Computer Generation of Vertex Degree Sets for Chemical Graphs from a Number of Vertices and Rings

Lowell H. Hall* and Jeffrey B. Fisk

Department of Chemistry, Eastern Nazarene College, Quincy, Massachusetts 02170

Received May 9, 1994®

A computer algorithm is presented for the generation of vertex degree sets for simple graphs from a given number of vertices and rings. The algorithm is based on a relation developed by Hall and Kier: $^1D - {}^3D - 2{}^{*4}D - {}^5D - 4{}^{*6}D = 2 - 2R$, involving vertex degree count, ^kD, and ring count, R. The algorithm is applied to graphs of chemical interest, limiting the maximum degree count to 4, although the process can be extended to higher order vertices. In addition, a procedure was developed for eliminating pseudographs and multigraphs from the generation process.

INTRODUCTION

A long time goal of QSAR analysis is the ability to develop a list of structures for molecules which are predicted to have property values in a desired range of values. The development over the past 18 years of indices of molecular structures drawn from chemical graph theory has provided a basis for advancement toward this goal.[1-4]

A central issue in the molecule generation problem is representation of a molecular structure in a way which facilitates both development of QSAR models and solution of the inverse of the QSAR problem: inversion of QSAR equations to generate structures consistent with a target activity range. The molecular connectivity χ indices, the κ shape indices, and the newer electrotopological states indices provide structure representations which serve as the basis for molecule generation.[5]

Two important links in the structure generation process have now been addressed. The first is the actual generation of graphs which represent molecular strutures. The work of Faradzhev[6] and Zefirov[7] has resulted in a computer algorithm for such generation. Faradzhev demonstrated an algorithm for generation of molecular graphs for vertex degree sets. Proof was presented that the algorithm is both exhaustive and nonredundant, thus avoiding the graph isomorphism problem for simple graphs.[6] Zefirov and his group have shown how to take QSAR equations based on the κ shape indices and on the $^1\chi$ index and generate structures using the Faradzhev algorithm.[7]

The second link was recently presented by Hall and Kier in a series of papers which developed the foundations of the inverse QSAR process.[8-11] Properties of chemical graphs were examined, and general relations were suggested via an inductive approach. This method involves the relations between path counts (which may be obtained from χ or κ indices in QSAR equations) and vertex degree sets. A final set of interconversion equations was rigorously derived.[9]

A set of interconversion equations was applied, in an inverse QSAR sense, to generate a list of molecules estimated to possess a property value in a preselected target range. The first example dealt with molar volume for alkanes.[8,10] The second example involves the isonarcotic activity for a set of esters, alcohols, and ketones in a fourth paper.[11] In these cases a limited number of degree sets was generated from a small number of path-count sets which were dictated by the QSAR equations together with the preselected target property value. This process meets criteria for the inverse QSAR problem but is a limited approach to the degree set generation problem.

The development of the fundamental equations in the first two papers,[8,9] however, suggests that it is possible and practical to generate all degree sets for a given number of atoms and rings. This paper presents the results of that investigation along with a simple computer algorithm for the vertex degree set generation. Recently, Bieber and Jackson[12,13] described a method for degree set generation which depends on a set of complex interrelationships among atom and molecule types. This present work presents conceptually a simpler and more straightforward method for degree set generation.

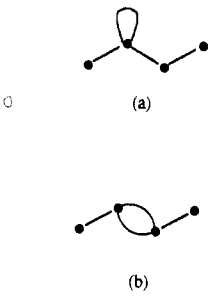


Figure 1. (a) A pseudograph for the degree set [2101] and (b) a multigraph for the degree set [2020].

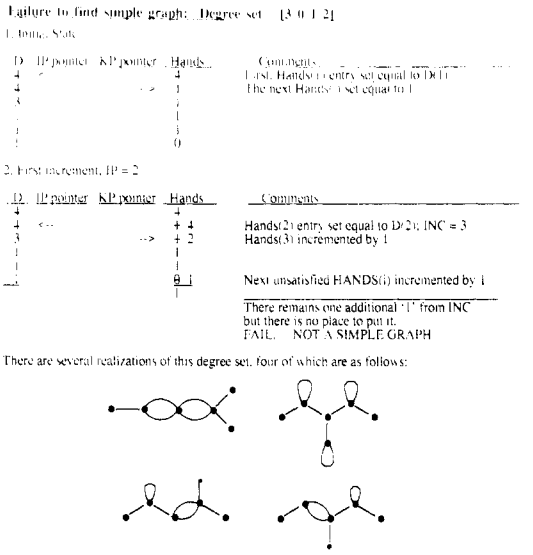


Figure 2. Illustration of the process for determining that a degree set does not correspond to a simple graph.

* Abstract published in *Advance ACS Abstracts*, August 15, 1994.

0095-2338/94/1634-1184$04.50/0

Table 1. Listing of FORTRAN Program for Degree Set Generation for Simple Graphs

```
      PROGRAM DEGREE
C
      INTEGER D1, D2, D3, D4, A, R, RMAX, RMIN, AMAX, AMIN
      INTEGER RMIN1, RMAX1, ALIMIT
      INTEGER COUNT, P1, PCOUNT, P2, P1MAX, RMAX
      INTEGER D(100), HANDS(100), UNSAT, REM
      INTEGER FAIL
      CHARACTER*5 SYMB
      CHARACTER SCREEN
C-----------------------------------------------------------------------
      OPEN (UNIT=11, FILE='DEGREE.OUT', STATUS='NEW',
     -    FORM='FORMATTED')
      OPEN (UNIT=12, FILE='DEGREE.NOSOL', STATUS='NEW',
     -    FORM='FORMATTED')
C-----------------------------------------------------------------------
C Input setup section
      WRITE (9,41)
41    FORMAT (//' Program Degree Set'//)
      WRITE (12,42)
42    FORMAT (//' FILE OF NO GRAPH SOLUTIONS '/)
      WRITE (11,41)
50    WRITE (9,96)
96    FORMAT (/' Do you want results printed on screen (N) ? :'$)
      READ (9,97) SCREEN
97    FORMAT (A)
      LENGTH = LEN(TRIM(SCREEN))
      IF (LENGTH.EQ.0) THEN
        SCREEN = 'N'
      ELSE IF (SCREEN.EQ.'N'.OR.SCREEN.EQ.'n') THEN
        SCREEN = 'N'
      ELSE IF (SCREEN.EQ.'Y'.OR.SCREEN.EQ.'y') THEN
        SCREEN = 'Y'
      ELSE
        GO TO 50
      ENDIF
      WRITE (9, 101)
101   FORMAT (' Enter RMIN :'$)
      READ (9,103) RMIN
      WRITE (9, 102)
102   FORMAT (' Enter RMAX :'$)
      READ (9,103)  RMAX
103   FORMAT (I5)
      WRITE (9,104)
104   FORMAT (' Enter AMIN :'$)
      READ (9,103) AMIN
        WRITE (9,106)
106     FORMAT (' Enter AMAX :'$)
        READ (9,103) AMAX
C  Begin computational section
        RMIN1 = RMIN + 1
        RMAX1 = RMAX + 1
        DO 10000 A=AMIN,AMAX
          WRITE (11,11)
          WRITE (12,11)
11        FORMAT (/'  Input          Output'/
     -      '  A  R    D1  D2  D3  D4  p1  p2')
          DO 9000 IR=RMIN1,RMAX1
            R = IR - 1
C  Limit number of rings, R, to maximum posible for this number of atoms. A
            P1MAX = (A*(A - 1))/2
            RMAX = P1MAX - (A - 1)
            IF (R.GT.RMAX) GO TO 10000
            DO 8000 D1 = 0,A
              ALIMIT = A - D1
              DO 7000 D3 = 0,ALIMIT
                COUNT = D1 - D3 + 2*R - 2
                IF (COUNT.LT.0) THEN
                  D4 = COUNT/2
                  GO TO 7000
                ENDIF
                D4 = COUNT/2
                D2 = A - D1 - D3 - D4
                IF (D2.LT.0) THEN
                  GO TO 7000
                ENDIF
C Check with Handshaking formula
                P1 = R + A - 1
                PCOUNT = D1 + 2*D2 + 3*D3 + 4*D4
                IF (PCOUNT.NE.(2*P1)) THEN
                  PCOUNT = PCOUNT/2
                  GO TO 7000
                ENDIF
C-----------------------------------------------------------------
C Set symbol for simple graph
                SYMB = '  '
```

METHOD

Fundamental Relations. The derivation and proof presented by Hall and Kier lead to the following equations

$$^1D + {}^4D + 3{}^5D + 6{}^6D = {}^2p - {}^1p + 3 - 3R \quad (1)$$

$$^2D - 3{}^4D - 8{}^5D - 15{}^6D = -2{}^2p + 3{}^1p - 3 + 3R \quad (2)$$

$$^3D + 3{}^4D + 6{}^5D + 10{}^6D = {}^2p - {}^1p + 1 - R \quad (3)$$

in which ^kD is the count of vertices with vertex degree k; ^np is the count of paths of the length n edges; and R is the number of (independent) rings in the hydrogen-suppressed graph, also known as the cyclomatic number: $\mu = R = {}^1p - (A - 1)$ where A is the number of vertices in the graph (or number of non-hydrogen atoms in the hydrogen-suppressed graph). Vertex degree is defined as the number of edges connecting a vertex to its neighboring vertices. These equations were derived explicitly for vertices with degree 1–6. Extension to higher order vertex degree can be done following the derivation set forth by Hall and Kier.[9] This paper will be limited to vertex degrees 1–4, but extension to higher order vertices is possible.

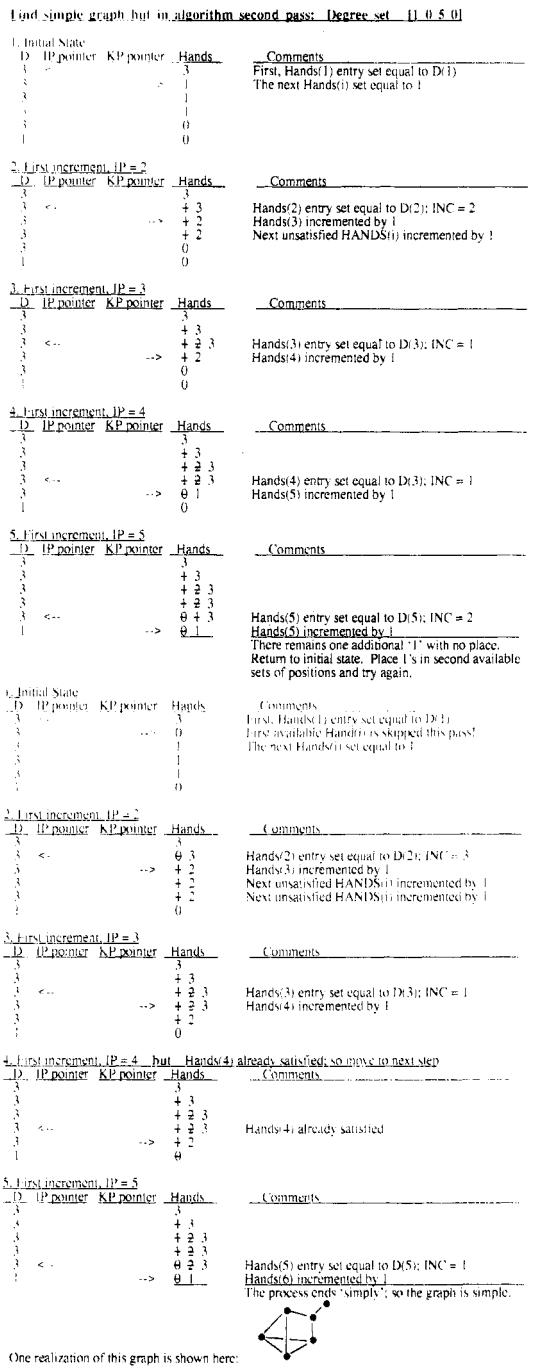


Figure 3. Illustration of the process for determining that a degree set corresponds to a simple graph. Two passes are required.

In their papers Hall and Kier pointed out that another relation can be obtained from these three interconversion equations (1–3), one which depends only upon vertex degrees:

$$^1D - {}^3D - 2{}^4D - 3{}^5D - 4{}^6D = 2 - 2R \quad (4)$$

Equation 4 is only given up to sixth order vertices, but it can be extended according to the method set forth by Hall and Kier.[9] This relation is important because it places restriction on the composition of any simple graph. It is rigorously true for simple graphs and is the basis of an algorithm for vertex

Table 2. Program Section for Checking for Pseudograph or Multigraph (Continuation from Table 1)

```
C Check routine for multigraph or pseudo graph
C
C  Setup list of degrees in descending order
C-----------------------------------------------------------------
                I = 0
                IF (D4.NE.0) THEN
                  DO 1110 J=1,D4
                    I = I + 1
                      D(I) = 4
1110              CONTINUE
                ENDIF
                IF (D3.NE.0) THEN
                  DO 1120 J=1,D3
                    I = I + 1
                      D(I) = 3
1120            CONTINUE
                ENDIF
                IF (D2.NE.0) THEN
                  DO 1130 J=1,D2
                    I = I + 1
                      D(I) = 2
1130            CONTINUE
                ENDIF
                IF (D1.NE.0) THEN
                  DO 1140 J=1,D1
                    I = I + 1
                      D(I) = 1
1140            CONTINUE
                ENDIF
C
                IF (I.GT.A) WRITE (9,1141) I,A
1141            FORMAT (//' *** Pointer count,'I4
     -          ' exceeds Atom count',I4//)
C-----------------------------------------------------------------
C Setup HANDS vector to accummulate connections
C  Initialize to 0
C  Set FAIL to 0; FAIL is incremented each time there is a failure
C  until a fatal failure
C
      FAIL = 0
1150  CONTINUE
                DO 1160 I=1,A
                  HANDS(I) = 0
1160            CONTINUE
C  Handle first atom
C  Put its degree into HANDS(1)
C  Put 1's into the next D(1) positions of HANDS
C
C  First check to see whether the required number of 1's exceeds the remaining
C  number of atoms
                IF ((D(1) + 1 + FAIL).GT.A) THEN
                  SYMB = ' NG-1'
                  GO TO 4000
                ENDIF
C
                DLIM = D(1)
                HANDS(1) = D(1)
                DO 1170 I=1,DLIM
                  HANDS(1 + I + FAIL) = 1
1170            CONTINUE
C
C  Continue with remaining atoms; set up loop.
C
                DO 3000 IP=2,A
C
C  Do a check
C  If HANDS(I) = D(I), this atom is already satisfied; go to next atom.
                  IF (HANDS(IP).EQ.D(IP)) GO TO 3000
C
C  Check for last atom.
C  If last atom is satisfied [ HANDS(IP) = D(IP) ], then checks are complete;
C     This is a simple graph.
C  If atom is not satisfied, then check for unsatisfied = inc
C  If this equality exists, then graph is simple;
C     last atom can be exactly incremented
C  If this equality does not exist, then this is not a simple graph.
C
                  INC = D(IP) - HANDS(IP)
C
                  IF (IP.EQ.A) THEN
                    IF (HANDS(IP).LE.D(IP)) THEN
C  Last atom already satisfied; so this is a simple graph.
C     or
C  Last atom exactly satisfied with exactly the increment available, INC;
C     or
C  Last atom can be incremented to D;
C     This is a simple graph.
                      GO TO 4000
                    ENDIF
                  ENDIF
C  Handle current atom at point = IP
                  HANDS(IP) = D(IP)
C  Increment INC atoms by 1, starting with the next INC atoms.
C     Three conditions to check:
C     1- the next atom is already satisfied in which case, simply procede to
C        the following atom.  Bump Hands pointer KP.
C     2- one of atoms to be incremented is the last atom AND the remaining
C        increment is greater than 1, in which case the last atom may be
C        incremented by more than 1.
C     3- The terminating condition for a multigraph or pseudograph:
C        There are remaining increments but no remaining atoms.
C
C  KP is the pointer in the HANDS vector at one position beyond IP.
C  Start loop to incrementing by 1's.
                  KP = IP + 1
C
                  DO 2000 K=1,INC
1200              CONTINUE
C  Last atom check in the HANDS vector
C  If this atom is last atom AND the remaining increment is more than 1 but
C  not to much, then satisfy this atom by incrementing more than 1.
C  If the 1's from atom one have not yet reached the last position (KP=A), then
C  increment FAIL by one and go back and try again.
C
                  IF (KP.EQ.A) THEN
                    UNSAT = D(KP) - HANDS(KP)
                      REM = INC - (K - 1)
                      IF (REM.GT.UNSAT) THEN
                        IF ((D(1) + 1 + FAIL).EQ.A) THEN
                          SYMB = ' NG-5'
                          GO TO 4000
                        ENDIF
C  Failure of present setup; go back and try with 1's moved down one position
                        FAIL = FAIL + 1
                        GO TO 1150
                      ELSE IF (REM.EQ.UNSAT) THEN
                        HANDS(KP) = HANDS(KP) + REM
                        GO TO 3000
                      ELSE
                        HANDS(KP) = HANDS(KP) + 1
                        GO TO 3000
                      ENDIF
                    ENDIF
C
C  If current atom at KP is satisfied, bump KP and go to start of increment loop.
                      IF (HANDS(KP).EQ.D(KP)) THEN
                        KP = KP + 1
                        GO TO 1200
                      ENDIF
C  Go ahead and increment this atom at pointer value KP
                      HANDS(KP) = HANDS(KP) + 1
                      KP = KP + 1
C  End of HANDS increment loop
2000              CONTINUE
C  End of Degree vector loop
3000            CONTINUE
C  Jump out point for last atom check in a simple graph
4000          CONTINUE
C
C  Compute p2
                P2 = D2 + 3*D3 + 6*D4
C
                IF (SCREEN.EQ.'Y') THEN
                  WRITE ( 9,113) A,R, D1,D2,D3,D4, P1,P2, SYMB
                ENDIF
                IF (SYMB.EQ.'    ') THEN
                  WRITE (11,113) A,R, D1,D2,D3,D4, P1,P2, SYMB
                ELSE
                  WRITE (12,113) A,R, D1,D2,D3,D4, P1,P2, SYMB
113               FORMAT (214,3X,415, 2X, 215, A)
                ENDIF
C
7000          CONTINUE
8000        CONTINUE
            WRITE (11,8999)
            WRITE (12,8999)
8999        FORMAT ( )
9000      CONTINUE
10000   CONTINUE
C
        WRITE (9,1001)
1001    FORMAT (' Program Degree in Completed.  Push Return.')
        PAUSE
        END
```

degree set generation. Given a value for R, it is possible to generate vertex degree sets which satisfy eq 4, a relation on integer values ${}^kD$ and R.

It should be observed that ${}^2D$ is absent from eq 4. Vertices of degree 2 are referred to as spacers in the graph. Such spacer vertices are more closely related to the overall size of the graph than to the character of the graph in terms of symmetry and shape. Hall and Kier point out an analogous relation involving edge types in which the count of edges between vertices of degree 2, also considered as spacers, is absent from the equation.[10]

Computer Algorithm. The algorithm for degree set generation consists of three parts:

(1) input of values for R (number of independent rings) and A (number of atoms)

(2) (a) generating values for ${}^3D$ and ${}^4D$ in DO LOOPS

(b) computing ${}^1D$ from eq 4

(c) computing ${}^2D$ from $A = \sum {}^kD$

(3) (a) check degree set for validity as a graph

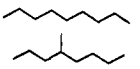
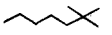
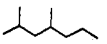
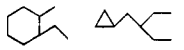
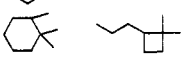
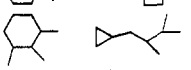
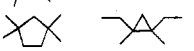
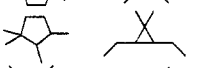
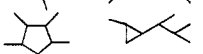
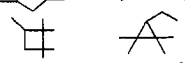
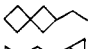
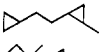
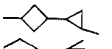
(b) check degree set to eliminate multigraphs and pseudographs

For each valid degree set, additional information may be generated, such as, the length of the longest path for acyclic graphs or the maximum number of vertices in rings for cyclic graphs.

Graph Validity Checks. Each set of vertex degrees is checked for validity as a graph in three ways. Each degree set must satisfy the following conditions. (1) A vertex degree count ${}^kD$ is a nonnegative because it is a count:

$$ {}^kD \geq 0 \qquad \text{for all } k $$

Table 3. Vertex Degrees Generated According to the Algorithm Described in This Paper for Nine Vertexes, with Degrees from 1- to 4, and up to the Maximum Number of Rings Allowed for Those Values, 10[a]

Vertices	Rings	^1D	^2D	^3D	^4D	^1p	^2p	Number of graphs	Typical graph
9	0	2	7	0	0	8	7	1	
9	0	3	5	1	0	8	8	5	
9	0	4	4	0	1	8	10	5	
9	0	4	3	2	0	8	9	9	
9	0	5	2	1	1	8	11	8	
9	0	5	1	3	0	8	10	3	
9	0	6	1	0	2	8	13	2	
9	0	6	0	2	1	8	12	2	
9	1	0	9	0	0	9	9		
9	1	1	7	1	0	9	10		
9	1	2	6	0	1	9	12		
9	1	2	5	2	0	9	11		
9	1	3	4	1	1	9	13		
9	1	3	3	3	0	9	12		
9	1	4	3	0	2	9	15		
9	1	4	2	2	1	9	14		
9	1	4	1	4	0	9	13		
9	1	5	1	1	2	9	16		
9	1	5	0	3	1	9	15		
9	1	6	0	0	3	9	18		
9	2	0	8	0	1	10	14		
9	2	0	7	2	0	10	13		
9	2	1	6	1	1	10	15		
9	2	1	5	3	0	10	14		
9	2	2	5	0	2	10	17		
9	2	2	4	2	1	10	16		
9	2	2	3	4	0	10	15		
9	2	3	3	1	2	10	18		
9	2	3	2	3	1	10	17		
9	2	3	1	5	0	10	16		
9	2	4	2	0	3	10	20		
9	2	4	1	2	2	10	19		
9	2	4	0	4	1	10	18		
9	2	5	0	1	3	10	21		
9	3	0	7	0	2	11	19		
9	3	0	6	2	1	11	18		
9	3	0	5	4	0	11	17		
9	3	1	5	1	2	11	20		
9	3	1	4	3	1	11	19		
9	3	1	3	5	0	11	18		
9	3	2	4	0	3	11	22		
9	3	2	3	2	2	11	21		
9	3	2	2	4	1	11	20		
9	3	2	1	6	0	11	19		
9	3	3	2	1	3	11	23		
9	3	3	1	3	2	11	22		
9	3	3	0	5	1	11	21		
9	3	4	1	0	4	11	25		
9	3	4	0	2	3	11	24		

[a] For graphs with up to three rings, one or two typical graphs are given as an illustration.

The handshaking Lemma[13] must be satisfied:

$$\sum k^{*k}D = 2\,{}^1p \qquad \text{where } {}^1p = R + (A - 1)$$

(3) If these criteria are met, the degree set corresponds to a graph. However, it is possible the graph is not simple. If the corresponding graph contains a loop (an edge which begins and ends on the same vertex), the graph is a pseudograph. For example, the graph shown in Figure 1a is a pseudograph for the degree set [2 1 0 1]. If two vertices are connected by more than one edge, the graph is called a multigraph, as illustrated in Figure 1b for the degree set [2 0 2 0].

There is no simple test for multigraphs or pseudographs which can be written in algebraic form. To test for such nonsimple graphs, one must engage in at least part of the process of construction of the graph from the degree set.

Our view of this process is that of the Handshaking Lemma. The vertex degrees may be thought of as a set of hands; the edges or connections may be seen as handshakes. In a pseudo- or multigraph there are more hands than handshakes possible for the given R and A and for a simple graph.

We have developed an algorithm which detects multigraphs and pseudographs. The essence of the method consists of building the adjacency matrix while keeping track only of the accumulation of connections rather than recording the whole matrix. The addition of a "1" in matrix location ij (row i, column, j) in an adjacency matrix results in incrementing the number of connections to vertex i as well as to vertex j (row j, column i). For each vertex, the degree is initialized to zero and incremented in the algorithm. The vertices are considered consecutively, starting with vertices of highest degree and proceeding to those of lowest degree in a given degree set.

For a simple graph, the end of the process of placing 1's in the adjacency matrix (i.e., incrementing connections) corresponds exactly with incrementing the degree of the last vertex to its degree value. By contrast for both the multigraph and pseudograph there are more 1's to place (in row j, column i)

Table 4.

vertices	rings	^1D	^2D	^3D	^4D	^1p	^2p
9	4	0	6	0	3	12	24
9	4	0	5	2	2	12	23
9	4	0	4	4	1	12	22
9	4	0	3	6	0	12	21
9	4	1	4	1	3	12	25
9	4	1	3	3	2	12	24
9	4	1	2	5	1	12	23
9	4	1	1	7	0	12	22
9	4	2	3	0	4	12	27
9	4	2	2	2	3	12	26
9	4	2	1	4	2	12	25
9	4	2	0	6	1	12	24
9	4	3	1	1	4	12	28
9	4	3	0	3	3	12	27
9	4	4	0	0	5	12	30
9	5	0	5	0	4	13	29
9	5	0	4	2	3	13	28
9	5	0	3	4	2	13	27
9	5	0	2	6	1	13	26
9	5	0	1	8	0	13	25
9	5	1	3	1	4	13	30
9	5	1	2	3	3	13	29
9	5	1	1	5	2	13	28
9	5	1	0	7	1	13	27
9	5	2	2	0	5	13	32
9	5	2	1	2	4	13	31
9	5	2	0	4	3	13	30
9	5	3	0	1	5	13	33
9	6	0	4	0	5	14	34
9	6	0	3	2	4	14	33
9	6	0	2	4	3	14	32
9	6	0	1	6	2	14	31
9	6	0	0	8	1	14	30
9	6	1	2	1	5	14	35
9	6	1	1	3	4	14	34
9	6	1	0	5	3	14	33
9	6	2	1	0	6	14	37
9	6	2	0	2	5	14	36
9	7	0	3	0	6	15	39
9	7	0	2	2	5	15	38
9	7	0	1	4	4	15	37
9	7	0	0	6	3	15	36
9	7	1	1	1	6	15	40
9	7	1	0	3	5	15	39
9	7	2	0	0	7	15	42
9	8	0	2	0	7	16	44
9	8	0	1	2	6	16	43
9	8	0	0	4	5	16	42
9	9	0	0	2	7	17	48
9	10	0	0	0	9	18	54

than can be accommodated by incrementing the vertex degees associated with the developing adjacency matrix.

The validation algorithm is given in Table 1 as a FORTRAN program. An example is given in Figure 2 to illustrate a degree set found not to correspond to a simple graph and in Figure 3 for a degree set which does correspond to a simple graph.

RESULTS

Degree sets have been generated for several input values of A and R. Two sets were generated to check the accuracy of the program. For the first test, degree sets were generated for alkane graphs from 2 to 12 vertices. These correspond to the familiar acyclic alkanes from ethane through all the dodecanes. These degree sets were generated and checked against the well-known list of these molecules. For this case the maximum degree was limited to 4. These are 67 degree sets for 663 graphs.

For the second test, degree sets were generated for $A = 2$–6 and from $R = 0$ to the maximum ring number for each value of A. This maximum value occurs for the complete graph in which each vertex is connected to each other vertex. The number of edges in the complete graph is $^1p = {}^1/_2A(A-1)$. Thus, $R_{max} = {}^1P_{max} - (A-1) = {}^1/_2A(A-1) - (A-1) = (A-1)(A-1)/2 = (A-1)^2/2$. The generated vertex degree sets were compared to the list of graphs given by Harary in Appendex I of his classic book.[14] Only connected graphs were considered. The maximum degree for this test set was set at 5, the maximum possible for these graphs. Both of these test sets of vertex degrees agreed exactly with the existing lists. There are 142 graphs in this test set. A listing of these degree sets along with a program listing may be obtained from the authors.

One additional set of vertex degrees was generated, for $A = 9$ and R up to the maximum possible value with the maximum vertex degree set at 4. In this case, $R_{max} = 10$. These degree sets are listed in Tables 3 and 4. A representative graphs is given for each degree set up to those with three rings.

DISCUSSION

The vertex degree set computer program is short and easily used. The user supplies a range of values for A and for R. The program computes the maximum possible R as a check on the input range for R for a given A. The validity checking routines ensure that degree sets correspond to simple graphs. If a user desires, code could be added to compute additional quantities which depend on the vertex degrees, A, and R. These quantities include the longest path for acyclic graphs and the maximum number of vertices which can be found in rings for cyclic graphs.[8,9]

The number of graphs which correspond to a given vertex degree set varies considerably. The following sets correspond to one graph: [2 n 0 0] corresponds to an unbranched graph in which $A = n + 2$; [0 n 0 0], which corresponds to the unbranched monocyclic graph with n vertices. On the other hand, [5 2 1 1] generates eight graphs, and [4 3 2 0] generates nine graphs.

CONCLUSIONS

The algorithm presented here permits generation of all the degree sets for simple graphs consistent with A atoms and R rings according to eq 4. This procedure permits generation of degree sets which may then generate graphs *via* the Faradzhev algorithm. In this manner complete sets of vertex degrees may be considered in various applications, such as the inverse QSAR problem, *de novo* drug design, and construction of complete libraries of hydrogen-suppressed graphs.

REFERENCES AND NOTES

(1) Balaban, A. T.; Motoc, I.; Bonchev, D.; Menkenyan, O. *Top. Curr. Chem.*, **1983**, *114*, 21–55.

(2) Kier, L. B; Hall, L. H. *Molecular Connectivity in Structure-Activity Analysis*; John Wiley and Sons; Research Studies Press: Chichester, England, 1986.

(3) Hall, L. H.; Kier, L. B. The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Relations. In *Reviews of Computational Chemistry*; Boyd, D.; Lipkowitz, K., Eds., VCH Publishers, Inc.: 1991; Chapter 9, pp 367–422.

(4) Trinajstić, N. *Chemical Graph Theory*; CRC Press: Boca Raton, FL, 1983; Vols. I and II.

(5) Hall, L. H. Computational Aspects of Molecular connectivity and its Role in Structure-Property Modeling. In *Computational Chemical Graph Theory*; Rouvray, D. H., Ed.; Nova Press: New York, 1990; Chapter 8, pp 202–233.

(6) Faradzhev, I. Generation of Nonisomorphic Graphs with Given Partition of Vertex degrees. In *Algorithmic Investigations in Combinatorics*; Faradzhev, I., Ed.; Nauka: Moscow, 1978; pp 11–19 (in Russian).

(7) (a) Baskin, I. I.; Gordeeva, E. V.; Devdariani, R. O.; Zefirov, N. S.; Palyulin, V. A.; Stankevitch, I. V. Solving the Inverse Problem of Structure-Property Relations for the Case of Topological; Indexes. *Dokl. Akad. Nauk. USSR* **1989**, *307*, 613–617. (b) Skvortsova, M. I.; Baskin, I. I.; Slovokhotova, O. L.; Palyulin, V. A.; Zefirov, N. S. The Inverse Problem in QSAR/QSPR Studies for the Case of Topological Indices Characterizing Molecular Shape (Kier Indices). *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 630–634. (c) Zefirov, N. S.; Palyulin, V. A.; Radchenko, Y. V. Problem of Generating Structures with Given Properties. Solution of the Inverse Problem for the Case of the Centric Balaban Index. *Dokl. Akad. Nauk. USSR* **1991**, *316*, 921–924.

(8) Kier, L. B.; Hall, L. H.; Frazer, J. W. Design of Molecules from Quantitative-Structure Relationship Models. 1. Information Transfer between Path and Vertex Degree Counts. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 143–147.

(9) Hall, L. H.; Kier, L. B.; Frazer, J. W. Design of Molecules from Quantitative-Structure Relationship Models. 2. Derivation and Proof of Information Transfer Relating Equations. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 148–152.

(10) Hall, L. H.; Kier, L. B. Design of Molecules from Quantitative-Structure Relationship Models. 3. Role of Higher Order Path Counts: Path 3. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 598–603.

(11) Kier, L. B.; Hall, L. H. The Generation of Molecular Structures from a Graph-Based Equation. *Quant. Struct.-Act. Relat.* **1994**, *12*, 383–388.

(12) Bieber, T. I.; Jackson, M. Applications of Degree Distribution. 1. (a) General Discussion of Computer Generation of Degree Distributions. *J. Chem. Inf. Comput. Sci.* **1993**, *33*. (b) Bieber, T. I.; Jackson, M. Maximal Degree Distributions, *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 696–700.

(13) Bieber, T. I.; Jackson, M. Applications of Degree Distribution. 2. Construction and Enumeration of Isomers in the Alkane Series. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 701–708.

(14) Wilson, J. R. *Introduction to Graph Theory*; Oliver and Boyd: Edinburgh, 1972; Chapter 2.

(15) Harary, F. *Graph Theory*; Addison-Wesley Pub. Co.: Reading, MA, 1972; Appendix 1.