2, 1	0.514, 27, 25 0.543, 21, 20 0.066, 3, 2 0.043, 2, 1 0.043, 2, 1	0.495, 25, 21 0.619, 22, 21 0.043, 2, 1	0.444, 21, 19 0.520, 18, 17 0.045, 2, 1
 <b>DBz0106</b> (Glycyrrhizic acid, derivative of)				0.413, 25, 26 0.024, 2, 1 0.024, 2, 1 0.024, 2, 1	0.473, 26, 25 0.046, 3, 2 0.028, 2, 1 0.024, 2, 1	0.291, 11, 10 0.097, 4, 3 0.097, 4, 3 0.069, 3, 2	0.548, 22, 21 0.115, 5, 4 0.038, 2, 1	0.679, 25, 24	0.500, 18, 17 0.043, 2, 1	
 <b>DBz0140</b> (Phytolaccasaponin B)				1.000, 37, 42	0.486, 31, 29	0.547, 31, 28	0.554, 22, 18	0.701, 29, 26	0.679, 25, 24	0.543, 20, 18
 <b>DBz0178</b> (Meristotropic acid, derivative of)					0.416, 34, 36 0.137, 12, 11 0.053, 5, 4 0.042, 4, 3 0.030, 3, 2	0.284, 16, 16 0.097, 6, 5 0.027, 2, 1 0.027, 2, 1	0.367, 22, 23 0.204, 13, 12	0.470, 26, 27 0.044, 3, 2	0.338, 19, 18 0.027, 2, 1 0.027, 2, 1	
 <b>DBz0155</b>					1.000, 92, 101	0.677, 58, 56	0.434, 26, 23	0.571, 35, 35	0.514, 29, 29	0.93, 23, 20
 <b>DBz0211</b> (Ferneno)							0.213, 11, 10 0.112, 6, 5 0.071, 4, 3 0.071, 4, 3 0.030, 2, 1	0.626, 33, 34 0.028, 2, 1 0.051, 3, 2 0.030, 2, 1	0.508, 25, 25 0.071, 4, 3 0.030, 2, 1	0.325, 16, 15 0.115, 6, 5
 <b>DBz0243</b> (Tetrahydroalstonin)						1.000, 70, 77	0.497, 27, 22	0.654, 35, 35	0.660, 34, 31	0.440, 22, 20
 <b>DBz0155</b> (Degueli)								0.348, 13, 12 0.098, 4, 3 0.098, 4, 3	0.318, 11, 10 0.106, 4, 3 0.076, 3, 2	0.422, 14, 13 0.203, 7, 6
 <b>DBz0178</b>							1.000, 31, 35	0.544, 21, 18	0.500, 18, 15	0.625, 21, 19
 <b>DBz0211</b>								0.544, 19, 20 0.209, 8, 7	0.503, 18, 17 0.043, 2, 1	0.516, 17, 16
 <b>DBz0243</b>								1.000, 37, 41	0.753, 27, 27	0.546, 20, 18
									0.516, 17, 16	
									1.000, 31, 35	0.516, 17, 16
										1.000, 29, 33

<sup>a</sup> Cosine similarity index and nodes and common edges are reported for each maximum common substructure matched.

In our work we have developed an algorithm that, based on a constraints satisfaction model, allows for obtaining the whole previously aforementioned information.

The theoretical computational cost of the algorithm lineally depends on  $O(nm)$  ( $n$  and  $m$  being the number of characteristics of the matched graphs) that has been tested empirically as shown in Figure 5. For that, we decide to compare the  $G1A$  graph with the remaining graphs of Figure 2, which has allowed designing two types of experiments in which the pattern graph ( $G1A$ ) is maintained constant and the target graph increases in complexity ( $G2A-G7A$ ,  $G2B-G7B$  and  $G2C-G7C$ ):

1. When  $G1A$  graph is matched with the  $G1A$ ,  $G1B$  y  $G1C$  graphs:

- The common subgraph is maintained constant ( $G1A$ ).
- Only an  $MCS$  is present (therefore,  $AMCS=MCS$ ).
- The noncommon graphs increase progressively ( $G1A, G1A$ ) < ( $G1A, G2A$ ) < ( $G1A, G3A$ ).
- The number of redundancies increases in great measure since a high increase of rings of six nodes exists among the matched graphs.

2. When the  $G1A$  graph is matched with the remaining graphs of Figure 2:  $G1A$  vs ( $G2A-G7A$ ),  $G1A$  vs ( $G2B-G7B$ ) and  $G1A$  vs ( $G2C-G7C$ ):

- Two  $MCS$  exist, as shown in Chart 12, for which the  $AMCS$  is composed of two elements  $MCS^1$  and  $MCS^2$ .
- The noncommon subgraphs increase progressively but in a very inferior order to the previous case, since in this

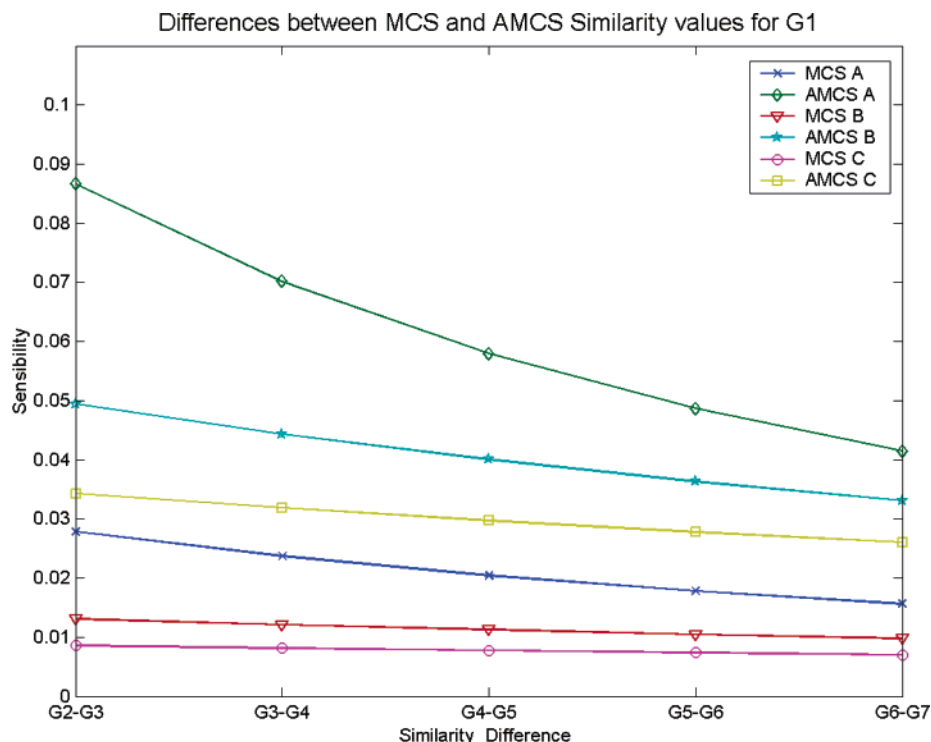


Figure 4. Difference of Tanimoto similarity measure obtained for the respective series of Figure 2.

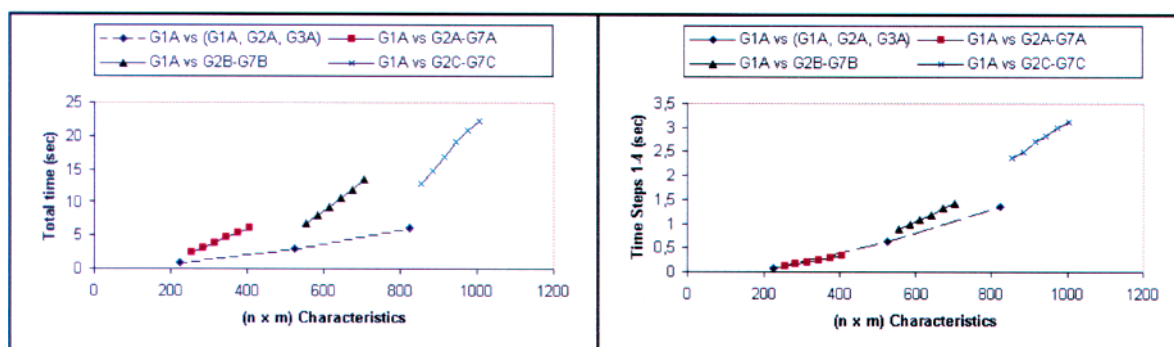
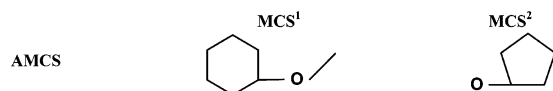


Figure 5. Behavior of the cost of the algorithm versus the product of number of characteristics for the matching of the *G1A* graphs with all graphs of the Figure 2.

#### Chart 12



case the increment is constant and equal to two noncyclic nodes (the difference among the *G2-G7* graphs of the three series), while in the previous case it was four (the difference between the *G1A* and *G1B* graphs) and eight (the difference between the *G1A* and *G1C* graphs) rings of six nodes.

- Although the number of redundancies increases, this increase is smaller (only two nodes) than in the previous case.

In Figure 5 the behavior of the total time and time of steps 1 to 4 of the algorithm for the two designed experiments has been represented, being proven at all times the existence of a lineal behavior in the computational cost with the product  $n \times m$  (coefficients of correlation of 0.99).

The use of a constraint satisfaction model reduces the number of operations carried out by the proposed algorithm,

which determine that correct results are always obtained without needing to include an additional computational cost (preprocessing) for the control of heuristics specific to the characteristics of the graphs considered.

Since the computational cost of the algorithm is affected by factors such as graphs size, presence of redundancies, number of *MCS*, presence of *MCS* of same size, etc., we are working on the design of new constraints in order to improve the performance of the algorithm.

The developed algorithm allows us to obtain, in a step-by-step process, each one of the maximum common substructures between two graphs, obtaining incremental similarity measures, from similarity measures based only on the maximum common substructure (*MCS*), until the based ones in the maximum overlapping (*AMCS*), which can be conveniently used in clustering and screening processes of chemical databases.

The model used by the algorithm is being updated in order to consider that the graph characteristics can be defined over nodes instead of one-depth subgraphs. This modification will

allow eliminating the inconvenience of the treatment of triodes that at the moment should be detected and solved, and we expect that these results may be the object of a forthcoming article.

## REFERENCES AND NOTES

- (1) Rouvray, D. H.; Balaban, A. T. *Chemical Applications of Graph Theory. Applications of Graph Theory*; Wilson, R. J., Beineke, L. W., Eds.; Academic Press: 1979; pp 177–221.
- (2) Ullmann, J. R. An Algorithm for Subgraph Isomorphism. *J. ACM.* **1976**, 23 (1), 31–42.
- (3) Messmer, B. T.; Bunke, H. Efficient Subgraph Isomorphism Detection: A Decomposition Approach. *IEEE Trans. Knowledge Data Eng.* **2000**, 12 (2), 307–323.
- (4) Epptein, D. Subgraph Isomorphism in Planar Graphs and Related Problems. *J. Graph Algorithms Applications* **1999**, 3(3), 1–27.
- (5) Pasari, R. *Visualization and Reduction Algorithms for Similarity Analysis of Molecules*; M.S. Thesis, Kent State University, Kent, Ohio, 1999.
- (6) Durand, P.; Pasari, R.; Baker, J.; Tsai, C. An Efficient Algorithm for Similarity Analysis of Molecules. *Internet J. Chem.* **1999**, 2, 1–12.
- (7) Chen, L.; Robien, W. MCSS: A New Algorithm for Perception of Maximal Common Substructures and Its Applications to NMR Spectral Studies. 2. The Applications. *J. Chem. Inf. Comput. Sci.* **1992**, 32, 507–510.
- (8) Parakulam, R. R.; Lesniewski, M. L.; Taylor-McCabe, K. J.; Tsai C. C. QSAR Studies of Antiviral Agents Using Structure–Activity Maps. *SAR QSAR Environ. Res.* **1999**, 10, 1–32.
- (9) Bunke, H.; Foggia, P.; Guidobaldi, C.; Sansone, C.; Vento, M. A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs. *Lecture Notes Comput. Sci.* **2002**, 2396, 123, 131.
- (10) McGregor, J. J. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. *Software Practice Experience* **1982**, 12, 23–34.
- (11) Garey, M. R.; Johnson, D. S. *Computers and Intractability: A Guide to NP–Completeness*; Freeman: 1979.
- (12) Haralick, R. M.; Elliot, G. L. Increasing Tree Search Efficiency for Constraint Satisfaction Problem. *Artificial Intelligence* **1980**, 14, 263–313.
- (13) De Jong, K. A.; Spears, W. M. *Using Genetic Algorithm to Solve NP–Complete Problems. Genetics Algorithms*; Schaffer, J. D., Ed.; Morgan Kaufmann: 1989; 124–132.
- (14) Herault, L.; Horaud, R.; Veillin, F.; Niez, J. J. Symbolic Image Matching by Simulated Annealing. *Proc. Br. Machine Vision* **1980**, 319–324.
- (15) Christmas, W. J.; Kittler, J.; Petrou, M. Structural Matching in Computer Vision Using Probabilistic Relaxation. *IEEE Trans. Pattern Analysis Machine Intelligence* **1995**, 17 (8), 749–764.
- (16) Ctdella, J.; Valiente, G. *A Relational View for Subgraph Isomorphism. Proceedings of Fifth Int. Seminar on Relational Methods in Computer Science*; Springer-Verlag: 2000; pp 72–78.
- (17) Simic, P. D. Constrained Nets for Graph Matching and Other Quadratic Assignment Problems. *Neural Computation* **1991**, 3, 268–281.
- (18) Raymond, J. W.; Gardiner, E. J.; Willet, P. Heuristic for Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm. *J. Chem. Inf. Comput. Sci.* **2002**, 42 (2), 305–316.
- (19) Larrosa, J.; Valiente, G. Graph Pattern Matching using Constraint Satisfaction. Proceedings of The European Joint Conferences on Theory and Practice of Software (ETAPS) 2000.
- (20) Cerruela García, G.; Luque Ruiz, I.; Gómez-Nieto, M. A. A New Algorithm to Obtain All Maximum Common Subgraphs in Molecular Graphs Using Binary Arithmetic and Constraints Satisfaction Model. Proceedings of International Conference of Computational Methods in Sciences and Engineering, Kastoria, Greece, September 2003, pp 156–159.
- (21) Willet, P.; Barnard, J. M.; Downs, G. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, 38 (6), 983–996.
- (22) Luque Ruiz, I.; Cerruela García, G.; Gómez-Nieto, M. A. Manuscript in preparation.

CI034167Y