Iterative Procedure for the Generalized Graph Center in Polycyclic Graphs[1]

DANAIL BONCHEV* and OVANES MEKENYAN

Department of Physical Chemistry, Higher School of Chemical Technology, BU-8010 Burgas, Bulgaria

ALEXANDRU T. BALABAN

Department of Organic Chemistry, The Polytechnic, 76206 Bucharest, Romania

Received March 9, 1988

An improvement in the algorithm for assigning vertex(es) as graph center(s) is achieved on the basis of the distances between vertexes and between edges, as well as proceeding from the vertex–edge incidence. The iterative vertex and edge centricities (IVEC) algorithm produces assignments that are in better agreement with the intuitive notion of graph center than those produced by the previous procedures.

INTRODUCTION

The problem of graph center has been first examined by Jordan, Sylvester, and Cayley[2] at the end of the 19th century. Introduced initially for acyclic graphs, the central vertexes have been similarly specified for cyclic graphs.[3] Alternative approaches have been developed based on the so-called mass center[4,5] or centroid[3] of a graph, as well as on the ideas of median vertexes[6] and vertex centrality.[4,7] Yet the existing graph center definitions proved to be impractical in the case of polycyclic structures, yielding frequently an unrealistically high number of central points. To avoid such difficulties, the graph center definition should meet the requirements for the minimum number and topological equivalence of the central vertexes (the latter should belong to the same orbit of the graph's automorphism group). This definition should also agree as much as possible with intuition. Such a program has only recently been initiated,[8,9] and the present study may be regarded as the final stage of its development.

Indeed, the more precise graph center concept is not only a question of academic interest. Among the numerous possible practical applications,[10] the graph center concept seems to be of particular use in coding of chemical structures.[11] Dubois et al.[12] have developed DARC data analysis methodology that starts with the coding of atomic environment in a molecule around a fragment regarded as central. Centric codes have been proposed for acyclic graphs by Lederberg et al.[13] and Read,[14] aiming at the enumeration and nomenclature of acyclic compounds. Proceeding from the centric description of vertexes, edges, and rings in the so-called kinetic graphs (which represent reaction intermediates, elementary steps, and reaction routes, respectively), Bonchev et al.[15] developed a classification and coding of chemical reactions with a linear mechanism. The chemical nomenclature is another area of interest where a general concept of the graph center is of importance, as demonstrated for benzenoid hydrocarbons[16] and, later, in proposing the principles of a universal chemical nomenclature capable of naming all chemical compounds in a unified manner.[17] The proper selection of the center may be of interest in pattern recognition[18] providing an objective local clustering procedure.

The complete centric ordering of graph vertexes (and edges) into equivalence classes is a convenient approach to structure–property correlations. This can be done by constructing centric topological indices[8,19] showing a high correlation with such a property as the alkane octane numbers.[20] The centric ordering of the atoms in a molecule may find a direct use in correlations with atomic properties, as shown for ^{1}H NMR chemical shifts of some benzenoid hydrocarbons.[9]

The concept of graph center is also of certain use for theoretical considerations of graphs. Providing an ordered list of vertexes and edges into equivalence classes, it is of interest in problems of graph isomorphism,[21] subgraph isomorphism (fragment search),[22] generation of graphs,[23] etc. By replacing the rings in benzenoid systems by vertexes,[24] such an approach may help to specify the central ring, which could find some useful applications. The concept of a "shell of neighbors" moving away from the center is of interest for different purposes including isometric graphs,[25,26] the unsolved problem of a canonical pictorial representation of graphs,[27] etc.

BACKGROUND

In our previous papers[8,9a,b] the concept of graph center was generalized by using a hierarchical set of criteria based on the distances and paths in the graphs. The distance between two graph vertexes i and j, $d(ij)$, is assumed to be the number of edges connecting these vertexes along the shortest path between them. For any vertex i the maximal distance to any other vertex j is called *vertex eccentricity*,[3] $e(i) = \max d(ij)$, and the sum of distances to all other vertexes, $d(i) = \sum_j d(ij)$, is called distance of a vertex[25,26,28] or *vertex distance sum* (rank),[8] or distance number.[9a,29] Whenever a given distance l occurs many times, say K_l times, one may group together such distances and express them as a *vertex distance code*: K_1, K_2, K_3, ..., K_{max}, K_l being the frequency numbers of the different distances. Thus, $d(i) = \sum_l K_l d_l(i)$ with each $d_l(i)$ value shown only once.

The distance-based criteria for a vertex i to belong to the graph center used earlier[8] are as follows:

(1D) Minimum vertex eccentricity, $e(i) = \min$. This is the classical definition of a graph center.[3]

(2D) Minimum vertex distance sum, $d(i) = \min$.

(3D) Minimum number K of occurrence of the largest distance, $K_{max}(i) = \min$. If the largest distance occurs the same number of times for several vertexes, the next largest distance is considered and so on.

(4D) By deleting all but the graph vertexes qualified as centers according to the first three criteria, one obtains a smaller graph (kernel). By iterating criteria 1D–3D over the kernel, one could obtain an even smaller kernel, and so on, until two subsequent iterations fail to reduce further the number of center vertexes. The result is the graph *polycenter*.

Criteria 1D–4D are applied hierarchically; i.e., each next criterion provides possibilities for a further discrimination of the graph vertexes qualified as central by the previous criterion.

In a subsequent paper[9a] it was shown how one may reduce further the number of center vertexes by making use of all

* To whom correspondence should be addressed.

0095-2338/89/1629-0091$01.50/0

paths in the graph. The length of the path $p(ij)$ between vertexes i and j (expressed as the number of edges along the path) is called its *elongation*[4] el(ij). The maximal elongation of a graph vertex is called *vertex path eccentricity*[28] $e_\tau(i)$ = max el(ij). The *vertex path sum*[9a] $P_i = \sum_j \text{el}(ij)$ is also introduced as the total number of edges in all paths connecting vertexes i and j. Finally, the concept for a *vertex path code*,[9a] $K_1, K_2, K_3, ..., K_{max}$ is useful, K_l' being the frequency number of a certain elongation el(ij) = l. A set of path criteria, analogous to the distance criteria, was proposed on this basis to qualify the vertex i as a graph center:

(1P) Minimum vertex path eccentricity, $e_\tau(i)$ = min.

(2P) Minimum vertex path sum, $P(i)$ = min.

(3P) Minimum number K' of occurrence of the largest elongation, $K'_{max}(i)$ = min. Should the largest elongation be the same for several vertexes, one compares the next largest elongation, etc.

The central vertex(es) resulted from criteria 1P–3P is (are) called *oligocenter*.

In principle one may adopt similar criteria based on self-avoiding walks for further reduction of the number of vertexes in the oligocenter. However, listing all paths or self-avoiding walks in a graph is a tedious process that generally requires a computer.

In the present paper we report an approach different from that using paths or self-avoiding walks, thus obviating the need of complicated calculations for obtaining these graph invariants. Moreover, in the new approach some graphs are assigned centers that come more closely to the intuitive idea of central vertexes than the assignments corresponding to the previous procedures. Indeed, according to criteria 1P–3P, when they become necessary as shown in Table I of part II in this series,[9a] the graph center is often a vertex adjacent to an endpoint. We shall show in Table III of the present paper that the new procedure results in a more reasonable assignment of centers.

In addition, the present approach does not involve artificial reduction in the number of central vertexes as in criterion 4D. Indeed, this criterion leads to a loss of perception of the graph symmetry and renders difficult the assignment of vertexes to the orbits of the graph automorphism group. We shall show in Table IV of the present paper that in some cases the new algorithm specifies the central vertexes in a manner more satisfying than criterion 4D.

THE BASIC IDEA

We start from the assumption that one may define the graph center by using both vertexes and edges in an iterative procedure: *central are those vertexes that are incident to the most central edges; conversely, central are those edges that are incident to the most central vertexes.*

The new algorithm, which we call iterative vertex and edge centricities (IVEC), provides a sequential reordering of the graph vertexes and edges on the basis of their metric properties and incidence. A brief sketch of this procedure was published earlier in a review paper under the name DISTANCE algorithm.[11]

In the initial (zero) iteration, criteria 1D–3D are applied to the graph vertexes ordering them in equivalence classes. Ranks 1, 2, 3, ... are assigned to those classes starting from the most central ones, i.e., from those vertexes having the lowest $e(i)$, $d(i)$ (in case several vertexes have the same lowest eccentricity), and $K_{max}(i)$ (in case several vertexes have the same lowest eccentricity and distance sum) values. Such vertexes receive rank 1, etc.

Then the same three criteria are applied to the graph edges, arriving at an analogous ordering and rank assignment on the basis of the edge distance matrix. This matrix is derived from the edge adjacency matrix in a manner similar to the derivation of the vertex distance matrix from the vertex adjacency matrix. The edge adjacency matrix entries are one or zero according to whether two edges have, or do not have, a common endpoint. In the edge distance matrix, the zero edge adjacencies are replaced by edge distances larger than one (by definition the graph vertex and edge distances are always integers).

In the first iteration the sum of ranks is calculated for the edges that are incident to each vertex, and these sums provide additional discrimination *within the current equivalence classes* formed after the zero iteration. New ranks are assigned to the vertexes on this basis. Then the same operation is effected for the graph edges by taking into account the sums of the new ranks of their incident vertexes.

In cases where the same rank sum is obtained from different summands (e.g., 1 + 4 = 2 + 3) the partition containing a smaller vertex rank is awarded priority (i.e., that edge/vertex is considered to be more central that is incident to a more central vertex/edge). Examples illuminating this point will be given in connection with Tables II and III.

In the next iteration step the same centric reordering and reranking of graph vertexes and edges continues, and the process is terminated when the same vertex and edge equivalence classes are obtained in two consecutive iterations. Usually the iterations yield a complete partitioning of the vertexes and edges into equivalence classes that coincide with the orbits of the graph's automorphism group.

THE ITERATIVE VERTEX AND EDGE CENTRICITY (IVEC) ALGORITHM AND TWO EXAMPLES FOR ITS IMPLEMENTATION

The IVEC algorithm has the following six steps:

Step 1. Each vertex and each edge in the hydrogen-depleted graph are labeled in an arbitrary manner (in the present paper, edges are indicated by the labels of their endpoints in increasing order without comma).

Step 2. For each vertex and edge the first three metric criteria 1D–3D of the first part of this series[8] are applied on the basis of the vertex distance matrix and edge distance matrix, respectively.

Step 3. The vertex and edge equivalence classes, specified and centrically ordered according to step 2, receive integer ranks starting with the rank of the most central class (0 ranking).

Step 4. (a) For each vertex i, the 0-edge-ranks (^0ER_i) of its incident edges are listed in increasing order: $^0ER_i(1)$, $^0ER_i(2)$, ..., and the sum ^0SE_i is calculated

$$^0SE_i = \sum_r {}^0ER_i(r)$$

New vertex ranks (^IVR_i) are obtained according to these sums.

(b) In cases of equal rank sums (^0SE_i) originating in different summands, a lower rank (i.e., more centric location) is given to the vertex whose edge-rank sum has a smaller summand or a larger number of smaller summands. In assigning ranks the previously established hierarchy is conserved so as to avoid oscillations in ranking.

Step 5. For each edge i, the I vertex ranks (^IVR_i) of its incident vertexes (endpoints) are listed in increasing order $^IVR_i(1)$, $^IVR_i(2)$, and the sum ^ISV_i is calculated

$$^ISV_i = \sum_t VR_i(t)$$

These sums (and their partitioning for equal sums) lead to new edge ranks (^IER_i).

Step 6. Steps 4 and 5 are repeated iteratively until the ranking of vertexes and edges undergoes no further modification:

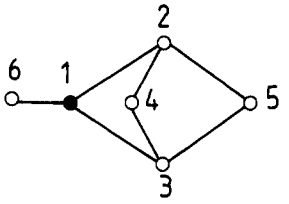
$$^{K+1}VR_i = {}^KVR_i \qquad {}^{K+1}ER_i = {}^KER_i$$

Table I. Example of the Application of the IVEC Algorithm to Graph 17 in Figure 1 and Table III[a]

steps 1–3						steps 4 and 5						
V_i	vertex distance code	${}^0VR_i$	E_i^l	edge distance code	${}^0ER_i$	V_i	edge incidence	$\sum {}^0ER_i = {}^0SE_i$	${}^IVR_i$	E_i^b	$\sum {}^IVR_i = {}^ISV_i$	${}^IER_i$
1	$1^4 2^1$	1	12	$1^6 2^3$	1	1	12, 13, 14, 15	1 + 1 + 1 + 2 = 5	1	12	1 + 1 = 2	1
2	$1^4 2^1$	1	13	$1^6 2^3$	1	2	12, 23, 24, 25	1 + 1 + 1 + 2 = 5	1	13	1 + 1 = 2	1
3	$1^4 2^1$	1	23	$1^6 2^3$	1	3	13, 23, 34, 35	1 + 1 + 1 + 2 = 5	1	23	1 + 1 = 2	1
4	$1^4 2^1$	1	14	$1^6 2^3$	1	4	14, 24, 34, 46	1 + 1 + 1 + 3 = 6	2	14	1 + 2 = 3	2
5	$1^3 2^1 3^1$	2	24	$1^6 2^3$	1	5	15, 25, 35	*c*	3	24	1 + 2 = 3	2
6	$1^1 2^3 3^1$	2	34	$1^6 2^3$	1	6	46	*c*	4	34	1 + 2 = 3	2
			15	$1^5 2^4$	2					15	1 + 3 = 4	3
			25	$1^5 2^4$	2					25	1 + 3 = 4	3
			35	$1^5 2^4$	2					35	1 + 3 = 4	3
			46	$1^3 2^6$	3					46	*c*	4

[a] The next iteration (not shown) repeats the numbering. [b] The edge notation implies the vertex incidence of each edge by its endpoints. [c] No sums are needed because vertexes 5 and 6 (as well as edge 46 on the last line) are each in separate classes; on dividing the previous vertex ranking 1 into two classes, vertexes 5 and 6 automatically are moved into higher ranks (by one class).

Table II. Second Example Needing Additional Discrimination of Sums (Boldface) for Graph 54 in Table IV[a]



steps 1–3						steps 4 and 5						
V_i	vertex distance code	${}^0VR_i$	E_i	edge distance code	${}^0ER_i$	V_i	edge incidence	$\sum {}^0ER_i = {}^0SE_i$	${}^IVR_i$	E_i	$\sum {}^IVR_i = {}^ISV_i$	${}^IER_i$
1	$1^3 2^2$	1	12	$1^4 2^2$	1	1	12, 13, 16	**1 + 1 + 3 = 5**	1	12	1 + 2 = 3	1
2	$1^3 2^2$	1	13	$1^4 2^2$	1	2	12, 24, 25	**1 + 2 + 2 = 5**	2	13	1 + 2 = 3	1
3	$1^3 2^2$	1	24	$1^3 2^3$	2	3	13, 34, 35	**1 + 2 + 2 = 5**	2	24	2 + 3 = 5	2
4	$1^2 2^2 3^1$	2	34	$1^3 2^3$	2	4	24, 34	2 + 2 = 4	3	34	2 + 3 = 5	2
5	$1^2 2^2 3^1$	2	25	$1^3 2^3$	2	5	25, 35	2 + 2 = 4	3	25	2 + 3 = 5	2
6	$1^1 2^2 3^2$	3	35	$1^3 2^3$	2	6	16		4	35	2 + 3 = 5	2
			16	$1^2 2^4$	3					16		3

[a] The next iteration (not shown) reproduces the same vertex and edge ranking and centric ordering.

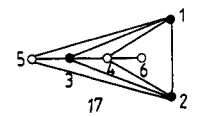
The resulting ranking has the vertex(es) with rank 1 as the graph center.

Two examples are presented in Tables I and II and Figure 1. The distance code indicates the distances under the form $d_l^{K_l}$ where the right superscript is the number of times the distance d_l occurs from vertex/edge i to all other vertexes/edges.

As shown in Table II the central vertexes 1, 2, and 3 have the same sum of the incident edge ranks: ${}^0SE_1 = {}^0SE_2 = {}^0SE_3 = 5$ (boldface). However, vertex 1 is awarded priority (rank 1) because it is incident to the two most central edges 12 and 13 (having rank ${}^0ER_{12} = {}^0ER_{13} = 1$), while vertexes 2 and 3 are incident to only one of these edges.

RESULTS AND DISCUSSION

Table III lists some of the polycyclic graphs with six or less vertexes labeled arbitrarily (no. 11–21); they were presented in Table I of part II in this series.[9a] These graphs were not amenable to having their centers located by criteria 1D–4D, and therefore path criteria were also used. On the left-hand side the center vertexes obtained earlier[9a] are shown as black points, and on the right-hand side the center vertexes according to the new IVEC algorithm are presented similarly. In the same table one may see the centric ordering or ranking of vertexes and edges: the first brackets for rank 1, the next for rank 2, and so on. (All vertex numberings are according to the IVEC ranking.) It can be seen that in all cases where the previous procedure had resulted in a graph center adjacent to an endpoint (graphs 11–13 and 15–18), the IVEC algorithm assigns the graph center to a vertex that is not adjacent to an endpoint. For graph 17, which was discussed in detail as an example in Figure 1 and Table I, the IVEC center comprises three vertexes, all of them differing from the single central vertex previously found.[9a] For graphs 19–21 devoid of terminal vertexes, the IVEC centers also differ from path-based centers[9a] which are connected with more (one for graphs 20 and 21 and two for graph 19) vertexes of degree two. Only for graph 14 is there coincidence between the new and the earlier assignment.



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

D

	12	13	23	14	24	34	15	25	35	46
12	0	1	1	1	1	2	1	1	2	2
13	1	0	1	1	2	1	1	2	1	2
23	1	1	0	2	1	1	2	1	1	2
14	1	1	2	0	1	1	1	2	2	1
24	1	2	1	1	0	1	2	1	2	1
34	2	1	1	1	1	0	2	2	1	1
15	1	1	2	1	2	2	0	1	1	2
25	1	2	1	2	1	2	1	0	1	2
35	2	1	1	2	2	1	1	1	0	2
46	2	2	2	1	1	1	2	2	2	0

ED

Figure 1. Vertex distance matrix D and edge distance matrix ED of graph 17. The vertex numbering is according to the IVEC ranking.

In finding the IVEC central vertexes of graphs 11–21 there is no need to use the priority condition given in step 4b of our algorithm. A priority analogous to the one provided by step 5 was used in several cases of equally ranked noncentral edges. These are 36 < 45 for graph 13; 25, 26 < 34 for graph 14;

Table III. Graphs from Table 1 of Ref 9a with Their Centers According to Criteria 1D–4D and 1P–4P and the Centers and Centric Ordering of Vertexes and Edges According to the IVEC Algorithm

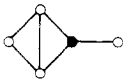
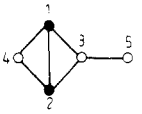
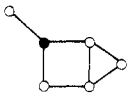
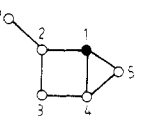
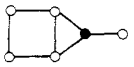
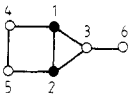
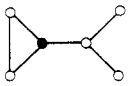
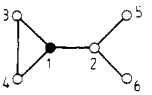
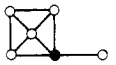
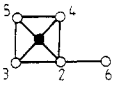
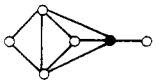
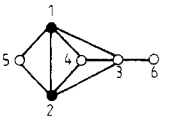
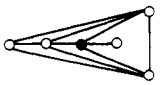
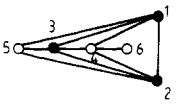
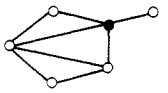
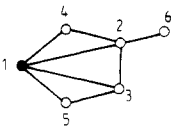
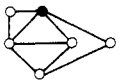
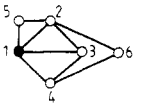
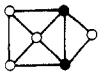
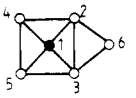
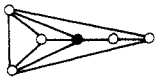
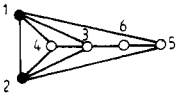
no.	center[9a]	IVEC center	centric ordering of vertexes	centric ordering of edges
11			(1,2) (3) (4) (5)	(12) (13,23) (14,24) (35)
12			(1) (2) (3) (4) (5) (6)	(12) (14) (23) (15) (34) (45) (26)
13			(1,2) (3) (4,5) (6)	(12) (13,23) (14,25) (36) (45)
14			(1) (2) (3,4) (5,6)	(12) (13,14) (25,26) (34)
15			(1) (2) (3,4) (5) (6)	(12) (13,14) (15) (23,24) (35,45) (26)
16			(1,2) (3) (4) (5) (6)	(12) (13,23) (14,24) (34) (15,25) (36)
17			(1,2,3) (4) (5) (6)	(12,13,23) (14,24,34) (15,25,35) (46)
18			(1) (2) (3) (4) (5) (6)	(12) (13) (23) (15) (14) (24) (35) (26)
19			(1) (2) (3) (4) (5) (6)	(12) (13) (14) (23) (35) (25) (34) (26) (46)
20			(1) (2,3) (4,5) (6)	(12,13) (23) (14,15) (24,35) (26,36) (45)
21			(1,2) (3) (4) (5) (6)	(12) (13,23) (14,24) (15,25) (34) (36) (56)

Table IV. Some Graphs from Table I of Ref 8 with Their Centers According to Criteria 1D–4D and the Centric Ordering (Ranking) of Vertexes and Edges According to the IVEC Algorithm

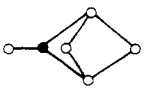
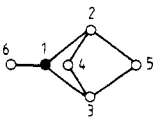
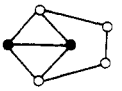
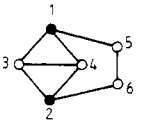
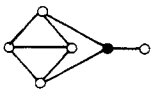
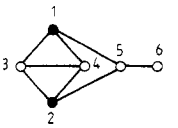
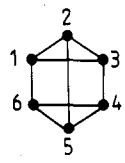
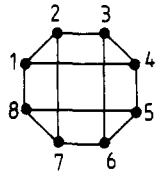
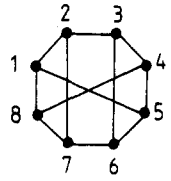
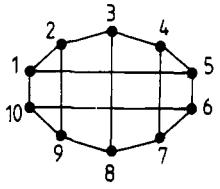
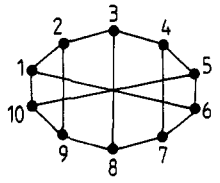
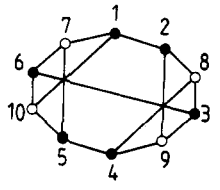
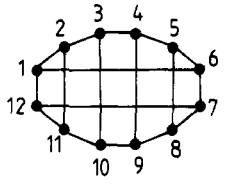
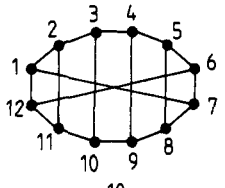
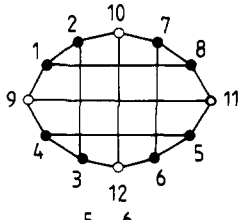
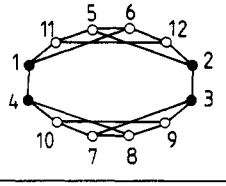
no.	center[8]	IVEC center	centric ordering (ranking) of vertexes	centric ordering (ranking) of edges
54			(1) (2,3) (4,5) (6)	(12,13) (24,25,34,35) (16)
66			(1,2) (3,4) (5,6)	(13,14,23,24) (15,26) (34) (56)
72			(1,2) (5) (3,4) (6)	(15,25) (13,14,23,24) (34) (56)

Table V. Some Symmetrical Cubic Graphs with Their Centers, Vertex Centric Ordering, and Edge Centric Ordering As Determined by the IVEC Procedure

no.	IVEC center	centric ordering (ranking) of vertexes	centric ordering (ranking) of edges
1		(1–6)	(all edges)[a]
2		(1–8)	(all edges)
3		(1–8)	(12,23,34,56,67,78,15,48) (18,27,36,45)
4		(1–10)	(12,23,34,45,67,78,89,910,15,610) (110,29,38,47,56)
5		(1–10)	(12,23,34,45,67,78,89,910,16,510) (110,29,38,47,56)
6		(1–6) (7–10)	(12,36,45) (all other edges)
7		(1–12)	(12,23,34,45,56,78,89,910,1011,1112,16,712) (112,211,310,49,58,67)
8		(1–12)	(12,23,34,45,56,78,89,910,1011,1112,17,612) (112,211,310,49,58,67)
9		(1–8) (9–12)	(18,45,23,67) (12,34,56,78) (210,710,511,811,312,612,19,49) (311,1012)
10		(1–4) (5–12)	(14,23) (16,25,37,48,111,212,39,410) (56,78,511,612,89,710,910,1112)

[a] See text.

15 < 23, 24 for graph 15; 14 < 23 and 25 < 34 for graph 19; 26, 36 < 45 for graph 20; and 15, 25 < 34 for graph 21. In three of these cases (graphs 15, 19 (case a), and 21) priority is awarded to the edge incident to a central vertex. In the remaining four cases a smaller rank is assigned to the edge incident to a vertex of higher degree; i.e., the priority rule in these cases may be reformulated in terms of vertex degrees: 1,3 < 2,2 for graphs 13 and 14, 2,4 < 3,3 for graphs 19 (case b) and 20. Clearly, the centric ranking of graph edges in these cases coincides with that based on molecular connectivity. In general, however, the centric ranking (or ordering) of graph vertexes and edges differs from their connectivity or extended connectivity ranking, and their interrelation needs additional analysis.

In Table IV are presented three graphs with their numbers from Table I of the first part in this series.[8] The centers, found by applying criteria 1D–4D, are indicated on the left-hand side as black points. On the right-hand side the IVEC centers are shown similarly. Vertexes are numbered according to the IVEC ranking, and this ranking of vertexes and edges is indicated as in Table III. For most of the 107 graphs included in Table I of the first part in this series,[8] criteria 1D–3D suffice and, indeed, the centers are identical with the IVEC centers. In eight graphs from the cited Table I the 4D criterion produces graph centers (polycenters) different from those specified by criteria 1D–3D. In six of them having original numbers 25, 54, 64, 102, 105, and 106 the polycenter coincides with the IVEC center. An example is graph 54 of Table IV, where the graph center is a vertex adjacent to an endpoint. The reason the IVEC algorithm gives this result can be traced to the additional rank discrimination introduced by comparing the summands within the partition as shown in Table II. However, for the other two graphs of Table IV the centers do not coincide in the two procedures. Thus, for graph 72 the IVEC center is no longer a vertex adjacent to an endpoint.

A final series of examples consists of symmetric planar and nonplanar regular trivalent graphs (cubic graphs), which are shown in Table V. The IVEC algorithm finds correctly all vertex and edge orbits of the graph's automorphism group except the trigonal prism (graph 1). In this case the IVEC procedure assigns the same rank 1 to all edges, although they belong to two orbits. This is so far the only instance when the new algorithm fails to discriminate edge orbits.

In a forthcoming paper[30] the problem of refining the IVEC algorithm for obtaining vertex and edge orbits will be discussed in detail. It can be shown that one may differentiate edges in the trigonal prism by applying the same IVEC procedure to the so-called line (or first derivative) graph in which each vertex stands for an edge in the initial graph while each edge corresponds to a three-vertex fragment ("a connection"), respectively. The iterative vertex–edge centric reordering from the initial graph is thus replaced by an iterative edge–connection centric reordering.

For the remaining cubic graphs 2–10 the IVEC procedure produces the vertex and edge partitioning into centrically ordered orbits. The cube (graph 2) is edge and vertex transitive, but its Moebius counterpart (graph 3) is vertex transitive and edge intransitive. This is also true for other prisms and Moebius counterparts: the pentagonal and hexagonal prism (graphs 4 and 7, respectively) and their Moebius counterparts (graphs 5 and 8, respectively) are all vertex transitive and edge intransitive. Graphs 6, 9, and 10 from Table V are vertex and edge intransitive.

CONCLUSION

We have presented a new procedure for assigning graph centers in polycyclic graphs starting from the basic idea to consider jointly the centric location of vertexes and edges. The new iterative concept of the graph center thus combines the graph metric with vertex–edge incidence. As shown in the previous sections the iterative vertex and edge centricity (IVEC) algorithm is advantageous as compared with the previous procedures. In many cases it produces reasonable center assignments in better agreement with intuition, e.g., by eliminating those previous results that specify the center of complicated polycyclic graphs to be located in a vertex adjacent to a terminal vertex. No portions of the graphs are eliminated during the procedure that preserves graph symmetry and usually provides a partitioning of vertexes and edges into orbits. The IVEC algorithm may thus be useful for the graph isomorphism problem as we shall discuss in a future paper.[30] One may conclude that the new graph center concept meets the requirements for a better agreement with intuition, as well as for the minimum number of central vertexes that when being more than one belong to the same orbit of the graph's automorphism group. The appreciable number of tested graphs also indicated that the first iteration step is almost always sufficient, and this qualifies the IVEC algorithm as a very fast one. The IVEC concept thus appears as a scheme that is not only general (i.e., applicable to any connected graph) and systematic (i.e., derivable in an orderly and hierarchical manner) but also practical (i.e., producing results without complicated computation).

REFERENCES AND NOTES

(1) This paper is part III in the series "Generalization of the Graph Center Concept". Parts I and II are ref 8 and 9a, respectively.
(2) König, D. *Theorie der Endlichen und Unendlichen Graphen*; reprinted, Chelsea: New York, 1950; p 64.
(3) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1969; p 151.
(4) Ore, O. *Theory of Graphs*; American Mathematical Society: Providence, RI, 1962.
(5) Polansky, O. E.; Bonchev, D. The Minimum Distance Number of Trees. *MATCH* **1987**, *21*, 314–344.
(6) Halberstam, F. Y.; Quintas, L. V. *Distance and Path Degree Sequences for Cubic Graphs*; Pace University: New York, 1982. *A Note on Table of Distance and Path Degree Sequences for Cubic Graphs*; Pace University: New York, 1982.
(7) Zelinka, B. Medians and Peripherians of Trees. *Arch. Math.* **1968**, *4*, 87–95. *Generalized Centers of Gravity of Trees*; University of Karlova: Prague, 1970; pp 127–136.
(8) Bonchev, D.; Balaban, A. T.; Mekenyan, O. Generalization of the Graph Center Concept and Derived Topological Centric Indexes. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 106–113.
(9) (a) Bonchev, D.; Balaban, A. T.; Randić, M. The Graph Center Concept for Polycyclic Graphs. *Int. J. Quantum Chem.* **1981**, *19*, 61–82. (b) Bonchev, D.; Balaban, A. T.; Randić, M. The Graph Center Concept for Polycyclic Graphs. *Int. J. Quantum Chem.* **1982**, *22*, 441.
(10) Bonchev, D. The Concept for the Centre of a Chemical Structure and Its Applications. *J. Mol. Struct.* (in press).
(11) Bonchev, D.; Mekenyan, O.; Balaban, A. T. Algorithms for Coding Chemical Compounds. In *Mathematics and Computational Concepts in Chemistry*; Trinajstić, N., Ed.; Ellis Horwood: Chichester, U.K., 1986; Chapter 5.
(12) Dubois, J. E.; Chretien, J. Data Analysis Methodology. DARC (Description, Acquisition, Retrieval, Correlation) Topological System. *J. Chromatogr. Sci.* **1974**, *12*, 811–821. Numerous publications of J. E. Dubois; see *Chemical Applications of Graph Theory*; Balaban, A. T., Ed.; Academic Press: London, 1976; p 333.
(13) Lederberg, J.; Sutherland, G. L.; Buchanan, B. G.; Feigenbaum, E. G.; Robertson, A. V.; Duffield, A. M.; Djerassi, C. Applications of Artificial Intelligence for Chemical Inference. 1. The Number of Possible Organic Compounds. Acyclic Structures Containing C, H, O, and N. *J. Am. Chem. Soc.* **1969**, *91*, 2973–2976. Masinter, L.; Sridharan, N. S.; Lederberg, J.; Smith, D. H. Applications of Artificial Intelligence for Chemical Inference. 11. Exhaustive Generation of Cyclic and Acyclic Isomers. *J. Am. Chem. Soc.* **1974**, *96*, 7702–7714.
(14) Read, R. C. A New System for the Designation of Chemical Compounds. 1. Theoretical Preliminaries and the Coding of Acyclic Compounds. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 135–149.
(15) Bonchev, D.; Kamenski, D.; Temkin, O. N. Graph Theoretical Classification and Coding of Chemical Reactions with a Linear Mechanism. *J. Comput. Chem.* **1982**, *3*, 95–111.
(16) Bonchev, D.; Balaban, A. T. Topological Centric Coding and Nomenclature of Polycyclic Compounds. 1. Condensed Benzenoid Systems (Polyhexes, Fusenes). *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 223–229.
(17) Bonchev, D. Principles of a Novel Nomenclature of Organic Compounds. *Pure Appl. Chem.* **1983**, *55*, 221–228.

(18) Ritter, G. L.; Isenhour, T. L. Minimal Spanning Tree Clustering of Gas Chromatographic Liquid Phases. *Comput. Chem.* **1977**, *1*, 145-153. Everitt, B. *Cluster Analysis*; Halsted: New York, 1974.
(19) Balaban, A. T. Chemical Graphs. XXXIV. Five New Topological Indices for the Branching of Tree-Like Graphs. *Theor. Chim. Acta* **1979**, *53*, 355-375.
(20) Balaban, A. T.; Motoc, I. Chemical Graphs. XXXVI. Correlations between Octane Number and Topological Indices of Alkanes. *MATCH* **1979**, *5*, 197-218.
(21) Read, R. C.; Corneil, D. G. The Graph Isomorphism Disease. *J. Graph Theor.* **1977**, *1*, 339-363.
(22) Stobaugh, R. E. Chemical Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 271-275.
(23) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. *Application of Artificial Intelligence for Organic Chemistry*; McGraw-Hill: New York, 1980.
(24) Balaban, A. T.; Harary, F. Chemical Graphs. 4. Enumeration and Proposed Nomenclature of Benzenoid Catacondensed Polycyclic Aromatic Hydrocarbons. *Tetrahedron* **1968**, *24*, 2505-2516. Polansky, O. E.; Rouvray, D. H. Graph-Theoretical Treatment of Aromatic Hydrocarbons. I. The Formal Graph-Theoretical Description. *MATCH* **1976**, *2*, 63-90.
(25) Entriger, R. C.; Jackson, D. E.; Snyder, D. A. Distance in Graphs. *Czech. Math. J.* **1976**, *26*, 283-296.
(26) Skorobogatov, V. A.; Khvorostov, P. V. Analiz Metricheskikh svoistv grafov. *Vychisl. Sist.* **1981**, *91*, 1-20.
(27) Balaban, A. T.; Mekenyan, O.; Bonchev, D. Unique Description of Chemical Structures Based on Hierarchical Ordered Extended Connectivities (HOC Procedures). I. Algorithms for Finding Graph Orbits and Canonical Numbering of Atoms. *J. Comput. Chem.* **1985**, *6*, 538-551.
(28) Skorobogatov, V. A.; Dobrynin, A. A. Metric Analysis of Graphs. *MATCH* **1988**, *23*, 105-151.
(29) Polansky, O. E.; Bonchev, D. The Wiener Number of Graphs. I. General Theory and Changes Due to Graph Operations. *MATCH* **1987**, *21*, 133-186.
(30) Bonchev, D.; Mekenyan, O.; Karabunarliev, S. The IVEC Algorithm for Coding of Chemical Compounds and Centric Ordering of Their Atoms and Bonds. Unpublished data.

SMILES. 2. Algorithm for Generation of Unique SMILES Notation

DAVID WEININGER, ARTHUR WEININGER, and JOSEPH L. WEININGER*
Daylight Chemical Information Systems, Irvine, California 92714

Received May 4, 1988

The chemical notation language SMILES is designed for the conversion of an arbitrarily chosen description of a chemical structure to one unique notation. This is accomplished in a two-stage algorithm, CANGEN. The first stage involves CANonicalization of structure, whereby the molecule is treated as a graph with nodes (atoms) and edges (bonds). Each atom is canonically ordered and labeled. In the second stage, starting with the lowest labeled atom, a molecular graph is GENerated, which is the unique SMILES structure.

INTRODUCTION

The SMILES chemical notation language was introduced in the first paper of this series.[1] Processing chemical information with greater efficiency than conventional methods, it represents a new approach to computerized chemical nomenclature. SMILES is simple to write because rules and hierarchical procedures, which are inherently difficult for the chemist, are relegated to computer algorithms. For a given chemical structure, arbitrary SMILES notation can take many equally valid forms. One must emerge as "unique" to serve as the identifier of the structure for database and other computer applications.

This is accomplished by a method called CANGEN that combines two separate algorithms, CANON and GENES. The first stage, CANON, labels a molecular structure with canonical labels. The structure is treated as a graph with nodes (atoms) and edges (bonds). Each atom is given a numerical label on the basis of its topology. In the second stage, GENES generates the unique SMILES notation as a tree representation of the molecular graph. GENES selects the starting atom and makes branching decisions by referring to the canonical labels as needed.

The combined procedure designates a unique SMILES notation for each chemical structure regardless of the many possible equivalent descriptions of the structure that might be input.

THEORETICAL BACKGROUND

Generally, graph theory has become important in applications to chemical information because it provides the basis for codification of nomenclature in chemical computer programs.[2] The classification and ordering of nodes in a graph is here applied to chemical structure notation. With an initial set of node properties and a given connectivity for a two-dimensional, nondirected graph (with N nodes and E edges), each node is assigned a rank. In CANGEN this ranking completely discriminates each node environment with respect to all initial mode properties. Aside from node and edge properties, the classification algorithm must recognize constitutionally symmetric nodes, i.e., nodes that are topologically equivalent in all respects. This step and the generation of unique node order, breaking all ties, graph construction, and identification are all essential parts of the CANGEN process.

Combinatorial and extended sums methods are two different approaches for characterization of graph nodes and their environments. The combinatorial process is suitable for analyses of small graphs (simple chemical structures) but becomes too cumbersome for more complex ones because of the need to characterize each node environment completely. Simple, exhaustive solutions that have orders of max $(N,E)!$ become impractical as N increases beyond 15. Partial characterization is therefore often attempted and is adequate for most symmetry perception problems. Such algorithms use a general approach of breadth-first optimization of a tree.[3] Nodes are characterized successively deeper into the total graph until the combined characterization is adequate. This usually reduces the base of the algorithmic order of N or E to the number of edges in the shortest path between the most distant nodes. However, these algorithms do not avoid the problem of factorial order for the general case.[4]

The sums method achieves greater efficiency by limiting the use of a combined description of connected nodes while ignoring all path-specific topological information. A sum vector S is modified iteratively by summing over the S elements of

* Address correspondence to this author at 809 Karenwald Lane, Schenectady, NY 12309.

0095-2338/89/1629-0097$01.50/0