

## GENERATION OF MOLECULAR GRAPHS WITH A GIVEN SET OF NONOVERLAPPING FRAGMENTS

Sergey G. Molodtsov

Institute of Organic Chemistry

Siberian Division of the Russian Academy of Sciences

Novosibirsk 630090, Russia

(received: April 1994)

### Abstract

An extension of algorithm and program GENM for molecular graph generation is described. It provides effective construction of graphs containing the set of nonoverlapping fragments. Examples of isomer generation based on various sets of fragments and corresponding computer time are given.

### 1. Introduction.

The previous article [1] describes an algorithm and program GENM for generation of all connected molecular graphs (isomers of chemical compounds with given molecular formula). This algorithm takes into account some structural constraints represented by maximal possible and precisely formulated multiplicity of edges for vertices with given labels. Even the simple molecular formula may lead to construction of hundreds of thousands various isomers. For example there are 184131 isomers with molecular formula  $C_5H_4FNO_2$ . Generation of great number of graphs takes a lot of computer time and leads to not easy problem of analysis of all constructed graphs. Meanwhile in many cases some additional information about possible structure constraints may essentially simplify the problem. For example in solution of many problems of structure elucidation using molecular spectra [2,3], prediction structures with desirable biological activity [4] we need to construct all compounds containing given set of fragments. If we shall make all checks on the presence of necessary fragments after complete construction of each molecular graph, the overall time of isomer generation increases drastically. Therefore we need to take into account these structural constraints in the process of graph generation.

In this article we describe the procedure of setting the information on fragments for molecular graph generation program GENM. It provides the effective construction of all graphs (described by adjacency matrices), containing given set of nonoverlapping fragments.

## 2. Definitions.

Let  $G = (V, E)$  be an undirected graph without loops and with multiple edges, where  $V$  is the set of labelled vertices and  $E$  the set of edges. Denote  $val_i$  the valence of vertex  $v_i$  and  $A = (a_{ij})$  the adjacency matrix of a graph  $G$ . Define a fragment of a graph  $G$  as an induced subgraph of  $G$  for which any two vertices are adjacent if and only if they are adjacent in the graph  $G$ . Note that we do not impose any constraint on the connectivity of an fragment. We can use nonconnected fragments for representation of subgraphs that are explicitly known as nonconnected.

Let  $F^k = (U^k, X^k)$  is a set of nonoverlapping fragments of a graph  $G$ , i.e.  $U^k \cap U^l = \emptyset$  for any  $k \neq l$ . Let us regard a partially filled matrix  $B = (b_{ij})$  with elements

$$b_{ij} = \begin{cases} a_{ij}, & i, j \in U^k \text{ for some } k \\ 0, & i = j \\ \delta, & \text{in all rest cases} \end{cases}$$

where  $\delta$  means still not filled elements.

Let us take the matrix  $B$  as initial matrix for generation of graphs. Graph generation algorithm [1] is a stepwise procedure. On each step it constructs all weakly canonical complements of strongly canonical matrix taken from the previous step and selects all strongly canonical ones among them. It is clear that both matrix  $B$  and all its complements contain adjacency submatrices corresponding to given set of fragments  $F^k$ .

Let  $fv_i = val_i - \sum_{b_{ij} \neq \delta} b_{ij}$  is remaining free valency of the vertex  $v_i$ . Here and further if not stated otherwise we shall use the term valency in the sense of remaining free valency of a vertex.

DEFINITION 1. We call vertices of fragments with free valencies as *external* and vertices without free valencies as *internal*. All vertices of a graph that do not belong to any initial fragments we call *free* vertices.

Evidently each fragment has external vertices that are connected with other vertices of a graph.

DEFINITION 2. We call fragments with only external vertex as *simple* fragments.

Here are some examples of simple fragments:



Free valencies are shown as dashes going out of vertices. We will assume that unlabelled vertices in a cycle have the label **C** with number of adjacent hydrogen vertices (label **H**)

calculated from the valency of C-vertex. Chemical nature of atoms implies the following valencies of vertices: C - 4, H - 1, N - 3, O - 2, F - 1.

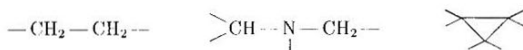
Define an equivalence relation on the set of vertices. We say vertices  $v_i$  and  $v_j$  of a fragment are *equivalent* if there exists an one-to-one correspondence of vertices preserving connectivity and labels that maps the vertex  $v_i$  to  $v_j$ . Then the set of fragment's vertices is divided into equivalence classes. These classes are known as *orbits*.

Let us take all fragments that have more than one external vertex.

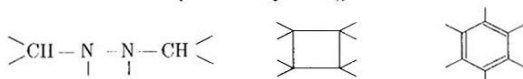
DEFINITION 3. We call the fragments preserving the equivalence relation after connection of any external vertex with other vertices of a graph as *intermediate*.

DEFINITION 4. We call the fragments changing the equivalence relation after connection of any external vertex with other vertices of a graph as *complex*.

Examples of intermediate fragments are the fragments with only two external vertices, fragments with all unique external vertices (belonging to different one-element orbits):



Here are some examples of complex fragments:



Remind that the initial data for the graph generation algorithm [1] are the set of labelled vertices with defined valencies, divided by initial classes  $V_1, V_2, \dots, V_p$  and the matrix of the maximal possible and necessary multiplicity of edges between vertices  $R = (r_{ij})$ .

Further we shall analyze examples of handling simple, intermediate and complex fragments.

### 3. Simple fragments.

Let  $F^k = (U^k, X^k)$  be the set of nonoverlapping isomorphic to each other simple fragments of a graph  $G$ .

Let us divide all vertices of the fragments  $F^k$  by two nonoverlapping classes. The first class will contain all internal vertices, the second class - all external ones. Free vertices will be divided into classes according to their labels and valencies.

Construct the partially filled matrix  $B = (b_{ij})$ . Define  $R = (r_{ij})$  to be a matrix of the

maximal and necessary multiplicity of edges between vertices as:

$$r_{ij} = \begin{cases} -b_{ij}, & b_{ij} \neq \delta \\ c_{kl}, & \text{for any } i \in V_k, j \in V_l \text{ such that } b_{ij} = \delta \end{cases} \quad (1)$$

where  $c_{kl}$  is the maximal possible multiplicity of edges between vertices from classes  $V_k$  and  $V_l$ . It is evident that  $r_{ij} \leq \min(fv_i, fv_j)$  for any  $i, j$  such that  $b_{ij} = \delta$ .

Then due to taken partition of vertices to classes and due to the matrix  $R$ , algorithm [1] will generate only graphs containing the set of fragments  $P^k$ . Adjacency matrices of fragments are included in initial partially filled matrix  $B$  by definition.

In the case of initial set of different simple fragments we split them by subsets of isomorphic fragments. After that we process each of subsets as above.

Isomorphic graphs may appear in generation of graphs from simple fragments. It will happen when the connection of some initial fragments and free vertices form a new fragment identical to one of the initial fragment.

EXAMPLE. We need to generate all carbonic acids with molecular formula  $C_6H_{10}O_4$  having the carboxyl fragment: The initial set of vertices is divided by the following classes:



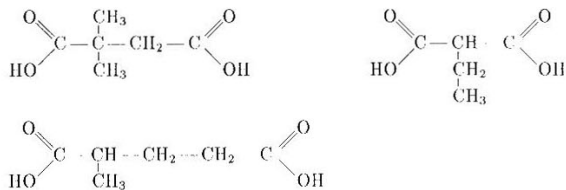
- $V_1$ : 2 vertices O and OH with valency 0;
- $V_2$ : 1 vertex C with valency 1;
- $V_3$ : 5 vertices C with valency 4;
- $V_4$ : 2 vertices O with valency 2;
- $V_5$ : 9 vertices H with valency 1.

The matrix  $R$  of the maximal and necessary multiplicity of edges with omitted hydrogen vertices will look like:

$$R = \left( \begin{array}{ccc|cccc|cccc} \mathbf{0} & \mathbf{0} & \mathbf{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{-2} & \mathbf{-1} & \mathbf{0} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 0 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 \\ \hline 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 & 0 \end{array} \right)$$

All elements of this matrix that belong to adjacency matrix of the carboxyl fragment are labeled bold. The maximal possible multiplicity of edges are equal 2 (that follows from the connectivity of graph and the maximal number of multiple edges).

Using these data as initial, the GENM program generates in 3.1 sec 1974 graphs of carbonic acids, 1971 of them are nonisomorphic. Time for generation of all possible 97394 isomers with molecular formula  $C_6H_{10}O_4$  without any other constraints is 2 min 38 sec on IBM PC AT 386/20 MHz. All twice generated graphs contain two asymmetrically placed carboxyl fragments (see figure below). In the process of generation the constructed carboxyl fragment is regarded as nonidentical to the initial carboxyl fragment.



#### 4. Intermediate fragments.

All intermediate fragments will be considered independently even if some of them may be isomorphic.

Suppose we have an only intermediate fragment  $F = (U, X)$ . Let us divide the set of vertices of the fragment  $F$  into orbits. Define initial classes of vertices for generation in the following way. The first class will include all internal vertices of the fragment and all other classes will contain vertices of fragment's orbits, corresponding to external vertices. Free vertices of a graph form initial classes according to their labels and valencies.

Construct a partially filled matrix  $B = (b_{ij})$ . Define the matrix  $R = (r_{ij})$  by expression (1). These initial data provide generation of graphs containing the intermediate fragment  $F$ .

In the case of multiple intermediate fragments we apply the above procedure consecutively to each intermediate fragment. Unlike the case of single fragments external vertices of intermediate fragments that belong to different, even isomorphic fragments should be placed in different initial classes in order to the matrix  $R$  satisfy (1).

In the case of combination of simple and intermediate fragments we apply the preliminary procedure of distribution of hydrogens among other vertices (just the same that we used for generation of isomers without any initial information about fragments). Note that in some cases there are vertices of fragments with known the number of hydrogen vertices attached. External vertices of these fragments should not have any additional hydrogen vertices. An optional characteristic of fragments indicating the maximal number of additional hydrogens for each external vertice exists.

Isomorphic graphs in generation using intermediate fragments may appear in two cases. First, when connection of initial fragments and free vertices leads to new classes of equiv-

alence. The second case corresponds to set of several identical intermediate fragments. These fragments are considered as nonidentical and their vertices belong to different initial classes.

EXAMPLE. We need to construct all isomers with the molecular formula  $C_6H_{10}N_2$ , containing  $-CH_2-NH-$  fragment without any additional hydrogens attached to it. In this case isomorphic graphs may appear both as a result of construction of the second fragment  $-CH_2-NH-$  and as a result of construction of symmetrical fragments  $-CH_2-NH-CH_2-$  or  $-NH-CH_2-NH-$ . In both fragments external vertices of the initial fragment become equivalent to free vertices attached to it. The generation time for 7644 isomers with the molecular formula  $C_6H_{10}N_2$ , containing  $-CH_2-NH-$  fragment without additional hydrogens attached to it was 7.5 sec, 6739 of them are nonisomorphic.

### 5. Complex fragments.

Suppose we have an only complex fragment  $F = (U, X)$ . Let us divide all of its vertices by orbits. Arrange these orbits (equivalence classes) according to decreasing number of free valencies of vertices in each orbit. Taking into account this partition we construct the canonical adjacency matrix of the fragment  $F$ .

Initial classes of vertices for generation are constructed on the basis of fragment orbits. Free vertices are divided by classes according of its labels and valencies.

Construct a partially filled matrix  $B = (b_{ij})$  including canonical submatrix, corresponding to the fragment  $F$ . Define the matrix  $R = (r_{ij})$  by expression (1).

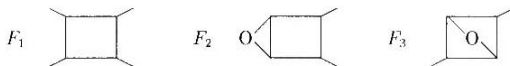
In this case we need to enforce the definition of strongly canonical matrix [1] for correct extension of the algorithm to new class of fragments. Let  $I$  and  $J$  be the sets of indices of the filled and unfilled rows of  $B$ . Let  $S(I, J)$  be the symmetrical group of permutations that independently interchanges filled and unfilled rows and columns of  $B$  within all classes. Define  $S(F, I, J)$  to be a subgroup of  $S(I, J)$  that preserves adjacency submatrix of fragment  $F$ .

DEFINITION 5. The partially filled matrix  $B$  is called *strongly canonical* if it is not increase by action of any permutation from  $S(F, I, J)$ , i.e.  $gBg^{-1} \leq B$  for every  $g \in S(F, I, J)$ .

This change of definition for strongly canonical matrix is necessary for preserving canonical adjacency submatrix of the fragment  $F$  in partially filled matrix  $B$ . In general case some permutations of rows and columns increasing the matrix  $B$  but decreasing the submatrix corresponding to the fragment  $F$  may exist in group  $S(I, J)$ .

It should be noted that vertices of the complex fragment must appear in the first initial classes of vertices. If we have at least one class containing vertices with free valencies in

front of initial classes of vertices corresponding to the complex fragment, many fillings of these rows of the matrix  $B$  become impossible. For example, consider the complex fragment  $F_1$  (see figure below). Evidently, all vertices of this fragment are equivalent and should belong to the same initial class. Suppose we have one more class in front of it containing the vertex labelled  $O$  with valency 2. In this case the only filling of the *piece of stability* [1] corresponding to vertices of the fragment will be (1,1,0,0) because the piece of stability always filled in decreasing order. This filling corresponds to the fragment  $F_2$ . But there exists the fragment  $F_3$  too.



By the same reason we cannot use the procedure of preliminary distribution of ligands among other vertices. This is why the handling of several complex fragments is not as easy as for simple or intermediate fragments. An acceptable solution of this problem was found in dividing the vertices of all complex fragments except the first one by initial classes in different ways. To construct all possible fillings of upper unfilled rows of the matrix  $B$  we divide all external vertices of a fragment by trivial (containing only one vertex) classes. In this case we obtain all possible connections of a complex fragment with preceding fragments. If there exists at least one connection of this type, further we use these trivial classes. Otherwise if the current complex fragment is not connected with preceding complex fragments, we divide its external vertices by classes according to the orbits of this fragment.

It is evident that presence of several complex fragments leads to generation of large enough number of isomorphic graphs. For example we have constructed 32554 isomers with molecular formula  $C_{12}H_{26}N_2O_4$  containing the following fragments for 4 min 2 sec:



There are 15778 nonisomorphic graphs among them.

## 6. Results of generation of some isomer classes.

Let us consider the generation of all isomers with the molecular formula  $C_{10}H_6N_2O_2$  with different sets of initial structure fragments. Table 2 contains the examples of fragments and their numbers. The first row fragments are complex ones, the second row fragments are intermediate and the third row fragments are simple.

Table 1. Results of generation of some isomer classes.

n	Fragments	Number of nonisomorphic graphs	Number of generated graphs	Time min:sec
1	1	76247	76247	5:54.0
2	1, 4	1845	2064	7.6
3	1, 7	802	802	5.0
4	1, 8	3855	3855	16.0
5	1, 9	203	203	0.9
6	1,10	3234	3234	15.0
7	1,12	13192	13192	46.0
8	1,10,12	553	553	2.3
9	2, 4	10264	13584	25.0
10	2, 9	860	860	1.9
11	2,10	19896	20316	52.0
12	2,10,10	456	456	1.5
13	3,10, 8	3147	6185	13.0
14	3,10,11	940	1891	3.5
15	3,10,12	17468	36575	51.0
16	5	52965	52965	1:14.0
17	5, 4	2945	2990	6.6
18	5, 7	1460	1460	3.3
19	5, 9	235	235	0.5
20	5,10	3560	3560	6.5
21	6	3838	3838	5.9
22	6, 4	215	220	0.5
23	6, 7	166	166	0.5
24	6, 8	328	328	0.8
25	6, 9	27	27	0.1
26	6,10	392	392	0.9
27	6,12	665	665	1.1
28	7, 7,11	4457	8825	15.0
29	7,10,11	12295	12295	19.0
30	9,11	5866	5866	8.7
31	10,10,11	7533	7533	13.3



Table 2. Examples of fragments.













			
			
			

Table 1 contains information about generation of various classes of isomers defined by the sets of fragments. We do not set any constraints on the number of additional hydrogen vertices for external vertices of fragments. The sets of fragments were designed for demonstration of work of program GENM with various initial data. We made the different combinations of complex, intermediate and simple fragments. The fragments have different number of cycles and multiple edges.

As mentioned above isomorphic graphs are generated in the cases of setting several complex fragments (examples n.2 and n.9 in Table 1), several identical intermediate fragments (n.28) and in case of appearing new classes of equivalence for the set of vertices (n.13, n.14 and n.15).

## 7. Conclusion.

The data listed in Table 1 afford us to evaluate both high efficiency and good results of applying the described method of setting information about necessary fragments. It should be noted that the adjacency matrices of generated graphs include adjacency submatrices of the initial fragments. It affords to use the generated graphs in subsequent investigations immediately.

The algorithm of graph generation based on the set of nonoverlapping initial fragments may be successfully used as a good instrument in structure elucidation systems using molecular spectroscopy data bases, in molecular design systems, in expert systems for construction of new substances with predefined properties.

**Acknowledgment.**

I would like to thank A.A.Dobrynin for his valuable remarks and suggestions and also ISF Long-Term Research Grant Program (No. NCH2-7227-0925) for financial support.

**References.**

1. S.G.Molodtsov, Computer-Aided Generation of Molecular Graphs, MATCH, in press.
2. W.Bremser and R.Neudert, Automation in the Spectroscopic Laboratory - Solutions and Perspectives, European Spect.News, No.75 (1987) 10-27.
3. K.S.Lebedev, Infrared and Mass Spectral Databases in Structure Elucidation of Organic Compounds, Zh.Anal.Khimii, 48 (1993) 851-863.
4. A.J.Stuper, W.E.Brugger and P.C.Jurs, Computer Assisted Studies of Chemical Structure and Biological Function, John Wiley & Sons, 1979.