Journal of Mathematical Chemistry 11(1992)259–270

MOLECULAR TOPOLOGY. VIII: CENTRICITIES IN MOLECULAR GRAPHS. THE MOLCEN ALGORITHM

M.V. DIUDEA*, D. HORVATH, I.E. KACSO and O.M. MINAILIUC
Department of Chemistry, "Babes-Bolyai" University, 3400 Cluj, Roumania

B. PARV
Department of Mathematics, "Babes-Bolyai" University, 3400 Cluj, Roumania

Abstract

Local and global centricities and corresponding complexity centricities are derived on the basis of matrices B (layer matrix of vertex degrees) by using appropriate distance operators. The MOLCEN algorithm computes these centricities by means of line derivatives (L_n) of graphs. It provides reliable centric ordering of subgraphs of various length in molecular graphs. The algorithm is implemented on a TURBO-PASCAL, TOPIND program and is exemplified within a set of molecular graphs.

1. Introduction

The concept of a graph center, initially introduced for acyclic graphs and later extended for cyclic graphs [1], was used in the coding of chemical structures [2] or chemical reactions [3], or also in chemical nomenclature [4]. Numeric centric indices [5–7] were successfully used in structure–property correlations. The centric ordering of vertices/edges is of interest in graph isomorphism [8].

Before detailing the graph center problem, some background definitions are needed: a graph $G = (V, E)$ is an ordered pair of two sets, V (vertices) and E (edges). The topological distance between vertices i and $j \in V(G)$, d_{ij}, represents the number of edges along the shortest path connecting the two vertices. For any vertex i, the maximal distance to any other vertex j is called vertex eccentricity, $e_i = \max_{ij \in V} d_{ij}$. The sum of distances of vertex i to all other graph vertices $D_i = \sum_j d_{ij}$ is called the vertex distance sum (see refs. [9, 10]). The radius of the graph, $r(G)$, represents the minimal eccentricity among the vertex eccentricities, $r(G) = \min_{i \in V} e_i$ [1]. A graph construction (a non-numeric characteristic) collecting a vertex set $i \in V(G)$ which obeys $e_i = r(G)$ is called the center of the graph [1, 10, 11]. Such a definition for the graph center is proper for acyclic graphs but not for cyclic/polycyclic structures, which require [9] "the central vertices to belong to the same orbit of the graph's automorphism group".

*To whom correspondence should be addressed.

Three approaches were proposed by Bonchev et al. [9] with the aim of solving the problem of the graph center in cyclic/polycyclic structures:

(i) *1D–3D criteria* form an iterative approach based on matrix F (layer matrix of vertex neighbourhood [11] or distance frequency matrix [7] or on vertex distance code (VDC), as was originally defined by Bonchev et al. [6]. The 1D–3D criteria are as follows:

- *1D: minimum vertex eccentricity*, $e_i = min$. This is just the radius of the graph [1] involved in the classical definition of the graph center (see above).
- *2D: minimum vertex distance sum*, $D_i = min$.
- *3D: minimum number f_{ij} in F matrix (or k in VDC) of occurrence of the largest distance*, $f_{ij,max} = min$. If $f_{ij,\max} = f_{kj,\max}$; $i \neq k$, the next largest distance ($j_{\max} - 1$) is considered, and so on. By deleting all but the central vertices, cf. 1D–3D, one obtains a kernel of G. Criteria 1D–3D are iterated over the kernel until the subsequent iterations fail to further reduce the number of central vertices. The result is a graph center (or a polycenter). Criteria 1D–3D are applied hierarchically [6].

(ii) *1P–3P criteria* are built by analogy to the 1D–3D criteria by replacing the distance notion with that of the path notion [12]. The application of 1P–3P criteria results in an oligocenter.

(iii) *IVEC algorithm* (iterative vertex and edge centricity) [9]) takes into account both the metric properties and vertex–edge incidence. Its basis assumption is "central are those vertices that are incident to the most central edges; conversely, central are those edges that are incident to the most central vertices". IVEC is built on VDC and EDC (vertex and edge distance code, respectively) and 1D–3D criteria.

The application of IVEC results in a different centric ordering of vertices/edges versus the other approaches. It finds correctly vertex and edge orbits of the graph's automorphism group [9] and is particularly useful in polycyclic graph analysis.

In this paper, we present a new approach for finding the graph center (with results comparable to those given by IVEC) and the centricity of any subgraph in the graph, of length ranging between zero and the diameter of the graph. We also introduce the notion of graph complexity center and we discuss the two graph constructions in terms of our MOLCEN algorithm, within a set of molecular graphs.

2. Vertex centricities

In the previous works of this series [7,10], we have shown that the graph center, in view of 1D–3D criteria, could easily be found by operating on the matrices F with an operator that enhances the contribution of more remote vertices (a distance progressive operator). This results in a parameter whose meaning is

close to the vertex eccentricity. In this paper, we operate with such an operator on the B matrix (layer matrix of vertex degrees [13]):

$$BE_i = \sum_{j=2}^{j_{max}} (b_{ij})^{j/d_{req}}, \tag{1}$$

where b_{ij} are entries in the B matrix and are defined as

$$b_{ij} = \sum_{k, d_{ik}=j-1} dg_k, \tag{2}$$

with dg_k being the degrees of vertices on a shell around the vertex i at a given distance $d_{ik} = j - 1$; d_{req} is a "request" distance, for unique description within a set of molecules, according to eq. (1) (usually twice the largest diameter in the considered set of graphs).

It is obvious that the inverse of BE_i should express the centricity (BC) of vertex i:

$$BC_i = [BE_i]^{-1}. \tag{3}$$

As we stated in ref. [13], the vertex regressive degree R_i defined on the basis of the B matrix provides information about local branching/complexity. Due to its distance regressive character, R_i is sensitive to distance. Hence, the center(s) of graph complexity, defined as the vertex (vertices) with maximal R_i-value(s), appears conceivable.

This kind of centricity is computed according to

$$BCX_i = [R_i + L_i]w_i, \tag{4}$$

where w_i is a weighting factor accounting for heteroatoms (see below); R_i is the regressive degree and L_i is the local contribution for multiple bonds:

$$R_i = \sum_{j=1}^{j_{max}} b_{ij} \times 10^{1-j}, \tag{5}$$

$$L_i = f_i \sum_{j=1}^{2} b_{ij} \times 10^{-j}, \tag{6}$$

$$f_i = \sum_{j=1}^{\upsilon} (c_{ij} - 1), \tag{7}$$

where c_{ij} denotes the non-zero entries in the connectivity matrix C and υ is the number of vertices in G.

3. The MOLCEN algorithm

Our algorithm makes use of the line derivatives of molecular graphs (hydrogen-depleted and arbitrarily numbered). The line derivative of a graph is constructed by representing its lines by points, and then joining two such points with a line if the lines they represent are adjacent in the orginal graph (which is a zero-order derivative $L_0(G)$). By iterating this procedure n times, one obtains $L_n(G)$.

The number of vertices $v(L_{n+1})$ and edges $e(L_{n+1})$ in the derivative $L_{n+1}(G)$ is given by the following relations [14, 15]:

$$v(L_{n+1}) = e(L_n), \tag{8}$$

$$e(L_{n+1}) = -e(L_n) + 1/2 \sum_{i \in L_n} [dg_i]^2, \tag{9}$$

where dg_i stands for the classical vertex degree in $L_n(G)$.

The vertices of $L_n(G)$, denoted by S_{ni} ($i \in V(L_n(G))$), represent subgraphs that trace the history of $L_n(G)$ building (see also ref. [16]) by starting from $L_0(G)$.

A problem arises when the complexity operator BCX is used: how to weight the line derivatives, which are virtual graphs versus $L_0(G)$? In molecular graphs, the weighting factor w_i must discriminate the chemical nature of atoms. In this respect, we have chosen the valence group electronegativities as defined in ref. [17], but w_i could be rather arbitrarily chosen to stress the importance of a given vertex. Hence, in $L_n(G)$ the geometric mean of electronegativities of the vertices belonging to the real subgraphs represented by S_{ni} is made in suggesting the concept of atomic electronegativities equalization when the atoms joined in a molecule (see ref. [18]). We extended this idea also in the case of arbitrary w_i factors.

The multibond factor L_i was calculated (by analogy with w_i) as the mean square root of all L_i contributors in an S_{ni}.

The MOLCEN algorithm consists of the following five steps:

Step 1 computes the $I(S_{ni})$ parameters according to eqs. (1)–(7) for $L_0 \ldots L_n(G)$ until $n_{max} = d(G) + 1$. This limit for the rank of $L_n(G)$ is related to the largest path (the diameter $d(G)$) whose centricity is tested assuming that, in general, a single line derivative is enough to solve the degeneracy in vertex centricities (see also the IVEC algorithm [9]).

Step 2 computes the normalized NI parameters by dividing the $I(S_{ni})$ values by the maximal value in the vector $I(S_{ni \in V(L_n)})$, cf. eq. (10):

$$NI(S_{ni}) = I(S_{ni})/\max I(S_{ni \in V(L_n)}). \tag{10}$$

Step 3 evaluates the contribution of the centricities of S_{ni} subgraphs to the centricity of the included S_{mj} subgraphs ($m \leq n$)., in agreement with the IVEC concept:

$$I(S_{mj} \subset S_{ni}) = \sum_{i:S_{mj} \subset S_{ni}} [NI(S_{ni})]^2, \tag{11}$$

$$i \in V(L_n); \quad j \in V(L_m); \quad m \leq n; \quad n \in [0, d(G)+1]; \quad m \in [0, d(G)]$$

and then computes the $NI(S_{mj} \subset S_{ni})$ values, cf. eq. (10).

Step 4 computes the final parameters by using a factor of ten to decrease the contribution of $I(S_{mj} \subset S_{ni})$ as n increases:

$$I(S_{mj}, \text{end } n) = \sum_{n=m}^{n} NI(S_{mj} \subset S_{ni}) \times 10^{m-n}. \tag{12}$$

The $NI(S_{mj}, \text{end } n)$ values are then calculated, cf. eq. (10). Since the contributions of line derivatives are weighted, this reminds one of the hierarchical criteria 1D–3D used in the IVEC procedure [9].

Step 5 selects the subgraphs S_{mr} (the real ones in G with m edges) among all S_{mj} subgraphs in $L_m(G)$ according to eq. (13):

$$NI(S_{mr}, \text{end } n) = \max NI(S_{mj}, \text{end } n), \qquad r \leq j \tag{13}$$

and orders them in a decreasing string of their centricities.

Complete information on the S_{mr} subgraphs in G is provided when $n = d(G) + 1$; however, the procedure can be stopped when no modifications appear in $ORD.NI(S_{mr}, \text{end } n)$ at two successive derivatives:

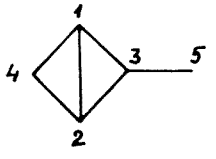
$$ORD.NI(S_{mr}, \text{end } k) = ORD.NI(S_{mr}, \text{end } k+1), \tag{14}$$

where *ORD* denotes the ordering given by the *NI* parameters. The *NI* values range in the [1.0000–0.0000] domain.

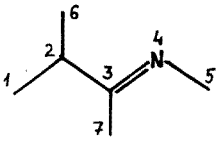
The summation in step 1 of the I_i values over all the vertices in $L_0, \ldots, L_n(G)$ provides global centricities, as will be shown below.

4. Examples for the MOLCEN algorithm implementation

Two different graphs were chosen for exemplifying the MOLCEN algorithm implementation: a polycyclic nonweighted graph G_1 and a weighted acyclic graph G_2.



G_1



G_2

Example 1 (for G_1):

Step 1 and 2:

	$NBCX(S_{0i})$		$NBCX(S_{1i})$		$NBCX(S_{2i})$
1	1.00000	1,2	1.00000	(1,2),(1,3)	1.00000
2	1.00000	1,3	0.98339	(1,2),(2,3)	1.00000
3	0.97638	2,3	0.98339	(1,3),(2,3)	0.99025
4	0.69055	1,4	0.76753	(1,2),(1,4)	0.84399
5	0.35748	2,4	0.76753	(1,2),(2,4)	0.84399
		3,5	0.53506	(1,3),(1,4)	0.83424
				(2,3),(2,4)	0.83424
				(1,3),(3,5)	0.68797
				(2,3),(3,5)	0.68797
				(1,4),(2,4)	0.67822

Step 3:

	$NBCX(S_{0j} \subset S_{0i})$	$NBCX(S_{0j} \subset S_{1i})$	$NBCX(S_{0j} \subset S_{2i})$
1	1.00000	1.00000	1.00000
2	1.00000	1.00000	1.00000
3	0.95331	0.86865	0.88145
4	0.47686	0.46093	0.54297
5	0.12779	0.11200	0.15687

	$NBCX(S_{1j} \subset S_{1i})$	$NBCX(S_{1j} \subset S_{2i})$
1,2	1.00000	1.00000
1,3	0.96707	0.91976
2,3	0.96707	0.91976
1,4	0.58910	0.54553
2,4	0.58910	0.54553
3,5	0.28628	0.27641

Step 4:

	$NBCX(S_{0j}$, end $n)$		
	end 0	end 1	end 2
1	1.00000	1.00000	1.00000
2	1.00000	1.00000	1.00000
3	0.95331	0.94562	0.94498
4	0.47686	0.47541	0.47608
5	0.12779	0.12636	0.12666

	$NBCX(S_{1j}$, end $n)$	
	end 1	end 2
1,2	1.00000	1.00000
1,3	0.96707	0.96276
2,3	0.96707	0.96276
1,4	0.58910	0.58514
2,4	0.58910	0.58514
3,5	0.28628	0.28539

Step 5:

$ORD.NBCX(S_{0j}$, end 0 – 2) = (1;2), (3), (4), (5)

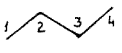
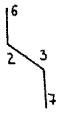
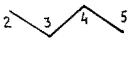
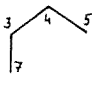
$ORD.NBCX(S_{1j}$, end 1 – 2) = (1,2), (1,3;2,3), (1,4;2,4),(3,5)

Here, we limited to $L_2(G_1)$ and S_{1j} as the maximal subgraph investigated.

The second example presents only step 5, namely the selection and ordering of real subgraphs S_{3r} (as $NI(S_{3r}$, end 4), for the weighted graph G_2.

Example 2 (for G_2):

Step 5:

ORD	Real subgraphs S_{3r}		*NBC*	*NBCX*
1	2,3,4,7		1.00000	1.00000
2	1,2,3,6		0.46659	0.64922
3	1,2,3,4		0.35503	0.35756
	2,3,4,6		0.35503	0.35756
4	1,2,3,7		0.33065	0.25100
	2,3,6,7		0.33065	0.25100
5	2,3,4,5		0.29208	0.19174
6	3,4,5,7		0.11609	0.08454

5. Results and discussion

The vertex ordering given by the MOLCEN algorithm was found to be identical to that given by the IVEC approach in a set of ten polycyclic graphs [9] G_3–G_{12} (table 1), with the exception of the first one. With the same exception (G_3), the two procedures correctly recognize the central edge. However, the global ordering of edges (subgraphs of type S_1) is quite different.

Table 1

IVEC and MOLCEN (*BC/BCX*; end 2) ordering of vertices and edges in G_3–G_{12}.

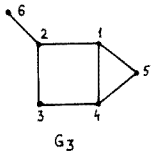
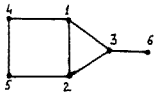
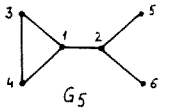
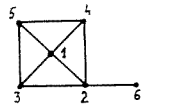
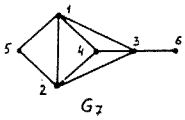
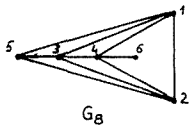
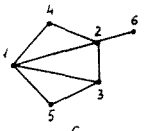
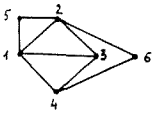
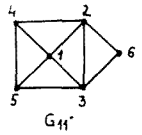
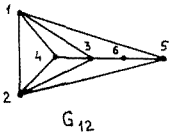
		Vertices	Global index
	IVEC	(1), (2), (3), (4), (5), (6)	
	BC	(1), (2), (3), (4), (5), (6)	1.9841
	BCX	(1), (4), (2), (3), (5), (6)	17.8770
G_3		Edges	
	IVEC	(12), (14), (23), (15), (34), (45), (26)	
	BC	(14), (12), (15), (34), (23), (45), (26)	2.3015
	BCX	(14), (12), (15), (34), (45), (23), (26)	29.7550
		Vertices	
	IVEC	(1;2), (3), (4;5), (6)	
	BC	(1;2), (3), (4;5), (6)	2.0029
	BCX	(1;2), (3), (4;5), (6)	17.8860
G_4		Edges	
	IVEC	(12), (13;23), (14;25), (36), (45)	
	BC	(12), (13;23), (14;25), (36), (45)	2.3342
	BCX	(12), (13;23), (14;25), (36), (45)	29.9440
		Vertices	
	IVEC	(1), (2), (3;4), (5;6)	
	BC	(1), (2), (3;4), (5;6)	1.8779
	BCX	(1), (2), (3;4), (5;6)	15.0120
G_5		Edges	
	IVEC	(12), (13;14), (25;26), (34)	
	BC	(12), (13;14), (25;26), (34)	1.9614
	BCX	(12), (13;14), (25;26), (34)	20.8680
		Vertices	
	IVEC	(1), (2), (3;4), (5), (6)	
	BC	(1), (2), (3;4), (5), (6)	2.1276
	BCX	(1), (2), (3;4), (5), (6)	24.2640
G_6		Edges	
	IVEC	(12), (13;14), (15), (23;24), (35;45), (26)	
	BC	(12), (13;14), (15), (23;24), (35;45), (26)	2.9357
	BCX	(12), (13;14), (15), (23;24), (35;45), (26)	63.5400
		Vertices	
	IVEC	(1;2), (3), (4), (5), (6)	
	BC	(1;2), (3), (4), (5), (6)	2.1995
	BCX	(1;2), (3), (4), (5), (6)	24,4530
G_7		Edges	
	IVEC	(12), (13;23), (14;24), (34), (15;25), (36)	
	BC	(12), (13;23), (14;24), (34), (15;25), (36)	2.9439
	BCX	(12), (13;23), (14;24), (34), (15;25), (36)	67.6800

Table 1 (continued)

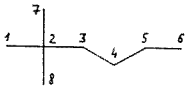
 G_8		Vertices	
	IVEC	(1;2;3), (4), (5), (6)	
	BC	(1;2;3), (4), (5), (6)	2.2088
	BCX	(1;2;3), (4), (5), (6)	27.6240
		Edges	
	IVEC	(12;13;23), (14;24;34), (15;25;35), (46)	
	BC	(12;13;23), (14;24;34), (15;25;35), (46)	3.1393
	BCX	(12;13;23), (14;24;34), (15;25;35), (46)	85.8600
 G_9		Vertices	
	IVEC	(1), (2), (3), (4), (5), (6)	
	BC	(1), (2), (3), (4), (5), (6)	2.1592
	BCX	(1), (2), (3), (4), (5), (6)	21.2730
		Edges	
	IVEC	(12), (13), (23), (15), (14), (24), (35), (26)	
	BC	(12), (13), (23), (14), (24), (15), (26), (35)	2.7449
	BCX	(12), (13), (23), (14), (15), (24), (26), (35)	50.0600
 G_{10}		Vertices	
	IVEC	(1), (2), (3), (4), (5), (6)	
	BC	(1), (2), (3), (4), (5), (6)	2.2939
	BCX	(1), (2), (3), (4), (5), (6)	24.1200
		Edges	
	IVEC	(12), (13), (14), (23), (15), (25), (34), (26), (46)	
	BC	(12), (13), (23), (14), (15), (25), (34), (26), (46)	2.9467
	BCX	(12), (13), (23), (14), (15), (25), (26), (34), (46)	59.7600
 G_{11}		Vertices	
	IVEC	(1), (2;3), (4;5), (6)	
	BCX	(1), (2;3), (4;5), (6)	2.2913
	BCX	(1), (2;3), (4;5), (6)	27.3000
		Edges	
	IVEC	(12;13), (23), (14;15), (24;35), (26;36), (45)	
	BC	(12;13), (23), (14;15), (24;35), (26;36), (45)	3.1439
	BCX	(12;13), (23), (14;15), (24;35), (26;36), (45)	77.5400
 G_{12}		Vertices	
	IVEC	(1;2), (3), (4), (5), (6)	
	BC	(1;2), (3), (4), (5), (6)	2.3048
	BCX	(1;2), (3), (4), (5), (6)	27.3000
		Edges	
	IVEC	(12), (13;23), (14;24), (15;25), (34), (36), (56)	
	BC	(12), (13;23), (14;24), (34), (15;25), (36), (56)	3.1531
	BCX	(12), (13;23), (14;24), (34), (15;25), (36), (56)	77.7200

Global centricity indices (for S_0 and S_1 fragments) are also included in table 1. The global values increase with graph centricity and centrocomplexity, respectively. A more detailed analysis of such global parameters will be made in future work.

In table 2, the ordering of S_0 and S_1 subgraphs in G_{13} is listed. One can see the different manner of ordering induced by the centricity *BC* operator versus the complexity *BCX* operator. However, since the line derivatives tend to focalize the graph towards the point of highest degree as n increases, the ordering given by the *BC* operator approaches that of the *BCX* operator. Thus, the ordering by *BC* will be reliable just for subgraphs below three edges.

Table 2

Centricities in 2,2-dimethylhexane (G_{13}).



G_{13}

S_0	$NBC(S_0$, end 2)	S_0	$NBCX(S_0$, end 2)
3	1.00000	2	1.00000
4	0.83085	3	0.35634
2	0.65856	4	0.28170
5	0.44023	5	0.24653
1	0.34160	1	0.11511
7	0.34160	7	0.11511
8	0.34160	8	0.11511
6	0.25250	6	0.06763
S_1	$NBC(S_1$, end 2)	S_1	$NBCX(S_1$, end 2)
3,4	1.00000	2,3	1.00000
2,3	0.72702	1,2	0.61859
4,5	0.40441	2,7	0.61859
1,2	0.38676	2,8	0.61859
2,7	0.38676	3,4	0.27214
2,8	0.38676	4,5	0.19538
5,6	0.20115	5,6	0.05276

In the following, we present the ordering of S_3 fragments in a cube (G_{14}, table 3), in terms of $NI(S_{3r}$, end 3). Note that the cube is a S_2 transitive graph.

Table 3

Centricities of S_3 fragments in a cube.

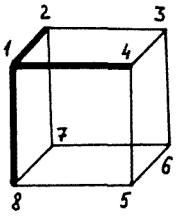
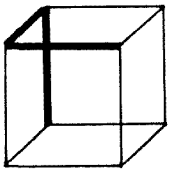
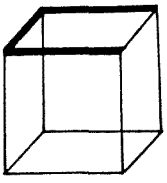
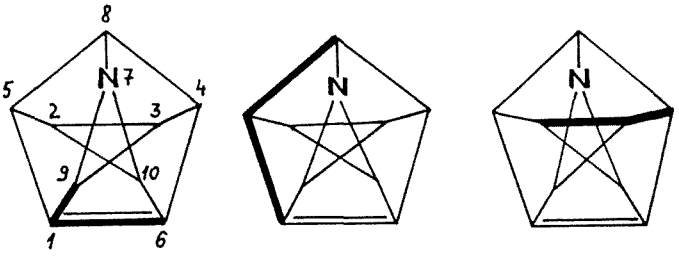
G_{14}			
$BC(S_{3r}$, end 3):	1.00000	0.41034	0.31845
$BC(S_{3r}$, end 3):	1.00000	0.34446	0.34100

Table 4

Centricities in Petersen's graph.

G_{15}			
S_2 :	(16–19) ,	(15–58) ,	(23–34)
$NBCX(S_{2j}$, end 3) :	1.0000000 ,	0.8538732 ,	0.5527499
S_0 :	(1,6) , (7) , (9,10) , (4,5) , (8) , (2,3)		
$NBCX(S_{0j}$, end 3) :	1.0000000 , 0.6724197 , 0.6291951 , 0.6282629 , 0.6185022 , 0.6175821		
S_1 :	(16) , (19;610) , (15;46) , (79;710) , (78) , (210;39)		
$NBCX(S_{1j}$, end 3) :	1,0000000 , 0.8598616, 0.8592246 , 0.5944589, 0.5866737 , 0.5697770		
	(48;58) , (25;34) , (23)		
	0.5697693 , 0.5693560, 0.5617945		

The ordering of S_0–S_2 fragments in the weighted Petersen's graph G_{15} (table 4) in terms of $NBCX(S_{mj}$, end 3) shows the ability of our algorithm to find the graph orbits.

Note that the unweighted Petersen's graph is S_2 transitive and its diameter equals two.

The MOLCEN algorithm was implemented on a TURBO-PASCAL, TOPIND, available upon request. The molecular structures were input in a dictionary form and the matrices are manipulated by using the heap.

6. Conclusions

The MOLCEN algorithm, developed on the basis of the IVEC concept [9] and the iterative line derivatives of graphs, enables one to evaluate the centricity and centocomplexity of subgraphs of various size. It provides reliable centric ordering of subgraphs, which could be useful finding graph orbits. Global parameters of centricity could characterize the molecular graphs in QSAR/QSPR studies.

Acknowledgements

Many thanks are addressed to Professors A.T. Balaban, Polytechnic Institute, Bucharest and D. Bonchev, Higher School of Chemical Technology, Burgas, for encouragement and helpful discussions. We also thank Dr. A. Imre, Department of Chemistry, "Babes-Bolyai" University, Cluj, for computer assistance.

References

[1] F. Harary, *Graph Theory* (Addison–Wesley, Reading, MA, 1969).
[2] D. Bonchev, O. Mekenyan and A.T. Balaban, in: *Mathematics and Computational Concepts in Chemistry*, ed. N. Trinajstić (Ellis Horwood, Chichester, 1986), Ch. 5.
[3] D. Bonchev, D. Kamenski and O.N. Temkin, J. Comput. Chem. 3(1982)95.
[4] D. Bonchev, Pure Appl. Chem. 55(1983)221.
[5] A.T. Balaban, Theor. Chem. Acta 53(1979)355;
A.T. Balaban and I. Motoc, MATCH 5(1979)197.
[6] D. Bonchev, A.T. Balaban and O. Mekenyan, J. Chem. Inf. Comput. Sci. 20(1980)106.
[7] M.V. Diudea and B. Parv, MATCH 23(1988)65.
[8] R.C. Read and D.G. Corneil, J. Graph Theory 1(1977)339.
[9] D. Bonchev, O. Mekenyan and A.T. Balaban, J. Chem. Inf. Comput. Sci. 29(1989)91.
[10] M.V. Diudea, T. Cipaianu and I.E. Kacsó, Molecular Topology 7. New metric characteristics in alkanes (4-trees), Studa Univ. "Babes-Bolyai" 35(1990)0000.
[11] V.A. Skorobogatov and A.A. Dobrynin, MATCH 23(1988)105.
[12] D. Bonchev, A.T. Balaban and M. Randić, Int. J. Quant. Chem. 19(1981)61.
[13] M.V. Diudea, O.M. Minailiuc and A.T. Balaban, J. Comput. Chem. 12(1991)527.
[14] N. Trinajstić, *Chemical Graph Theory*, Vol. I (CRC Press, Boca Raton, 1983), p. 17.
[15] S.H. Bertz, in: *Discrete Applied Mathematics*, ed. J.W. Kennedy and L.V. Quintas (North-Holland, Amsterdam, 1988), p. 65.
[16] G. Rücker and C. Rücker, J. Chem. Inf. Comput. Sci. 30(1990)187.
[17] M.V. Diudea and I. Silaghi-Dumitrescu, Rev. Roum. Chim. 34(1989)1175.
[18] R.T. Sanderson, *Chemical Bonds and Bond Energy* (Academic Press, New York, 1971).