A SIMPLE ALGORITHM FOR COMPUTING
TOPOLOGICAL INDICES OF DENDRIMERS

M. GHORBANI and M. JALALI
INSTITUTE OF NANOSCIENCE AND NANOTECHNOLOGY,
UNIVERSITY OF KASHAN, KASHAN 87317-51167, IRAN
EMAIL: GHORBANI@KASHANU.AC.IR

ABSTRACT. Dendritic macromolecules’ have attracted much attention as organic examples of well-defined nanostructures. These molecules are ideal model systems for studying how physical properties depend on molecular size and architecture. In this paper using a simple result, some GAP programs are prepared to compute Wiener and hyper Wiener indices of dendrimers.

Keywords: Dendrimer, Wiener index, Hyper Wiener index, GAP.


1. INTRODUCTION

Dendrimers are macromolecular nanoscale objects that are widely recognized as precise, mathematically defined, covalent core-shell assemblies. Since dendrimers are well defined organic molecules in the size range of (1 to 15) nm and are known to act as hosts for guest molecules, they are promising candidates as templates for the formation of inorganic nanoclusters [1].

Let \( G \) be a simple molecular graph without directed and multiple edges and without loops, the vertex and edge-shapes of which are represented by \( V(G) \) and \( E(G) \), respectively. The graph \( G \) is said to be connected if for every vertices \( x \) and \( y \) in \( V(G) \) there exists a path between \( x \) and \( y \). In this paper we only consider connected graphs. The distance between a pair of vertices \( u \) and \( v \) of \( G \) is denoted by \( d(u, v) \).

1Corresponding Author.

Received 28 May 2007; Accepted 9 December 2007
©2007 Academic Center for Education, Culture and Research TMU
A topological index is a real number related to a molecular graph. It must be a structural invariant, i.e., it does not depend on the labelling or the pictorial representation of a graph. There are several topological indices have been defined and many of them have found applications as means to model chemical, pharmaceutical and other properties of molecules. Here, we consider two of topological indices containing Wiener and hyper Wiener of dendrimers. These topological indices define as follows:

- The Wiener index $W(G)$ [2-4] of a molecular graph $G$ is defined as the sum of the distances between all pairs of vertices. In other words,
  $$W(G) = \frac{1}{2} \left( \sum_{i=1}^{n} P_i \right),$$
  where $P_i$ is length of the path that contains the least number of edges between vertex $i$ and vertex $j$ in graph $G$ and $n$ is the maximum possible number of $i$ and $j$.

- The Wiener matrix [5] introduced by Randic, as a potential source of structural invariants for need of QSAR. The $(i, j)$ entry of the Wiener matrix is based on enumeration of paths in a graph which contain the path $(i, j)$. Then the hyper Wiener index $R$ can be expressed as the half sum of the Wiener matrix entries.

The goal of this article is to obtain some computer programs for calculating the Wiener and hyper Wiener indices of dendrimers, Figure 1. We apply our program on some dendrimers to compute their Wiener and hyper Wiener indices.

2. Computational Details

GAP [6] stands for Groups, Algorithms and Programming. The name was chosen to reflect the aim of the system, which is a group theoretical software for solving computational problems in group theory. The last years have seen a rapid spread of interest in the understanding, design and even implementation of group theoretical algorithms. These are gradually becoming accepted both as standard tools for a working group theoretician, like certain methods of proof, and as worthwhile objects of study, like connections between notions expressed in theorems. GAP was started as an attempt to meet this interest. This software was constructed by GAP’s team in Aachen. We encourage the reader to consult [1,7] for background materials and computational techniques related to applications of GAP in solving some problems in chemistry and biology.

To describe our method we first label the vertices of dendrimers $D$ by sequences of 0,1,2,3,4. We label the main root by 0, and then the first four neighbors of 0 by 01, 02, 03 and 04. The other vertices are labeled according
to its position in the lattice by sequence of \{0,1\}, \{0,2\}, \{0,3\} or \{0,4\}, see Figure 1. Therefore \(D\) can be divided to exactly four parts in which very part has the same lattice structure. We have the following crucial lemma:

**Lemma 2.1.** Suppose \(u = u_1u_2\ldots u_m\) and \(v = v_1v_2\ldots v_n\) are two vertices of \(D\) and \(u_1 = v_1, u_2 = v_2, \ldots, u_r = v_r\). Then \(d(u, v) = m + n - 2r\).

**Proof.** Suppose \(u\) and \(v\) are two arbitrary vertices of \(D\) such that \(u = u_1u_2\ldots u_m\) and \(v = v_1v_2\ldots v_n\) and \(u_1 = v_1, u_2 = v_2, \ldots, u_r = v_r\). To prove the lemma, we first assume that \(u\) and \(v\) lie in the same part of \(D\). Thus \(u_i\)'s and \(v_j\)'s are sequences of 0,\( p\) such that \(p \in \{1, 2, 3, 4\}\). In this case one can see that \(r \geq 2\) and \(d(u, v) = m - r + nr = m + n - 2r\). If \(u\) and \(v\) lie in two separate parts of \(D\), then \(r = 1\) and we must compute \(d(u, 0p)\) and \(d(v, 0q), p \neq q\). So \(d(u, v) = d(u, 0p) + d(v, 0q) + 2 = m + n - 2\).

Using this lemma we prepare a GAP program for computing the Wiener and hyper Wiener indices of dendrimers \(D\). We mention here, it is possible to revise our program for computing all distance definable topological indices. In Table I, we compute the values of Wiener and hyper Wiener indices of \(D\) for \(2 \leq k \leq 11\), where \(k\) denotes the \(k^{th}\) row. Here, if \(u = u_1u_2\ldots u_m\) we call \(m\) the length of \(u\).

### Table 1. Computing wiener and hyper wiener indices for \(k \leq 11\)

<table>
<thead>
<tr>
<th>(n = 2^{k+2} + 1)</th>
<th>Hyper Wiener Index</th>
<th>Wiener Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>44</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>900</td>
<td>216</td>
</tr>
<tr>
<td>29</td>
<td>9652</td>
<td>1688</td>
</tr>
<tr>
<td>61</td>
<td>77396</td>
<td>10488</td>
</tr>
<tr>
<td>125</td>
<td>526740</td>
<td>57720</td>
</tr>
<tr>
<td>253</td>
<td>3230484</td>
<td>295416</td>
</tr>
<tr>
<td>509</td>
<td>18448916</td>
<td>1442296</td>
</tr>
<tr>
<td>1021</td>
<td>100062228</td>
<td>6815736</td>
</tr>
<tr>
<td>2045</td>
<td>522008596</td>
<td>31455224</td>
</tr>
<tr>
<td>4093</td>
<td>2641981460</td>
<td>142598136</td>
</tr>
</tbody>
</table>

### A GAP Program for Computing Wiener and Hyper Wiener Index of Dendrimers

```gap
f:=function(k)
    local d1, i1, j1, x2, xx1, s1, p1, a1, b1, a1, n1, str, s2, zz, v, xx, sts, h, i, j, y1, y2, yyy,
        y3, yy, w, d, dd, iii, jjj, n, m, ii, ss, r, m, a, r, l, p, x, l, u, v, q, sss, qq, 11, 11;
    s1:=0; p1:=[ ]; m1:=0; qq:=0; 11:=[ ]; 11:=[ ]; k:=k-3;
    for i1 in [1..k] do
        s1:=s1+2^i1; Add(p1, s1);
    ```
od;
a1 := []; b1 := []; aa1 := []; x2 := []; xx1 := [];
for i1 in [2..5] do
    Add(aa1,[1,i1]);
od;
for i1 in p1 do
    for j1 in [i1..2*i1+1] do
        Add(a1,j1); Add(a1,2*j1+2); Add(b1,j1); Add(b1,2*j1+3);
        Add(aa1,a1); Add(aa1,b1); a1:=[ ]; b1:=[ ];
    od;
    od;
ww := []; p := 0; u := 0; v1 := []; ss := 0;
for i in [1..k] do
    Add(ww,0); Add(ww,1);
    od;
str:=[];ff:=[]; y:= [0,0]; y1:= [0,1]; y2:= [0,2]; y3:= [0,3]; yyy:= [];
s:=1; v:=[]; xx:=[]; sts:=[];
while s <= Length(ww) do
    Add(v,s); s:=s+1;
    od;
    h:= Combinations(v);
Sort(h, function(v,w) return Length(v) < Length(w); end);
s:=1; r:=1;
while s <= Length(h) do
    while r <= Length(h[s]) do
        Add(str,ww[h[s][r]]); r:=r+1;
    od;
    Add(sts,str); s:=s+1; r:=1;
    if Length(str) <= k and Length(str) >= 1 then
        AddSet(xx,str); fi;
    str:=[];
    od;
    for i in xx do
        for j in i do
            Add(y,j); Add(y1,j); Add(y2,j); Add(y3,j);
        od;
        Add(yyy,y); Add(yyy,y1); Add(yyy,y2); Add(yyy,y3);
        y:=[0,0]; y1:=[0,1]; y2:=[0,2]; y3:=[0,3];
    od;
    Add(yyy,[0,0]); Add(yyy,[0,1]); Add(yyy,[0,2]); Add(yyy,[0,3]);
    Add(yyy,[0]);
    Stor(yyy, function(v,w) return Length(v) < Length(w); end);
dd:=[];
for x in yyy do
d:=[[]];
for x1 in yyy do
    n:=Length(x);m:=Length(x1);
    if n < m then nn:=n;
    else
        nn:=m; fi;
    for ii in [1..nn] do
        if x[ii] <> x1[ii] then ii:=ii=i; break;fi;
    od;
    ss:=n+m-(2*ii);Add(d,ss);
    Add(dd,d);
od;
v1:=[[]];Add(v1,4);
for i in [2..2^(k+2)−3]do
    Add(v1,3);
    Add(v1,1);
    Print("++++++++++++++++++++++++++++++","\n");
    for i in dd do
        for j in i do
            p:=p+j;
        od;
    od;
    for j in [1..Length(dd)]do
        for i in [1..Length(dd)]do
            if j < i then qq:=qq+(dd[j][i])*(dd[j][i]+1);fi;
        od;
    od;
    Print("Hyper Wiener Indexxex=".qq."\n");
    Print("++++++++++++++++++++++++++++++","\n");
    Print("Wiener Indexxex=");
    return(p/2); Print("++++++++++++++++++","\n");
end;
Figure 1. 2D Graphical Representation of a Dendrimer
Acknowledgment. This research was in part supported by a grant from the Center of Excellence of Algebraic Methods and Applications of Isfahan University of Technology.

References