match no. 27 pp. 3-30 1992

COMPUTER-ASSISTED STRUCTURE GENERATION FROM A GROSS FORMULA. 4. FIGHTING AGAINST GRAPH-ISOMORPHISM DISEASE.

Ivan P. Bangov

Institute of Organic Chemistry, Building 9

Bulgarian Academy of Sciences, Sofia 1113, Bulgaria

(received: February 1991)

A B S T R A C T

The isomorphism problem in the framework of the structure generation is discussed. A new approach aiming at avoiding most of the isomorphic structures is suggested.

The graph isomorphism problem following Read[1] is *"to devise a good algorithm for determining if two graphs are isomorphic"*. However this definition is impractical in the particular case of structure generation. We shall re-define it as: *"to devise a good algorithm for an a priori avoidance of generation of isomorphic graphs"*.

The difference between the two definitions is essential: whereas the former implies a comparison of each newly generated structure with the canonical representations of the structures previously generated, the latter leads to a straightforward construction of non-isomorphic structures.

In the present paper we report a new development of our structure generation approach[2-4] leading to a further substantial reduction of the number of generated isomorphic structures.

FORMAL THEORY

Chemical structures are mathematically treated with a special class of graphs: *irregular* (with vertices of different degrees, corresponding to atoms of different valences), *colored* (vertices corresponding to different types of atoms) *multi graphs* (having multiple edges but no loops).

A graph $\mathbf{g} = (\mathbb{V},\mathbb{E})$ is defined as a combination of two sets: the set $\mathbb{V}$ of the vertices (nodes) v_i and the set $\mathbb{E}$ of the edges e_{ij} where $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$. The Cartesian product $\mathbb{V} \times \mathbb{V}$ may be deemed as producing all the pairs of vertices (v_i, v_j). As only some of these pairs result in edges we can re-write the graph definition as follows:

$$\mathbf{g} = (\mathbb{V}, f\ \mathbb{V} \times \mathbb{V}) \qquad (1)$$

Here f is an incidence operator which selects only those pairs (v_i, v_j) forming edges e_{ij}.

There is one-to-one correspondence to chemical structure which can also be defined as $\mathbf{s} = (\mathbb{A},\mathbb{B})$ where $\mathbb{A}$ is the set of atoms a_i and $\mathbb{B}$ is the set of bonds b_i.

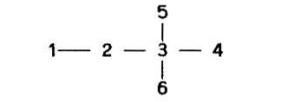
An adjacency matrix $|\mathbf{g}|$ can be juxtaposed to each graph. It is defined as a matrix whose entries obey the following relations:

$$g_{ii} = 0$$
$$g_{ij} = 1 \quad \text{if } (v_i, v_j) = e_{ij} \in \mathbb{E}$$
$$g_{ij} = 0 \quad \text{if } (v_i, v_j) \notin \mathbb{E}$$

As we shall discuss bellow other representations may also be employed.

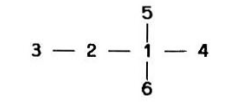
Let $\mathbb{S}_n$ is a symmetric group of operations and $U \in \mathbb{S}_n$ then the graph $\mathbf{g}^*$ is isomorphic to $\mathbf{g}$ if $|\mathbf{g}^*| = U.|\mathbf{g}|.U^{-1}$. The operation U leads to equivalent permutations between two rows and two columns of $|\mathbf{g}|$. For example the two graphs (**a**) and (**b**) with their adjacency matrices in **Figure 1** are obviously isomorphic. Note that the matrix (**b**) is obtained from matrix **a** by simultaneous permutations of

rows **1** and **3** and columns **1** and **3**. So long as there are **N!** permutations in total a combinatorial explosion of generated isomorphic graphs will result in the generation process.



	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	1	0	0	0
3	0	1	0	1	1	1
4	0	0	1	0	0	0
5	0	0	1	0	0	0
6	0	0	1	0	0	0

(a)



	1	2	3	4	5	6
1	0	1	0	1	1	1
2	1	0	1	0	0	0
3	0	1	0	0	0	0
4	1	0	0	0	0	0
5	1	0	0	0	0	0
6	1	0	0	0	0	0

(b)

Figure 1 Two isomorphic graphs and their adjacency matrices.

To avoid the perception of the isomorphic graphs in the course of the generation the following approach was suggested:[5,6,7]

Consider the set of isomorphic graphs $\mathfrak{G}_{is}$. One out of all graphs $g \in \mathfrak{G}_{is}$ is selected as canonical and the generation process is further directed to generation of canonical graphs only.

This rule can be expanded to incomplete structures. Hence the structure examines itself whether the continuation is canonical or not during the generation process. Thus, all branches of the generation tree leading to isomorphic structures are pruned and the generation problem alleviated.

To achieve this scheme a predicate for canonicity must be defined. It is usually related to the adjacency

matrix. Thus Faradjiev et al[5] suggested n^2 -dimensional vector $(g_{11}, g_{12}, g_{13}, \ldots, g_{nn})$, where n is the dimension of the matrix $|\mathbf{g}|$, and the entries g_{ij} are the matrix elements ranked column-wise. In this way all the matrices $\mathbf{g} \in \mathbf{G}_{is}$ may be compared lexicographically. The greatest matrix (having the greatest vector) is considered canonical.

For example the characteristic vectors of the two structures from **Figure 1** are:

| 010000 101000 010111 001000 001000 001000 |
(**a**)
| 010111 101000 010000 100000 100000 100000 |
(**b**)

Evidently, the structure (**b**) will be selected as canonical in this case. However, in the case of the complete set $\mathbf{G}_{is}$ other structure may appear canonical.

Kudo and Sasaki[6,7] have introduced connectivity stack as a canonicity predicate. Thus, the relation between the adjacency matrix elements (g_{ij}) and the position in the stack k is given by the following correspondence rule:

$$k = i + (j - 1)(j - 2)/2$$

Here again the greatest connectivity matrix is considered canonical.

This approach shows two substantial drawbacks:

First, all possible permutations of the segments corresponding to a given component must be carried out at this step. Only one of them produces the canonical form. (For definitions of the notions "*segment*" and "*component*" see ref.7)

Second, as written in ref. 7 "*the method for detection of isomorphism may accommodate only components which have bonding sites of single nature*". This makes impossible the treating of the partial structures (fragments) in the same manner as the single atoms. They have to be examined whether or not they are contained in the structure after

construction, i.e. first combinatorial operations are carried out for their construction from simple segments, then a time -demanding substructure search procedure is applied for their pruning. This makes the whole procedure too much computationally demanding.

Our efforts to the development of a new method for structure generation[2-4] were directed toward the solution of the above problems:

First, we do not favor the generation of the numerous isomorphism detection error-and-trial permutations;

Second, the treating of the fragments must be in the same way as the single atoms and groups, i.e. no combinatorial operations should be carried out for their construction and no substructure search procedures used afterwards.

FUNDAMENTALS OF THE METHOD

To implement these requirements into a practical algorithmic form a different view on the structure representation and generation and related to them isomorphism problem was necessary.

First, we consider directed graphs. A structure represented by a directed graph can be drawn as shown in **Figure 2.**

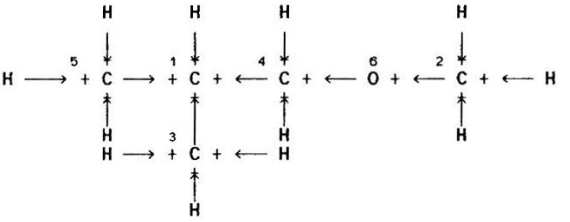


Figure 2. Drawing of directed graph.

One can see that there are two types of bonding sites (**bss**): ⟵ and +. In ref. 4 they were called Saturating

valences (**sv**s) and Saturation Sites (**ss**s). The following rules can be drawn from the representation given in **Figure 2**: (**i**) all the atoms but the first have n-1 **ss**s. and one **sv**; (**ii**) the first atom has n **ss**s and no **sv**. Here n is the number of **bs**s (free valences) of a given vertex (atom). One can see from **Figure 2** that a chemical bond is formed from the pair ($\mathbf{sv}_i$,$\mathbf{ss}_j$) i.e. we can define the set of bonds (edges) as a subset of the Cartesian product $\mathbb{E} \in \mathbb{SV} \times \mathbb{SS}$ where **ss** $\in \mathbb{SS}$ and **sv** $\in \mathbb{SV}$.

The atoms forming multiple bonds are considered varieties of the basic atoms. Thus, carbon, oxygen and nitrogen atoms forming double bonds are represented as newly defined **=C**, **=O** and **=N** atoms, having n = 3,1,2 respectively, and atoms forming triple bonds: **#C**, **#N**, with n = 2,1. The are subjected also to the rules **i** and **ii**.

The practical implementation of our method is illustrated in **Figure 3** with the generation of the structure from **Figure 2**. The forming of the **SUBS** and **GRAPH** arrays is presented in **Figure 3 (a)**. Here the first row elements of the **GRAPH** array are the **bs**s and the second row "+ " elements are the free unsaturated valences. The **sv**s form the vector array **SUBS**, and the **ss**s a two-row array **GRAPH**. Following rule **i** the first atom has four **ss**s and no **sv**, all other atoms providing one **sv** in the array **SUBS** and $n - 1$ **ss**s in the array **GRAPH** according to rule **ii** (n is the atom valence: n = 4 for carbon and n = 2 for oxygen atoms). A permutation of the m **ss**s selected from the L (m = 5, and L = 17 in our case) **ss**s without repetition and their substitution with the m **sv**s leads to a chemical structure as shown in **Figure 3 (b)**.

A juxtaposing of a first row with a second row element give the bonding between a **sv** and **ss** elements. Thus, all the ${}^{m}P_{L}$ permutations of the second row elements in **Figure 3 (b)** provide the full set of structures.

After the generation of the structure skeleton (**Figure 3 (b)**) the remaining "+" elements are saturated with **H** atoms (**Figure 3 (c)**).

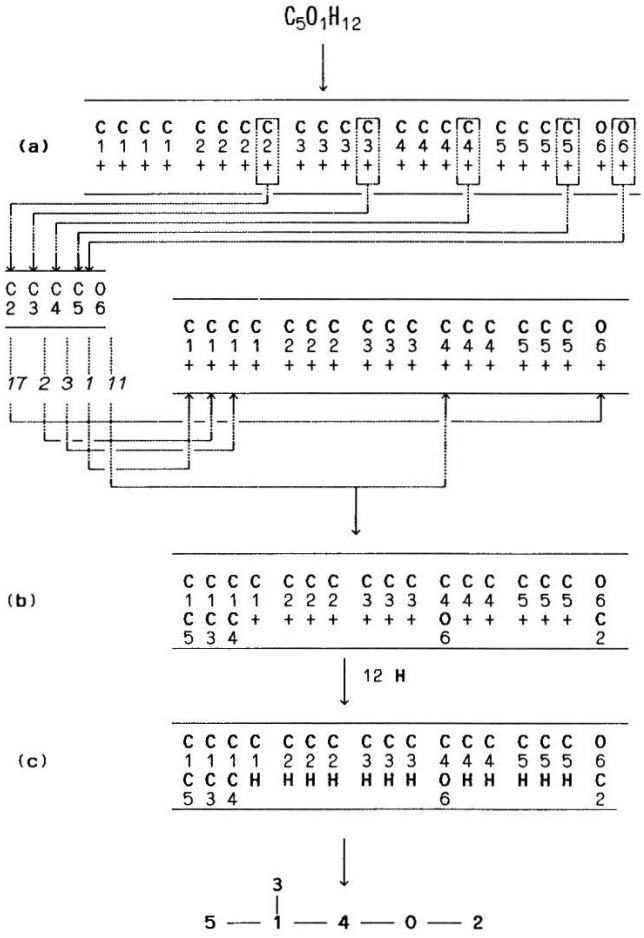


Figure 3. Two row-matrix representation in the process of structure generation *(permutation 17,1,2,3,11)*

If we have spectral information about the **C–H** adjacency, e.g. from off-resonance ^{13}C NMR or DEPT spectra, the second row **GRAPH ss**$_s$ may be filled with **H** atoms. Hence the number of **ss**$_s$ will be reduced and the combinatorial problem alleviated.

The reason that we use **bs**$_s$ (**sv**$_s$ and **ss**$_s$) instead of single atoms is the need of treating fragments (partial structures) as *superatoms*. Following ref. 8 the fragments are considered *superatoms* when they possess any number of free valences (**bs**$_s$). The inner structure of the fragment (atoms and their saturation valences) is "invisible" for the combinatorial process, only the **bs**$_s$ participate in it. Here the valence of a fragment *n* is equal to the number of its **bs**$_s$. The same rules as for single atoms are applied to them:

iii *n–1* **bs**$_s$ of each fragment but the first are of the **ss** type and one **bs** of **sv** type;

iv the first fragment has all its *n* **bs**$_s$ **of ss** type.

However, a peculiarity in the treatment of fragments, is that their **bs**$_s$ are no more equivalent as it is with the single atoms and chemical groups. Hence each nonequivalent **bs** of any but the first fragments is selected once as **sv** while the other **bs**$_s$ are **ss**$_s$. Only one of the equivalent **bs**$_s$ must be selected as **sv**.

Further, in order to reduce the combinatorial operations we treat the multiple bonds as multiple bond fragments such as **C=C**, **C=N**, **C=O**, etc., rather than single atoms **=C**, **=N**,**=O**. They obey also the rules **iii** and **iv**.

The treatment of fragments is illustrated with the generation of two isomers of the gross formula $C_5O_1H_{12}$ in **Figure 4(a** and **b)**. Two fragments: $CH(CH_3)_2$, CH_2O and one chemical group CH_3 participate in the generation. The first isomer **(a)** is constructed by taking the **sv** of the second fragment from atom **C** and the second isomer, by taking the **sv** from atom **O (b)**. Note that the second fragment has changed its direction from ← **CH** + ← **O** to ⟵ **O** + ⟵**CH**.

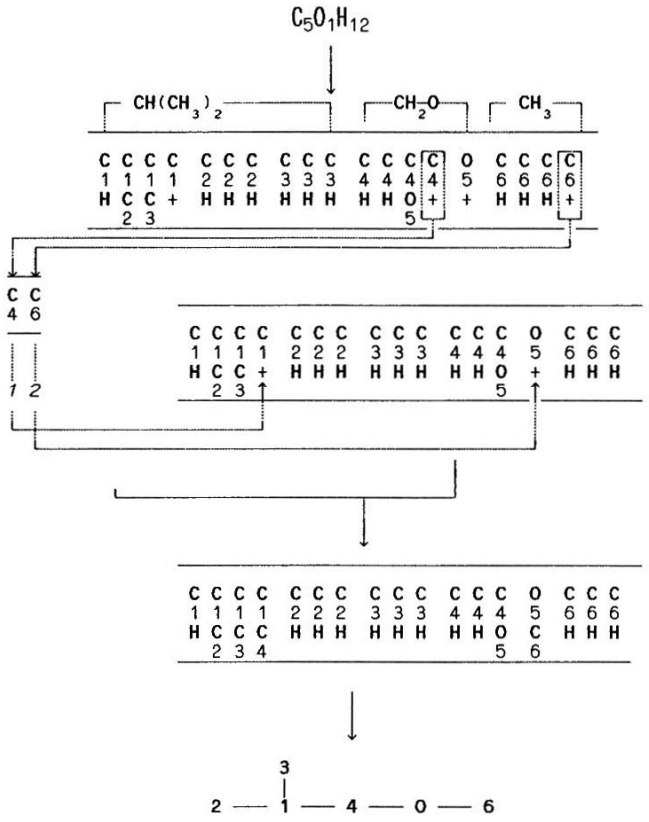


Figure 4 (a) Generation of an isomer with the second fragment **sv** from C_4 atom taken (permutation *1,2*).

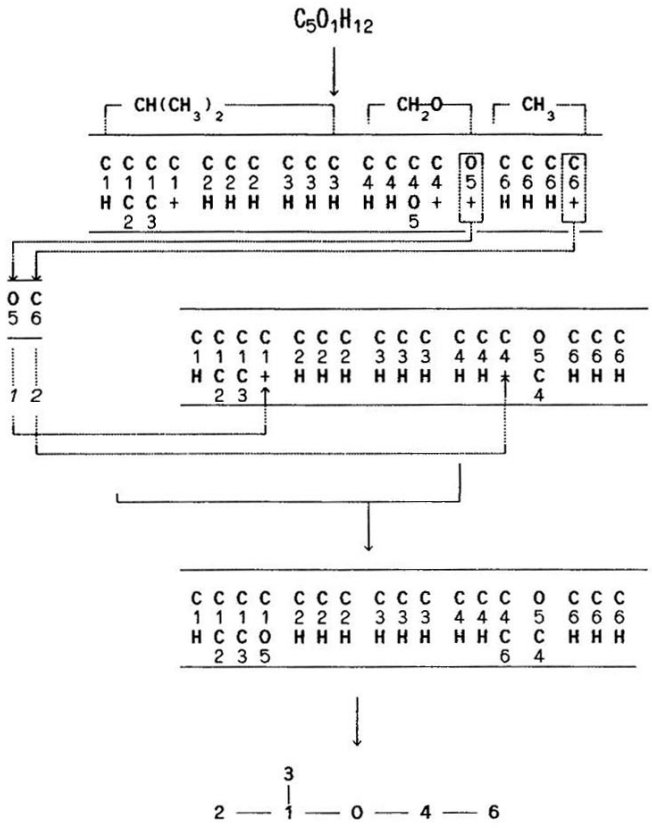


Figure 4 (b) Generation of an isomer with the second fragment **sv** from O_5 atom taken.

ISOMORPHISM TREATMENT

The use of directed graphs reduces the number of permutations between the **bs**s from $N!$ to ${}^{m}P_{L} = L!/(L - m)!$, where N is the total number of **bs**s, m is the number of **sv**s and $L = (N - m)$ is the number of **ss**s.

However, a great part of the structures generated in this manner are either disjointed or isomorphic. The sources of isomorphism were studied in ref. 4. It was shown that isomorphism is a function of the automorphism and the following rule was devised:

To avoid the generation of isomorphic structures any permutation between **bs**s *from equivalent (automorphic) atoms must be avoided.*

Following this rule two interrelated schemes: **H**ierarchical **S**aturation with **E**quivalent **Sv**s (**HSESV**) and **H**iearchical **S**election of the **Ss** (**HSSS**s) were developed.[4]

The former consists of a partitioning of the **sv**s according to the topological equivalence of their atoms: $*_1 \in \mathbb{V}_1$, $*_2 \in \mathbb{V}_2$, $\ldots . *_{n-1} \in \mathbb{V}_{n-1}$, $*_n \in \mathbb{V}_n$. It was shown that to avoid permutations between equivalent **sv**s ${}^{mi}C_{Li}$ combinations are generated hierarchically between elements of each equivalence class. Here mi is the number of equivalent **sv**s in the i-th class $\mathbb{V}_i$ and Li are the **ss**s selected through the **HSSS** approach from the $L - \sum m_j$ **ss**s left from the previous saturations. Thus

$$ {}^{m1}C_{L} \cdot {}^{m2}C_{L - m1} \cdot {}^{m3}C_{L-m1-m2} \cdot \ldots \cdot {}^{mn}C_{L-\sum mn-1} \qquad (2) $$

combinations are generated in total. This approach is illustrated in **Figure 5** with the generation of the structure from **Figures 2** and **3**.

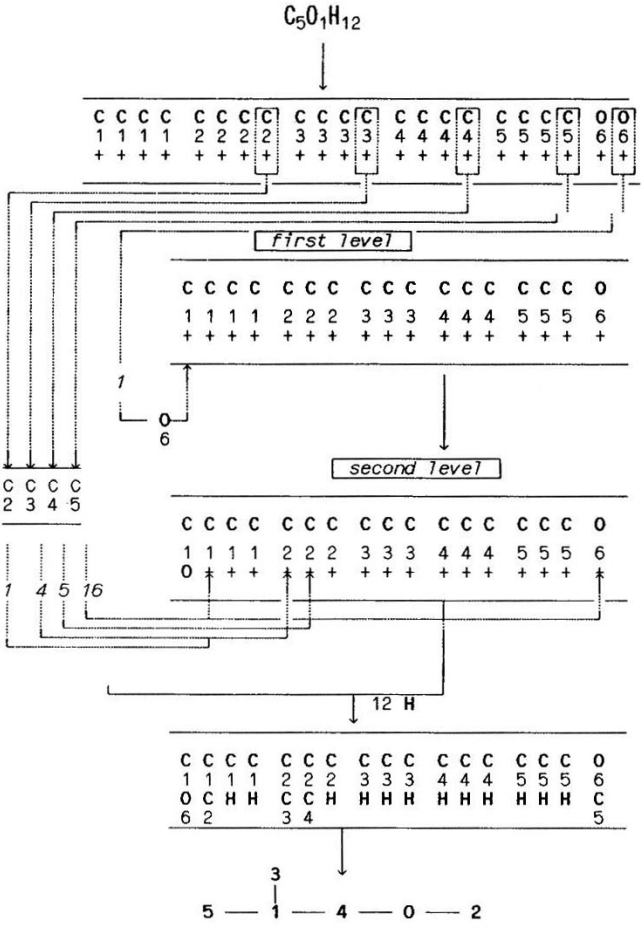


Figure 5. Hierarchical structure generation (first level combination *1*, and second level *(1,4,5,16)*

Here the **sv**s are partitioned into 2 equivalence (automorphism) classes (**HSESV** approach): the former containing the sole **O** atom **sv**, and the latter 5 **C** atom **sv**s.

17 **ss**s can be selected (**HSSS** approach) for the first level in **Figure 5**. However one **ss** of them is at the **O** atom and it must be avoided because it leads to bonding of the atom to itself. All other 16 **ss**s emanate from equivalent **C** atoms. Since we have one **O** at this level only one **ss** (according to the **HSSS** aproach) must be selected and saturated. Thus, we have only one combination for this level.

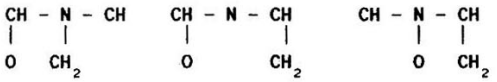
$^{5}C_{15}$ combinations will be generated at the second level of **Figure 5**, each time saturating 5 **ss**s selected from the 15 unsaturated **ss**s with the 5 **sv**s. Thus all the structures are generated.

Note that the structure generated in **Figure 5** is isomorphic to the structures from **Figures 2** and **3**, but it has different numbering. In contrast to the latter the numbering of the equivalent **C** atoms in the former is in alphanumerical order (2,3,4,5), and no combination violates this order. The isomorphic structure from **Figure 3** is due to inverse permutations. Thus the use of combinations within this hierarchical scheme ensures an automatic obtaining of the canonical form of the structures, the non canonical forms being simply skipped. The advantage of this method is that it avoids most of the trial-and-error permutations generated to determine the canonical form.

However the number of generated combinations $^{5}C_{15}$ = 3003 is still rather large, for such a simple case and there still remain a number of isomorphic structures. In ref.4 we solved this problem by employing spectral information. The ^{13}C - ^{1}H direct multiplicity from ^{13}C NMR spectra provides the adjacent **C** -**H** connectivity which leads to a further partitioning of the **C** atoms into **CH**,

CH_2, CH_3 groups. This partitioning reduces our problem to the real case, hence the combinatorial problem is alleviated.

It is obvious that a number of combinations from the product (2) leads also to isomorphic structures which are due to combinations of two or more **sv**s to two or more equivalent **ss**s e.g. from the three isomeric substructures:



the first and the third are isomorphic.

Further we report our efforts to develop a graph-topological approach to the reduction of the product (2) hence to a further reduction of the isomorphism problem.

To achieve this the following generation scheme was developed:

(1) All the atoms are partitioned into automorphism classes: $*_1 \in \mathbb{V}_1$, $*_2 \in \mathbb{V}_2$, ... $*_n \in \mathbb{V}_n$. This is carried out by assigning each atom with a **LO**ocal **T**opological **I**ndex (**LOTI**). The nature of this index will be discussed bellow. The **sv**s and **ss**s follows the partitioning of the corresponding atoms.

(2) The **sv**s are ranked according to their **LOTI**s forming a hierarchical system of levels, the greater is the **LOTI** of a given **sv** the lower level it occupies. Only one **sv** is placed on a separate level. The **sv**s of the same **LOTI**s form a set of classes (cells) of equivalent levels.

(3) The structure generation is a hierarchical depth -first procedure of dynamical execution of the steps: **LOTI** -determination, **sv** -partitioning into different levels, **ss**- selection through a modification of the **HSSS** approach, and saturation of the **ss**s with **sv**s.(**ss**- saturation). The first two steps are carried out only when the program execution passes from one cell of equivalent levels to

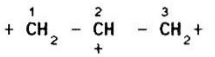
another. The modification of the **HSSS** consists of a selection at each current level of **ss**s for the higher level which obeys the following rules:

(i) if the higher and current level are of the same equivalence cell then the selection of the higher level **ss**s starts from **ss** saturated at the current level. Thus, the equivalent **sv**s follow the arrangement that the lower level **sv** number is always preceding the higher level **sv** number. Consequently, the generated structures have the greatest adjacency matrix i.e. they are canonical.

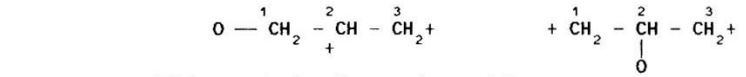
(ii) In cases that the current level and the higher levels are not equivalent then the **ss** selection starts from the first atom, hence inverse permutations between non- equivalent **sv**s are also generated.

(iii) Only one (from the atom bearing the lowest number) out of all **ss**s belonging to a given automorphism class is selected for the next higher level **ss** -saturation step i.e. the selected **ss**s are of atoms belonging to different automorphism classes.

For example, 3 **ss**s are to be selected for the higher level **O** atom **sv** of the substructure given below:



As the **ss**s of atoms 1 and 3 belong to the same automorphism class only **ss**s from atoms 1 and 2 will be selected at this level which will result in the following substructures:



which are obviously non-isomorphic.

The work of our algorithm is depicted with the flow-chart in **FIGURE 6.** and exemplified in **FIGURE 7** with the isomer generation from the gross formula C_6OH_{14} . The corresponding matrix representation is provided in **Figure 8.**.

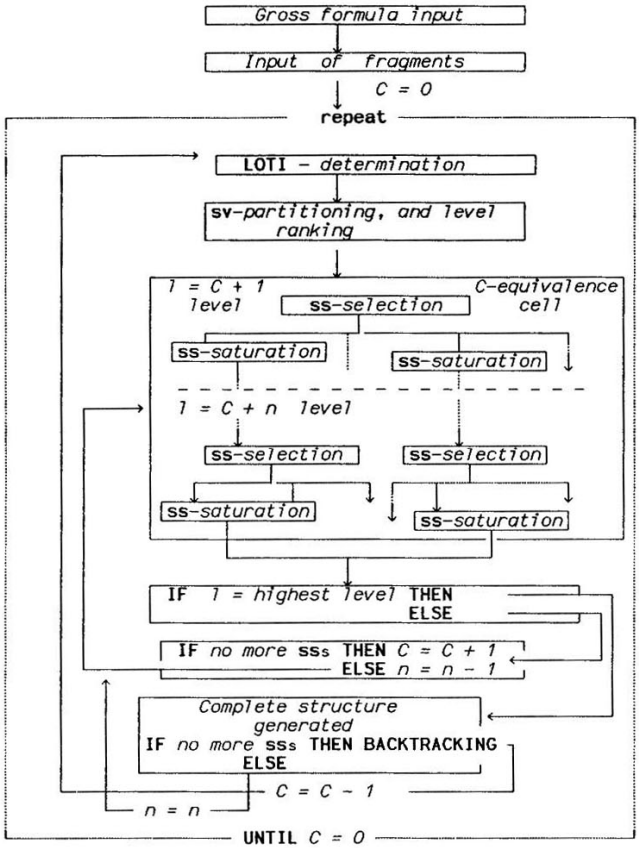


FIGURE 6. Flow-chart of the generation process. *l* is the levels rank, *C* - equivalence cell number and *n* - number of levels in a given cell.

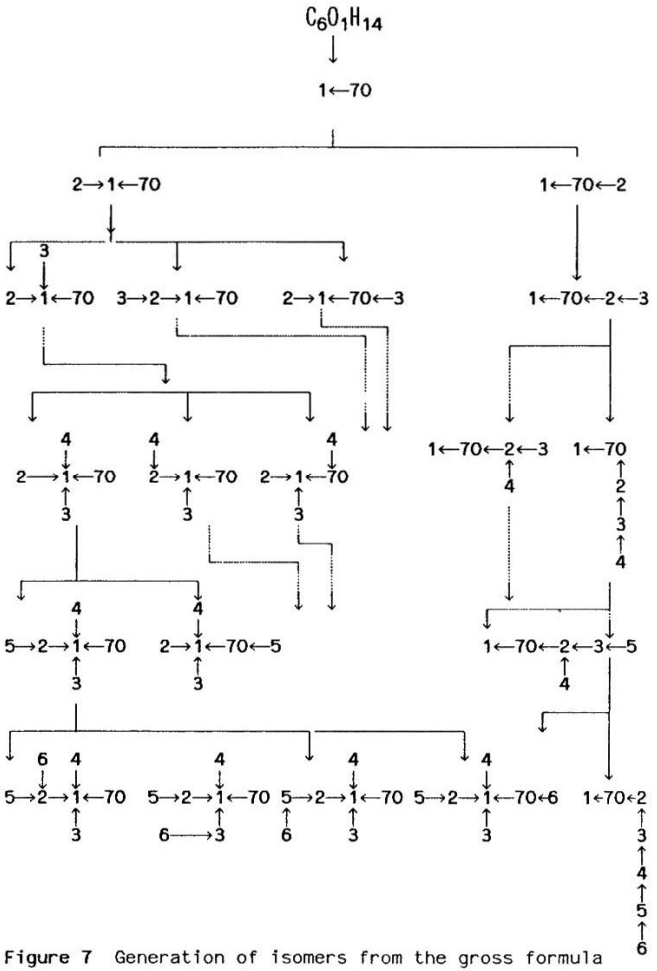


Figure 7 Generation of isomers from the gross formula $C_6O_1H_{14}$. A number indicates a carbon atom. Number and atom identifier a heteroatom.

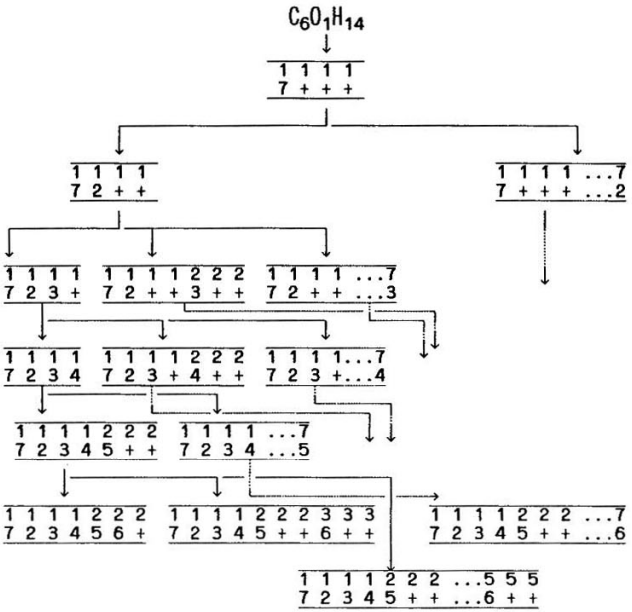


Figure 8. Matrix representation of the generation process depicted in **Figure 7.**

The only **O sv** at the first level is bonded to the 1st atom **ss**. Only one **ss** is selected for the sole **O** atom **sv**, because all the **C** atoms are equivalent. Next are the five equivalent **C** atom **sv** levels.

Two **ss**s from the two non-equivalent atoms of the substructure are selected for the second level saturation. The latter produces two substructures. The former **2→1←70** provides 3 **ss**s from the two non-equivalent **C** atoms, and

from the only **O** atom, and the latter **1**⟵ **70**⟵ **2**, one **ss** from atoms **1** (atoms **2** is equivalent to atom **1** therefore the program selects **ss** from the atom with the lowest number). In the same way only one **ss is** selected from the three equivalent atoms **2,3,4** of the first substructure at the third level.

Thus, following rules **i..iii** the program constructs the full structure in such way that the numbering of the equivalent atoms follows an arrangement: the **sv** of the higher numbered atom saturates the **ss** of lower numbered atom. In case of non-equivalent atoms e.g. the **O** atom in our case, this arrangement is violated. Thus the generation process is directed toward the building of canonical (having the maximal adjacency matrix) structures. It should be pointed out though that even in this case some duplications still appear, e.g. the two substructures **2**⟶**1**⟵**70**⟵**3** and **1**⟵**70**⟵**2**⟵**3** at the second level. These duplications are recognized as early as possible and pruned. Their recognition is discussed bellow.

CYCLIC STRUCTURE GENERATION

While the construction of acyclic structures might be independent from the current numbering, the generation of closure bonds forming the rings depends both on the equivalence of the atoms and on their numbering. For example if we take a given spanning tree the rings formed by two closure bonds incident to two equivalent pairs of atoms, might have different sizes. Therefore the following hierarchical approach for the cyclic structure generation was adopted:

First, the spanning tree (acyclic skeleton) of the given cyclic structure is generated following the method described above, e.g. the acyclic structure generated at level 5 in **Figure 7** may be used in **Figure 9.**

Second, a hierarchical scheme of generation of

closure bonds incident to the free valences is developed. The number of levels (hereafter they will be called cyclic levels) is determined by the following relation :

$$Dc = Du - N_{db} - 2 .N_{tb} \qquad (3)$$

Here Du is the degree of the unsaturation, N_{db} and N_{tb} are the numbers of double and triple bonds respectively.

A closure bond (**cb**) is a directed bond which is formed of a **sv** (we call it closure bond **sv** (**cbsv**)) and a closure bond **ss** (**cbss**). The generation of closure bonds starts from the highest level of generation of the spanning tree with the selection of the **cbsv**s for the first cyclic level. This procedure obeys the following rules:

(i) All the bonding sites (**bs**s) are partitioned into different equivalence classes according to the partitioning of the corresponding atoms.

(ii) All but the last **bs**s are traced and a set of **cbsv** levels is formed: the first **cbsv** occupying the first level and the higher levels are formed of **bs**s from atoms having higher numbers than the previous level selected **cbsv**. For example, if the 1st level **cbsv** in **Figure 9** is from atom **3** then next level **cbsv** may be selected only from atoms bearing higher numbers such as **5,7**. However the last **bs** (of atom 7) is not taken because no partner **cbss** can be selected following the rules for the **cbss** selection given bellow;

(iii) Only one (having the lowest number) out of all the **cbsv**s belonging to a given equivalence class is selected, i.e. only **bs**s non-equivalent to any of the previously selected can be transformed into **cbsv**s;

Thus, following the rules (i)..(iii) only **bs**s from atoms **2,3,5** may be selected as **sv**s at the first level in **FIGURE 9**. The **bs**s from atoms **4,6,7** are excluded: **bs**s from atoms **4,6** because they are equivalent to atoms **3,5** respectively (violating rule iii, and the **bs** from atom 7

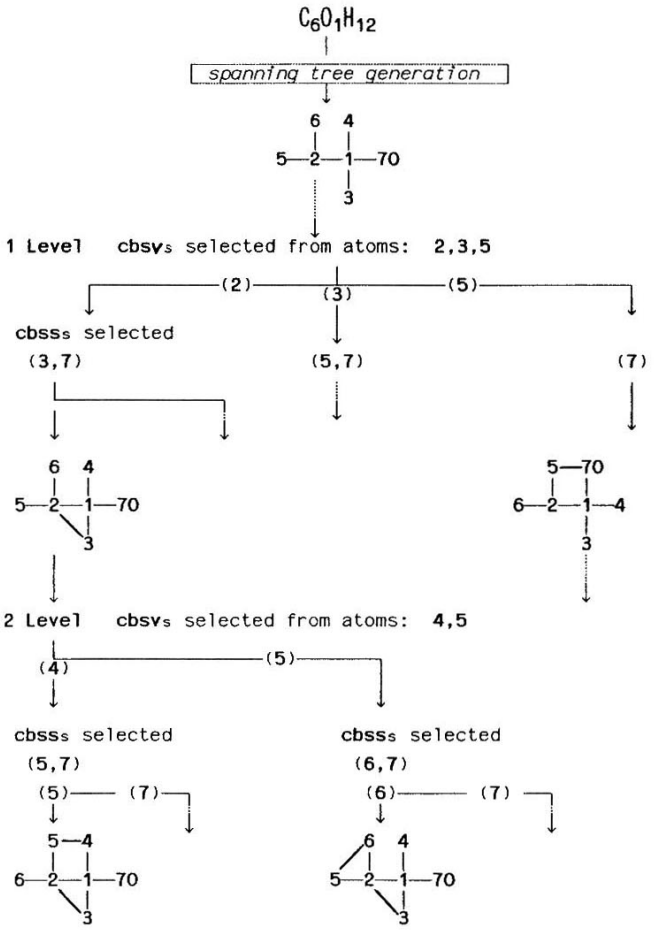


Figure 9. Generation of cyclic structures from a spanning tree.

because it is the last atom (violating rule **ii**).

After a selection of a given **cbsv** the higher level **ss**s are subsequently selected according the following rule:

All the **bs**s starting from the n +1 **bs** (n is the current level **cbsv**) are traced and a **bs** is selected as closure bond **ss** (**cbss**) if the following requirements are met:

(**j**) The **ss** is either not equivalent to any **cbss** previously selected (**cbss**$_{pr}$) or the topologigal distance (D) to the partner **cbsv** (**ss-cbsv** distance) is not equal to the **cbss**$_{pr}$-**cbsv** distance (D_{pr}). i.e. if the **ss** under selection is equivalent to any **cbss**$_{pr}$ and the D is the same as the D_{pr} then **ss** is discarded.

(**jj**) Only **bs**s forming preset sizes of rings with the partner **cbsv** are selected for **cbss**s. Note that the topological distance between the **cbsv** and the **cbss** atoms gives the size of the generated ring -1 one. On the one hand, **bs**s which are of distance D = 1 from the partner **cbsv** are automatically discarded, since they form double and triple bonds which are treated in different way in our case. On the other hand, this rule allows us to avoid the generation of structures with forbidden ring sizes;

Thus if the **cbsv** in **Figure 9** is selected from atom **2** then **cbss**s from atoms **3** and **7** only will be selected since the distances $D_{2,5}$ and $D_{2,6}$ are equal to one (no ring can be formed between atoms **2** and **5** and **2** and **6** according to rule **jj**) On the other hand atom **4** is equivalent to atom **3** and $D_{2,3} = D_{2,4}$ (rule **j**), hence the program chooses the lower number atom **3**. The same procedure is repeated at the higher cyclic level.

The matrix representation of a branch of the cyclic structure generation from **Figure 9** is provided in **Figure 10.** Here the **ss** selected as **cbsv** is coded as **[** (it is considered dummy **ss**) and the real **cbsv** as **]**.

As discussed in ref 4 the **HSSS** approach allows imposing different constraints having either chemical nature or induced from the structural information

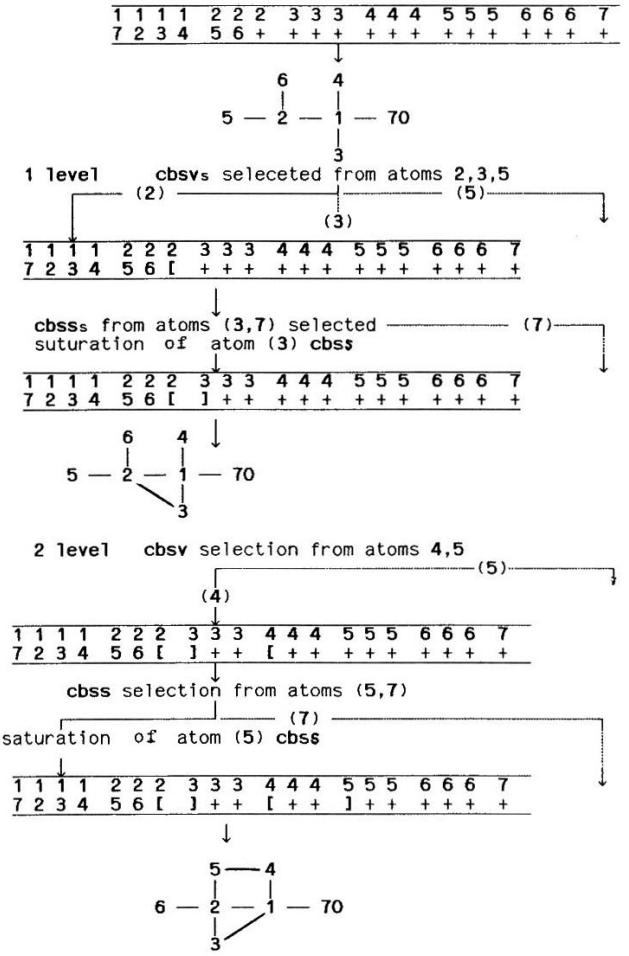


Figure 10. Matrix representation of a cyclic structure generation provided in Figure 9.

available. Additionally, it is worth mentioning the restriction that no closure bond can be formed between **Ph** ring atoms, i.e., the selection of **Ph** ring **ss**$_s$ for any **Ph** ring **sv** is forbidden.

AUTOMORPHISM PARTITIONING AND ISOMORPHISM PERCEPTION

It is clear from the above presentation that the automorphism partitioning carried out after each level **ss**-saturation is of underlying importance for our method.

Generally speaking the automorphism partitioning is associated with the vertex classification within a given structure usually based on permutational procedure. It was pointed out[1] that the number of permutations can not be less than $\prod (\omega_i!)$, where ω_i are the vertices belonging to a given automorphism class.

Here we use the notion automorphism partitioning for a vertex classification of the separate units (single atoms, chemical groups and atoms in fragments) which are to form the structure.

It should be emphasized that we can not use any permutational scheme because it is too much time demanding in the case of structure generation. Therefore we have been forced to employ some local invariants (indices) to this end. In recent papers[4,9] we devised the **A**tom in Structure Invariant Index (**ASII**):

$$\mathbf{ASII} = \mathbf{ASII}_o + \mathbf{N}_H - \mathbf{Q}_{at} \qquad (4)$$

where $\mathbf{ASII}_o$ is a constant associated with the type and hybridization state of the given atom, $\mathbf{N}_H$ is the number of hydrogen atoms attached to a given vertex (non-hydrogen atom), and $\mathbf{Q}_{at}$ is the charge density of the atom calculated through a fast and efficient method (we use the **IPEOE** method of Gasteiger et al.[12]). So long as this method is applied not only to complete structures but also to fragments, the free valences (**bs**$_s$) are considered new

univalent dummy atoms '+' with arbitrary chosen values for the **IPEOE** parameters. The $ASII_o$ values are given in **Table 1.**

The partitioning procedure is the following:
First, the vertices are partitioned into the following groups:

h- group, of the heteroatoms;
m- group of atoms having π- electrons;
c- group of sp^3 carbon atoms;

Further within each group the vertices are partitioned according to Equation 4. This partitioning is dynamic and according to **Figure** 6 it is carried out for each higher level not equivalent to the current one and vice versa not carried out between equivalent levels.

Table 1. Initial values $ASII_o$ for different types and hybridization states.

Type and hybridization state	$ASII_o$
C	
sp^3	4
sp	7
sp^2 (olefinic)	11
sp^2 (aromatic)	13
N	
sp^3	15
sp^2	18
sp	20
O	
sp^3	23
sp^2	25
F	32
S	28
Cl	33
Br	34
I	35

As mentioned above, despite the efforts to avoid the generation of isomorphic structures, a small number still appear. They must be perceived at an as low as possible level and pruned. We use our Electro-Topological Index (**ETI**).[4,11] to this end. It is formed from the **ASIIs** of the separate atoms and has the following form:

$$\mathbf{ETI} = \left(\sum_{i} \sum_{j} (\mathbf{ASII}_i . \mathbf{ASII}_j) / D_{ij} \right)^{1/2} \qquad (5)$$

Here the summation over i. and j is over all the atoms (bonded and non-bonded) and D_{ij} are the corresponding topological distances. Thus the **ETI** of each substructure or complete structure generated at this level is compared with the **ETIs** of substructures or structures previously generated and the process of generation goes to the higher level if no **ETI** of the latter is equal to that of the former.

RESULTS

The work of our generator is exemplified with some results presented in **Table 2**. They are compared with data from the compilation of numbers of generated structures provided in the paper of Kerber et al.[13] However, some discrepancies are due to the different manner of treating the unsaturation by the two methods, on the one hand, and that some groups having atoms =C=, have not been still included in our program, on the other.

Thus, indicative is the case **32** from **Table 2**. Here the difference is due to the fact that our program considers the first carbon atom of the constraint $C\text{-}C(OH)_2$ as sp^3 carbon atom hence no structures having the fragment $C{=}C\text{-}C(OH)_2$ can emerge from it. On the other hand such structures are generated by adding a vinyl group to the second carbon atom. By examining the the 222 structures in the case **30** of the **Table** we found their number to be 50.

As the whole number of structures containing the constraint $C{=}C{-}C(OH)_2$ emerging from both carbon atoms is 90 (case **29** in the **Table**) then the number of structures which are not generated by our approach is 40 and the sum 524 + 40 = 564 gives just the number of the structures generated by the program of Kerber et al.

Table 2. Number (No) of structure generated from a given gross formula under given constraints, compared with the corresponding number from ref. 13.

	Gross formula	constraints	No	No(ref.13)
1.	C_6H_{10}	1 vinyl group and 1 ring	37	
2.	C_6H_{10}	2 rings	17	
3.	C_6H_{10}	2 vinyl groups	11	
4.	C_6H_{10}	1 C≡C group	7	
5.	C_6H_{10}	no[a]	72	77[b]
6.	C_6H_{12}	no	25	25
7.	C_6H_{14}	no	5	5
8.	C_7H_{14}	1 vinyl group	27	
9.	C_7H_{14}	1 ring	29	
10.	C_7H_{14}	no	56	56
11.	$C_6O_1H_{10}$	2 vinyl groups	76	
12.	$C_6O_1H_{10}$	1 vinyl and 1 C=O groups	34	
13.	$C_6O_1H_{10}$	1 acetylene group	47	
14.	$C_6O_1H_{10}$	1 vinyl group, 1 ring	335	
15.	$C_6O_1H_{10}$	1 C=O, 1 ring	33	
16.	$C_6O_1H_{10}$	2 rings	170	
17.	$C_6O_1H_{10}$	no	695	747[b]
18.	$C_6O_1H_{12}$	1 vinyl group	95	
19.	$C_6O_1H_{12}$	1 C=O	14	
20.	$C_6O_1H_{12}$	1 ring	102	
21.	$C_6O_1H_{12}$	no	211	211
22.	$C_6O_1H_{14}$	1 OH	17	
23.	$C_6O_1H_{14}$	only ether structures	15	

Table 2 *continued*

24.	$C_6O_1H_{14}$	no	32	32
25.	$C_6N_1H_{15}$	no	39	39
26.	$C_6N_1O_1H_{11}$	1 C≡N group	64	64
27.	$C_6N_1O_1H_{11}$	1 C≡N, 1 OH groups	31	31
28.	$C_8O_2H_{16}$	1 COC, 1 C=O groups	320	320
29.	$C_8O_2H_{16}$	$C=C-C(OH)_2$	90	
30.	$C_8O_2H_{16}$	$C-C(OH)_2$, 1 vinyl group	222	
31.	$C_8O_2H_{16}$	$C-C(OH)_2$, 1 ring	302	
32.	$C_8O_2H_{16}$	$C-C(OH)_2$	524	564[c]
33.	$C_{12}H_{16}$	$CH_2-\underset{\displaystyle Ph}{\overset{\displaystyle |}{C}}-CH_3$, 1 ring	528[d]	
34.	$C_{12}H_{16}$	$CH_2-\underset{\displaystyle Ph}{\overset{\displaystyle |}{C}}-CH_3$, 1 ring	59[e]	
35.	$C_{12}H_{16}$	$CH_2-\underset{\displaystyle Ph}{\overset{\displaystyle |}{C}}-CH_3$, 1 ring	29[f]	

---

[a] Hereafter no constarint is formed as a sum of the number of generated cyclic structures plus the number of generated structure having unsaturated fragments.
[b] The difference is due that structures with =C=, and =C=O groups are not generated by our method.
[c] This difference is explained in the text.
[d] All the positions in the Ph ring are free valences (unsaturated with H atoms).
[e] The two meta- positions are free valences.
[f] The para- position is a free valence.