Journal of Chemical Information and Modeling (2006), 46, 52

A Simple Algorithm for Unique Representation of Chemical Structures - Cyclic/ Acyclic Functionalized Achiral Hydrocarbons

Yenamandra S. Prabhakar[1] and Krishnan Balasubramanian[2]*

[1]Medicinal and Process Chemistry Division,
Central Drug Research Institute, Lucknow-226 001 (U.P), India
E-mail: yenpra@yahoo.com
[2]University of California, Chemistry and Material Science Directorate,
Lawrence Livermore National Laboratory, Livermore, CA 94550, USA
Department of Mathematics & Computer Science, California State University, East Bay,
Hayward, California 94542-3092
E-mail: balu@mcs.csuhayward.edu

Abstract
An algorithm, based on 'vertex priority values' has been proposed to uniquely sequence and represent connectivity matrix of chemical structures of cyclic/ acyclic functionalized achiral hydrocarbons and their derivatives. In this method 'vertex priority values' have been assigned in terms of atomic weights, subgraph lengths, loops, and heteroatom contents. Subsequently the terminal vertices have been considered upon completing the sequencing of the core vertices. This approach provides a multilayered connectivity graph, which can be put to use in comparing two or more structures or parts thereof for any given purpose. Furthermore the basic vertex connection tables generated here are useful in the computation of characteristic matrices/ topological indices, automorphism groups, and in storing, sorting and retrieving of chemical structures from databases.

Keywords. Chemical structure representation; graph theory; connectivity table; cyclic/ acyclic functionalized achiral hydrocarbons.

* Corresponding author

1 INTRODUCTION

Graph isomorphism and vertex-based graph automorphism partitioning problems have a long history in both mathematical and chemical literatures.[1-15] In graph theoretical connotation a chemical structure is a collection of vertices (atoms) connected by means of edges (bonds) in a predefined fashion. This has a diverse role in the elucidation and comprehension of various physical, chemical and biological behavioral aspects of chemical entities. Quantum chemistry, spectroscopy, molecular symmetry and physical/ biological property prediction are a few aspects among many others, which revolve around graph theoretical concepts of chemical structure.[1-7] In all the graph theoretical approaches, partitioning of vertices is crucial and this gives rise to characteristic applications to the defined vertex partitioning procedure. Among many graph theoretical applications, Balasubramanian and co-workers[4,5] have demonstrated the usefulness of automorphism partitioning of vertices in spectroscopy and quantum chemistry. Moreover vertex-partitioning procedures have significant role in chemical database operations. Most of the chemical database operations involve sequenced chemical structures. In the absence of any priority (for a vertex), a chemical graph with **n** vertices can be sequenced in **n!** (**factorial n**) ways. If the graph has no symmetry, then each one of these **n!** representations will be different from the rest. In this environment, sequencing of the chemical graph based on a preset vertex priority offers a unique advantage to many complex problems like similarity searches, etc. There are a number of methods for unique sequencing of chemical graphs, e.g., Morgan's algorithm,[8] Wipke and Dyott[1] Wiswesser Line Notation (WLN),[9] Balaban's hierarchically ordered extended connectivities (HOC) procedure,[10] Randić's canonical labeling method,[11] etc. Many of these techniques while are very useful they have difficulties with highly transitive graphs. In one of the oldest Morgan's algorithm, the current atom is the one with the highest extended connectivity (EC) value, and if there are any attachments to the current atom which have not been assigned sequence numbers, then they are assigned sequence numbers in a decreasing order of EC values of the attachment, which includes terminal attachments (EC value is one) even before other atoms with higher EC values. Here we present a rule-based algorithm to partition the vertices of chemical graphs with an approach that we call "inside-in and outside-out". This results in unique sequence for a given chemical structure according to the vertex priorities generated in the algorithm, where the terminal vertices are considered separately after completely sequencing the core vertices. In this method the progress of sequencing of core vertices take place along the uninterrupted vertex paths. Also, this sequencing procedure generates traditional connection table of the chemical graph and finds application in different chemical graph related operations. The method can also be extended to generate the automorphisms of a given structure or a graph.

2 METHOD

In a chemical graph, a hydrogen suppressed chemical structure, it is well known that the connectivity value of a vertex (an atom) is equal to the number of edges (bonds) with which it is joined to all other immediate neighboring vertices (non-hydrogen atoms). In

these graphs, an edge represents either a sigma-bond or 'sigma + pi'-bonds. An algorithm has been designed to sequentially prioritize the vertices of chemical graphs in a hierarchical manner based on the connectivity values and several other associated characteristics, such as atomic weights, sub-graph lengths, loops, heteroatom content. It may be mentioned that a loop in this algorithm represents a cyclic system identified (or encountered) in direction of vertex sequencing and/ or propagation. Here, the vertices are sequenced in a decreasing order of priority - that is, a vertex with higher priority will be addressed and labeled first. In this procedure, the connectivity values, number of sigma bonds as well as 'sigma + pi' bonds, of all vertices will be computed and used for prioritization. Based on the vertex connectivity values (sigma bond alone), the vertices of the graph will be divided into two groups – one group corresponds to vertices with connectivity values more than or equal to two and the other group corresponds to vertices with connectivity values equal to unity. The vertices with connectivity values more than or equal to two form the core of the graph and will be prioritized successively based on their connectivity values and other associated characteristics. Once the priority of a vertex in the graph is identified and fixed, the sequencing of subsequent vertices will proceed by locating a vertex with next highest priority that is directly connected to the just prioritized vertex. Once the sequencing process encounters an 'end point' in the current fragment propagation 'direction', a successive stepwise backward integration starts to vertex **1** to prioritize any vertices left behind. The sequencing 'end point' on any current fragment propagation 'direction' arises due to the completion of prioritization of all vertices with connectivity values more than or equal to two in that 'direction'. After fixing the priorities of vertices with connectivity values more than equal to two (core vertices), the priorities of vertices with unit connectivity values (terminal vertices) will be fixed using the same priority rules of core vertices. The successive steps of the sequencing algorithm are as follows:

Step Computation
Decision

1 Compute connectivity (**Cn**) (sigma bond) of all vertices (**Vts**) (non-hydrogen atoms) in a given hydrogen suppressed chemical graph. Segregate all **Vts** into two groups – one with **Cn** values more than or equal to two and the other with **Cn** value equal to one. Consider all **Vts** with **Cn** values of two or more as competing **Vts**.

2 Find number of **Vts** with the highest **Cn** (sigma bond only).
If only one **Vt**, then go to step **13**

3 Find number of **Vts** with highest **Cn** (sigma + pi bonds)
If only one **Vt**, then go to step **13**

4 Find atomic weights (**Wt**) of **Vts** with highest **Cn** (sigma + pi bonds)
If only one **Vt**, then go to step **13**

5 Divide the molecule into fragments (**Frs**) in such a way that each **Fr** contains one and only one competing **Vt** with maximum **Wt** and highest **Cn**.

6 Find the maximum **length** of **Frs**.
If only one **Fr**, then go to step **13**

7 Find the highest number of **loops** in **Frs** with maximum **length**
If only one **Fr**, then go to step **13**

8 Among **Frs** with maximum **length** and highest number of **loops**, find maximum **chain length** with competing **Vt**.
If only one **Fr**, then go to step **13**

9 Among **Frs** with maximum **length**, highest number of **loops** and maximum **chain length** with competing **Vt**, find the maximum number of **heteroatoms**.
If only one **Fr**, then go to step **13**

10 Among **Frs** with maximum **length**, highest number of **loops**, maximum **chain length** with competing **Vt** and also **heteroatoms**, find the maximum **weight** of Frs.
If only one Fr, then go to step **13**

11 Compute distance matrices for **Frs** with maximum **length**, and having the highest number of **loops**, maximum **chain length** with competing **Vt**, highest number of **heteroatoms** and also **weight**. Compare the **distances** between competing **Vts** and **heteroatoms** of each **Fr**. Find **Frs** with compactly connected competing **Vt** and **heteroatoms**.
If only one Fr then go to step **13**

12 Element of symmetry exists. Arbitrarily consider one of the competing **Vts**.

13 Prioritize (label) the **Vt** as 1(one) (subsequently with successive numbers).
If all **Vts** with **Cn** values more than or equal to two are prioritized then go to step **14** else consider the **Vts** connected to the just prioritized vertex as competing **Vts** after excluding already prioritized **Vts**, if any, from the list and go to step **2**.

14 Prioritize the **Vts** with **Cn** value of unity according to the priority set by competing **Vt** (steps **3** and **4**), **Cn** of immediate neighboring **Vt**.

15 End of graph sequencing.

2.1 Chemical Data

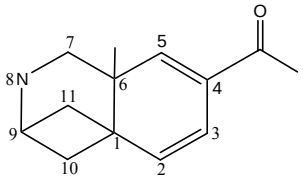
This algorithm has been used to prioritize 1-(6-methyl-8-aza-tricyclo[7.1.1.$0^{1,6}$]undeca-2,4-dien-4-yl)-ethanone (Figure 1) and toluene (Figure 2). Also, a few selected examples representing the chemical graphs of cyclic hydrocarbons with high degree of symmetry and loops - namely tetradecahydrocyclopenta[*fg*]acenaphthylene (Figure 3),

octadecahydro-chrysene (Figure 4), octadecahydrobenzo[*c*]phenanthrene (Figure 5) and cubane (Figure 6) - have been used to explain the procedure. For all the chemical graphs, the vertex priorities assigned by the present algorithm have been compared with those of Morgan's algorithm.

3 RESULTS AND DISCUSSION

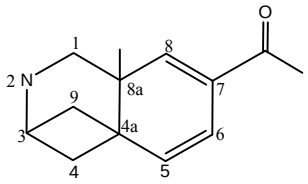
In the IUPAC system of nomenclature of the compound 1-(6-methyl-8-aza-tricyclo[7.1.1.$0^{1,6}$]undeca-2,4-dien-4-yl)-ethanone (Figure 1), the priorities of molecular subunits will result in its expression as the ethanone system (the main frame) carrying the (6-methyl-8-aza-tricyclo[7.1.1.$0^{1,6}$]undeca-2,4-dien-4-yl) fragment on it. Accordingly, Figure 1a shows the numbering (priorities) of the various centres (vertices) of 8-aza-tricyclo[7.1.1.$0^{1,6}$]undeca-2,4-dien-4-yl fragment. The same molecule, in a variant of IUPAC nomenclature procedure, may also be expressed as 1-(1,3,4,8a-tetrahydro-8a-methyl-2*H*-3,4a-methanoisoquinolin-7yl)-ethanone. Figure 1b shows the numbering of various centres of 1,3,4,8a-tetrahydro-2*H*-3,4a-methanoisoquinolin-7yl fragment of the same molecule. The IUPAC and other similar nomenclature procedures are based on a very high level of human visualisation perceptions and comprehension. One motivation for the investigations into the alternative procedures of vertex prioritisation approaches is to embed the highest possible information into the graph system at the earliest possible level of vertex sequencing. With this in view the present algorithm has been explained in comparison with the Morgan's algorithm in sequencing the vertices of chemical graphs.

Chemical Formula Representation



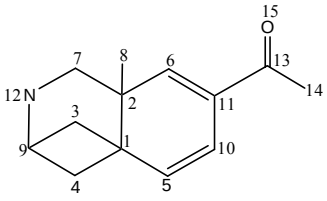
(IUPAC)

1a

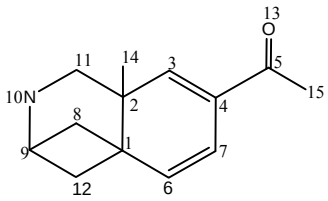


(A variant of IUPAC)

1b



1c



1d

Graphical Representation

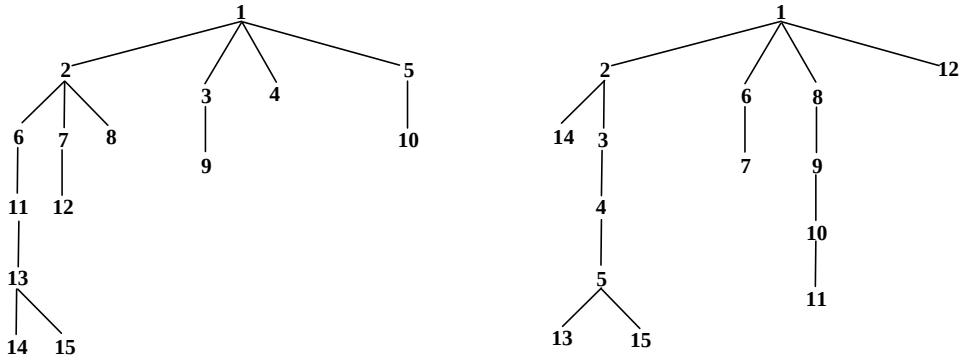


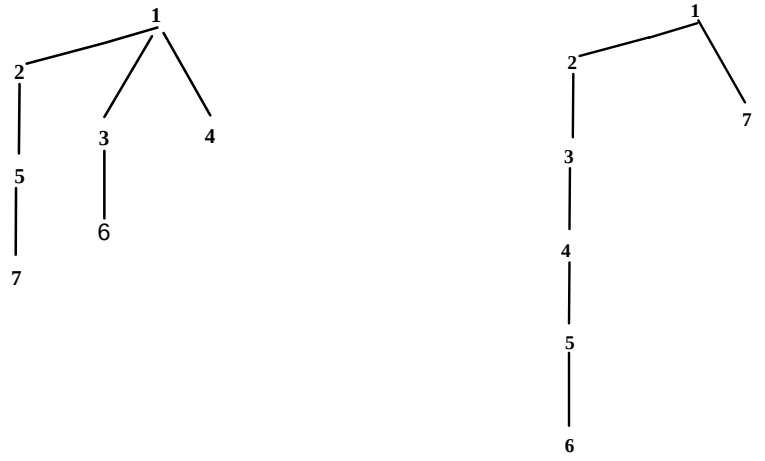
Figure 1: Vertex prioritization of 1-(6-methyl-8-aza-tricyclo[7.1.1.0^{1,6}]undeca-2,4-dien-4-yl)-ethanone in IUPAC System, Morgan's and Present algorithms

The vertex prioritizations, according to Morgan's as well as the present algorithm, of the hydrogen suppressed chemical graph of 1-(6-methyl-8-aza-tricyclo[7.1.1.0^{1,6}]undeca-2,4-dien-4-yl)-ethanone have been shown in the chemical formula representation form (Figures 1c and 1d) as well as in graphical representation form (Figures 1e and 1f). In this example, in terms of core vertices, a vertex path **1-2-3-4-5** of present algorithm (Figure 1f) is identical with that of Morgan's vertex path **1-2-6-11-13** (Figure 1e)**.** In Morgan's algorithm, the oxygen of the ethanone fragment (Figure 1c) has been sequenced as vertex **15** and its methyl carbon as vertex **14.** As due consideration has been given to the atom types in the present algorithm, the oxygen of the ethanone fragment (Figure 1d) gets higher priority (vertex **13**) over the methyl carbon (vertex **15**). Also in the present algorithm the methyl carbon (vertex **14**) at the ring junction (vertex **2**) and methyl carbon of ethanone fragment (vertex **15**) (Figure 1d) have been demarcated with distinct priorities. The vertex path **1-8-9-10-11** of 1-(6-methyl-8-aza-tricyclo[7.1.1.0^{1,6}]undeca-2,4-dien-4-yl)-ethanone in Figure 1f is characteristic to the present algorithm. In Morgan's algorithm, vertices corresponding to this path have been distributed in two fragments namely the vertex path **1-3-9** and the vertex path **1-2-7-12**. Moreover further scrutiny of Figures 1e and 1f indicate that the present algorithm leads to less segmented (or less fragmented) and more compact graphs.

Chemical Formula Representation

2a **2b**

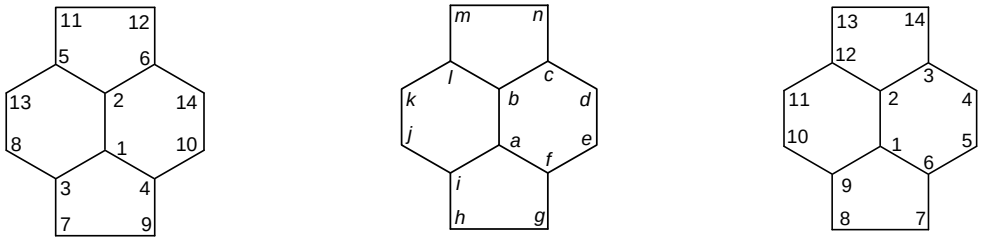
Graphical Representation



(Morgan's) **(Present)**
2c **2d**

Figure 2: Vertex prioritization of toluene in Morgan's and Present algorithms

Similarly in toluene, in Morgan's algorithm the methyl carbon of toluene has been sequenced as vertex **4** much before the other carbons of phenyl ring (Figure 2a). In the present algorithm, this methyl carbon has been sequenced as the end as vertex **7** (Figure 2b). Also, the graph of toluene is less segmented in the present algorithm (Figure 2d) when compared to that of Morgan's algorithm (Figure 2c). For the purpose of brevity the remaining illustrations of polycyclichydrocarbons have been limited to the structures of the chemical formula only. In these cases, the Figure numbers suffixed with '**a**' show the priorities of the vertices according to Morgan's algorithm and those suffix with '**b**' correspond to that of present algorithm. Unlike the previous cases, all the following ones have varying degree of symmetry embedded in them. Keeping this in view, the vertex prioritisation in these cases has been explained with the help of alphabet labelled (in italics) chemical graphs.

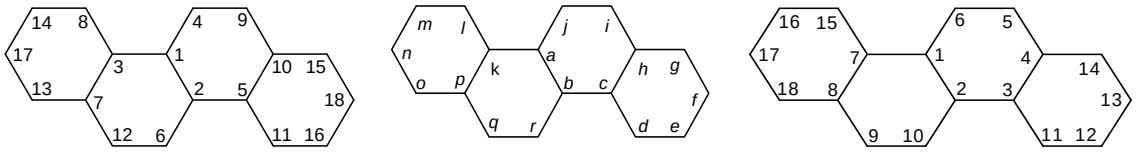


(Morgan's) **(Present)**
3a **3** **3b**

Figure 3: Vertex prioritization of tetradecahydrocyclopenta[*fg*]acenaphthylene in Morgan's and Present algorithms

In tetradecahydrocyclopenta[*fg*]acenaphthylene (Figure 3), due to the symmetry in the graph, the vertices '*a*' and '*b*' have equal priority. Arbitrarily, vertex '*a*' has been assigned as 1 and '*b*' has automatically become vertex 2. Again for the same reason of

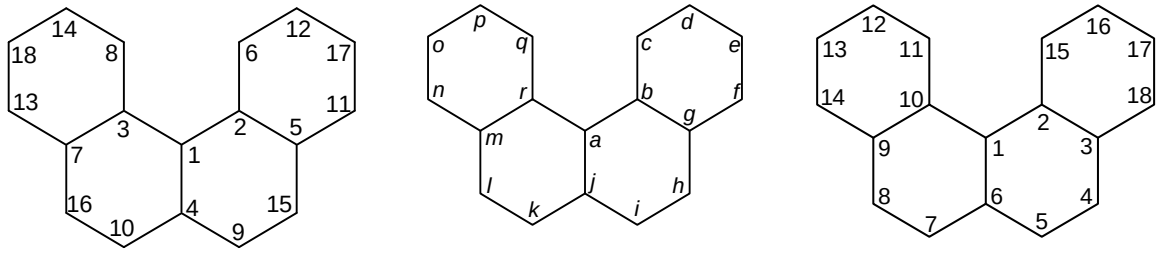
symmetry in the system the next competing vertices '*c*' and '*l*' have equal priority to take the position 3. In the arbitration, vertex '*c*' has been assigned as 3. The vertices competing for prioritisation in the fourth position are'*d*' and '*n*'. Here, '*d*' gets priority over '*n*' because of the fact that the smallest loop involving '*d*' is larger (6 member ring) than that of its competing vertex '*n*' (5 member ring). It may be mentioned that a loop in this algorithm represents a cyclic system. In this molecule, the priority of '*d*' over '*n*' is a clear case of the participation of loop size in the decision making process. In the vertex propagation direction of '*d*', after two vertices ('*e*' and '*f*'), the propagation encounters a vertex '*a*' which is already a prioritised vertex and thereby existence of a loop (6 member ring) comes into effect. For the same two vertices propagation in '*n*' chain ('*m*' and '*l*'), here also the propagation recognises the encounter of prioritised vertex '*b*' and thereby existence of a loop (in this direction, 5 member ring) comes into effect. As the 6 member loop is larger than that of 5, the vertex which is part of the former one will get the priority. Hence '*d*' has been labelled as 4. The prioritisation of the remaining vertices proceeds in the same direction till the end.



(Morgan's) 4a **4** **(Present 4b**

Figure 4: Vertex prioritization of octadecahydrochrysene in Morgan's and Present algorithms

In octadecahydrochrysene (Figure 4) also, due to the element of symmetry, both '*a***'** and '*b***'** have equal priority. Hence, arbitrarily, '*a***'** has been assigned as **1** and '*b***'** has been assigned as **2.** After this, '*c*' and '*r*' become the competing vertices; as '*c*' is more heavily substituted (when compared to '*r*'), it has been assigned 3. The next competing vertices are '*d*' and '*h*'; due to the heavier substitution '*h*' becomes vertex 4 . Now, the competing vertices are '*i*' and '*g*'. The vertex propagation in the '*i*' direction gets early branching compared to that of '*g*' chain direction. Accordingly, '*i*' has been assigned as 5. The succeeding vertex '*j*' gets 6. And also the chain propagation reaches an end point. At this stage the propagation traverse back for the vertex with next highest priority, which is in this case is '*k*' and gets assignment as 7. For the same reasons as discussed above, vertices '*p*', '*q*' and '*r*' becomes 8, 9 and 10, respectively. This leads to the second end point. Now, the vertices to be prioritised stem from '*c*'(3), '*h*'(4), '*k*'(7) and '*p*'(8). Being equal on all counts, the new vertex prioritisation starts adjacent to the vertex with highest priority (lowest vertex number), that is '*c*'(3) and proceeds through '*d*'(11) to '*g*'(14). This leads to third end point. The remaining vertices '*l*' to '*o*' are prioritised as 15 to 18 in the same manner. Interestingly, in this case, the inner two rings have remained inside flanked by the peripheral rings.

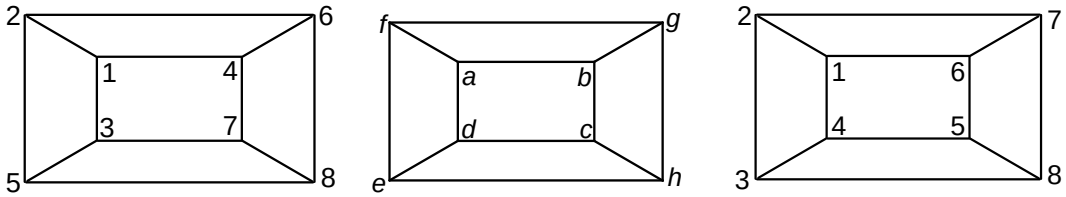


(Morgan's) **(Present)**

5a **5** **5b**

Figure 5: Vertex prioritization of octadecahydrobenzo[*c*]phenanthrene in Morgan's and Present algorithms

In octadecahydrobenzo[*c*]phenanthrene (Figure 5) vertex '*a*' is the vertex with highest priority and automatically has been assigned as 1. The vertices '*b*' and '*r*' attached to '*a*' have the equal priority. Arbitrarily, vertex '*b*' has been assigned as 2. After this, the competing vertices are '*c*' and '*g*'; here '*g*' gets priority over '*c*' due to the formers heaviness, hence labelled as 3. The next available vertices are '*f*' and '*h*'. The vertex of choice is '*h*' as in this direction early heaver substitution is encountered. Accordingly '*h*' has been assigned as 4. The vertex propagation proceeds in the same direction as shown in Figure 5b till reaching first end point after prioritising vertex '*n*' as 14. In the backward integration process, the vertices '*c*' to '*f*' will be prioritised as 15 to 18. In this case, only one end point has been encountered before complete prioritisation of all vertices. Here also, similar to the previous case, the inside rings have been prioritised much before the outside rings.



(Morgan's) **(Present)**

6a **6** **6b**

Figure 6: Vertex prioritization of cubane in Morgan's and Present algorithms.

After some initial arbitration, the algorithm offers scope to prioritise even the symmetric graphs like cubane (Figure 6). In cubane, all eight vertices are identical. In the first arbitration, vertex '*a*' has been assigned as 1. After fixing vertex 1, three vertices, '*f*', '*b*' and '*d*', will become competing ones for the second position. In the second arbitration, '*f*'

has been assigned as vertex 2. In the third step the vertex to be prioritised will be either '*e*' or '*g*'. Being equal in all respects either of '*e*' or '*g*' can be opted as vertex 3. In the present case '*e*' has been assigned as 3. After fixing the third vertex, the remaining five vertices are amenable for prioritisation without any prior arbitration. It follows as shown here. The vertices '*d*' and '*h*' await prioritisation after the vertex '*e*' (3). Now, the proximity of vertex '*d*' to the already prioritised vertex 1'*a*' (formation of loop also), gives it edge over its competitor vertex '*h*'; accordingly '*d*' has been assigned as vertex 4. Now, vertex '*c*' becomes the 5^{th} one in the line. Similarly, the proximity of vertex '*b*' to the already prioritised vertex 1'*a*', gives it edge over its competitor vertex '*h*' (this is also near to vertex '*e*' (3), but of lowered priority); accordingly '*b*' becomes vertex 6. The remaining two vertices, '*g*' and '*h*' will take the positions 7 and 8, respectively, to complete the cubane.

In both the procedures, the Morgan's and the present algorithm, the vertex with maximum connectivity gets the highest priority. In Morgan's algorithm, irrespective of the vertex position, that is, core or terminal (or peripheral) vertex in the graph, all the connected vertices of the just prioritized vertices will be sequenced simultaneously. However, in the algorithm described here the vertices of the graph are at first demarcated in terms of core vertices and terminal vertices. The prioritization of the terminal vertices will be addressed after sequencing the core vertices. This provides an easy handle to study the core and terminal vertices. Moreover, this approach provides a multilayered connectivity graphs, which can be put to use in comparing two or more structures or parts thereof for any given purpose. Also, the algorithm allows the progress of vertex sequencing in a specific direction till exhausting all the core vertices of the subgraph under consideration. This facilitates the identification of sub-graphs with maximum lengths and allows their compact representation.

4 CONCLUSIONS

The present algorithm provides an efficient and more convenient technique to examine the terminal as well as core vertices of given chemical graphs. If the graph under consideration exhibits automorphic symmetry, this procedure arbitrarily considers one of the equivalent vertices as the vertex of choice for the propagation of sequence of the graph. Consequently, equivalent vertices would have the same prioritization paving the way for automatic generation of automorphism group of the graph. The sequential identification of vertices of subgraphs facilitates the formation of connectivity tables. This can be used for the computation of characteristic matrices, polynomials and several topological indices. Moreover the method would find application in storing, sorting and retrieving of chemical structures and databases. As this typically demarcates the core and terminal vertices in graphs, it can be put an easy use in comparing two or more structures or parts thereof for any given purpose. Another important feature of our current algorithm is that it considers heteroatoms, loops in graphs and weighted graphs and thus expanding its applicability to a wider spectrum of chemical graphs. Thus we believe that the approach developed here has several advantages over other algorithms including Morgan's algorithm. The applicability of the current algorithm to directed graphs might

also be explored.

5 ACKNOWLEDGEMENTS

The research at CalState, Eastbay was supported by the National Science Foundation under Grant No. CHE-0236434. The work at LLNL was performed in part under the auspices of the US department of Energy by the University of California, LLNL under contract number W-7405-Eng-48.

6 REFERENCES

[1] Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm, *J. Am. Chem. Soc.* **1974,** *96*, 4834-4842.
[2] Kier, L.B.; Hall, L.H. "Molecular Connectivity in Chemistry and Drug Research" Academic Press, Inc., New York, **1976**.
[3] Randić, M.; Brissey, G.M.; Wilkins, C.W. Computer Perception of Topological Symmetry via Canonical Numbering of Atoms, *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 52-59.
[4] Balasubramanian, K. Application of Combinatorics and Graph Theory to Spectroscopy and Quantum Chemistry, *Chem. Rev.* **1985**, *85*, 599-618.
[5] Liu, X.; Balasubramanian, K.; Munk, M. E. Computer-Assisted Graph-Theoretical Construction of ^{13}CNMR Signal and Intensity Patterns, *J. Magn. Reson.* **1990,** *87*, 457-474.
[6] Liu, X.; Balasubramanian, K; Munk, M. E. Computational Techniques for Vertex Partitioning of Graphs, *J. Chem. Inf. Comput. Sci.* **1990,** *30*, 263-269.
[7] Razinger, M; Balasubramanian, K.; Munk, M. E. Graph Automorphism Perception Algorithms in Computer-Enhanced Structure Elucidation, *J. Chem. Inf. Comput. Sci.* **1993,** *33*, 197-201.
[8] Morgan, H.L. The Generation of a Unique Machine Description for Chemical Structure – A Technique Developed at Chemical Abstracts Service, *J. Chem. Doc.* **1965,** *5*, 107-113.
[9] Wiswesser, W.J. *A Line Formula Chemical Notation*, T.Y. Crowell Comp, New York, 1954.
[10] Balaban, A. T.; Mekenyan, O.; Bonchev, D. Unique Description of Chemical Structures Based on Hierarchically Ordered Extended Connectivities (HOC Procedures). I. Algorithms for Finding Graph Orbits and Canonical Numbering of Atoms, *J. Comput. Chem.* **1985**, *6*, 538-551.
[11] Randić, M. On Canonical Numbering of Atom in a Molecule and Graph Isomorphism, *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 171-180
[12] Herndon, W. C. In *Chemical Applications of Graph Theory*; King, R. B. Ed.; Elsevier: Amsterdam, 1983, Vol. 28, pp231-242.
[13] Balasubramanian, K. Symmetry Groups of Chemical Graphs, *Int. J. Qunat. Chem.* **1982**, *21*, 411-418.
[14] Read, R.C.; Corneil, D. G. Graph Isomorphism Disease, *J. Graph. Theory*, **1977**, *1*, 339-363.
[15] King, R. B. In *Chemical Applications of Graph Theory*; King, R. B. Ed.; Elsevier: Amsterdam, 1983, Vol. 28, pp108-122.