MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **60** (2008) 883-896

ISSN 0340 - 6253

Computing some topological indices by GAP program

A. Iranmanesh[*] and Y. Alizadeh

Department of Mathematics, Tarbiat Modares University

P. O. Box: 14115-137, Tehran, Iran

iranmana@modares.ac.ir

(Received February 14, 2008)

Abstract

In this paper we give a GAP program for computing the Wiener , Schultz, Szeged and the PI indices of any graph.

1. Introduction

One of the main distinctive characteristics of modern chemistry is the use of theoretical tools for the molecular modeling of physicochemical processes, chemical reactions, medicinal and toxicological events, etc., in which chemicals are involved. The success of the molecular modeling is judged by the insights that it offers on the nature of the processes studied, which permit better comprehension and a rational modification of them. These properties, measured experimentally, are almost invariably expressed in quantitative terms, think for instance of boiling point, refraction index, transition state energy, percentage of inhibition of some enzymatic activity, lethal dose, and so forth. The paradigm for the modeling of such properties is the relationship that exists between them and the molecular structure of chemical. This fact presupposes for the first challenge in the molecular modeling: the properties are expressed as numbers while the molecular structure is not. The way to solve this problem is by using molecular descriptors that are numbers representing information about different molecular features, to describe quantitatively the properties under study. These models are known as quantitative structure-property (**QSPR**) and

[*] corresponding author

quantitative structure-activity relationships (**QSAR**), depending on the physicochemical or biological nature of the properties studied, respectively.

Topological indices are numerical descriptors derived from the associate graphs of chemical compounds. Some indices based on the distances in graph are widely used in establishing relationships between the structure of molecules and their physico-chemical properties. Usage of topological indices in chemistry began in 1947 when the chemist Harold Wiener [1] introduced Wiener index to demonstrate correlations between physicochemical properties of organic compounds and the index of their molecular graphs. Wiener originally defined his index (W) on trees and studied its use for correlations of physicochemical properties of alkanes, alcohols, amines and analogous compounds [2]. Starting from the middle of the 1970s, the Wiener index gained much popularity and, since then, new results related to it are constantly being reported. For a review, historical details and further bibliography on the chemical applications of the Wiener index see [3-5].

Let G be a connected graph. The vertex-set and edge-set of G are denoted by V(G) and E(G), respectively. The degree of a vertex $i \in V(G)$ is the number of vertices joining to i and denoted by $v(i)$. The (i, j) entry of the adjacency matrix of G is denoted by $A(i, j)$.

The Wiener index of a graph G is denoted by *W (G)* and defined as the sum of distances between all pairs of vertices in *G*:

$$W(G)=\frac{1}{2}\sum_{\{i,j\}\subseteq V(G)} d(i,j) \qquad (1)$$

where $d(i, j)$ is the distance between vertices i and j. Another topological index is the Schultz index (MTI), introduced by Schultz in 1989, as the molecular topological index [6], and defined by:

$$MTI = \sum_{\{i,j\}\subseteq V(G)} v(i)(\, d(i,j) + A(i,j)\,) \qquad (2)$$

The molecular topological index was studied in many papers [7-11].

In a series of articles, the Wiener index of some nanotubes is computed [12-17]. Another topological index was introduced by Gutman and called the Szeged index, abbreviated as Sz [2]. Let *e* be an edge of a graph G connecting the vertices u and v. Define two sets of non-equidistant vertices with respect to *e*:

$$N_1(e|G)=\{u \in V(G) \,|\, d(x,u) < d(x,v)\}$$

$$N_2(e|G)=\{v\in V(G)|d(x,v)<d(x,u)\}$$

The number of elements of $N_1(e|G)$ and $N_2(e|G)$ are denoted by $n_1(e|G)$ and $n_2(e|G)$ respectively.

The Szeged index of the graph G is defined as

$$Sz(G)=\sum_{e\in E(G)} n_1(e|G)\cdot n_2(e|G) \quad (3)$$

By the reason of the coincidence of Wiener and Szeged indices, in the case of trees, the authors in [18] and [19] introduced another Szeged/Wiener-like topological index, called Padmakar-Ivan index, abbreviated as PI. This new index, denoted by PI(G), is defined on the ground of non-equidistant edges $m(e|G)$:

$$PI(G)=\sum_{e\in E(G)} m_1(e|G)+m_2(e|G) \quad (4)$$

Applications of the PI index to QSRP/QSAR were studied in [20]. The index was mostly compared with the Wiener and the Szeged index. It turned out the PI index has similar discriminating power as the other two indices and in many cases it gives better results. As we already mentioned, the Szeged index incorporates the distribution of vertices of a molecular graph, while the PI index does this for edges. Hence it seems that a combination of both could give good results in QSRP/QSAR studies. Indeed, the combination of the PI index and the Szeged index is the best for modeling polychlorinated biphenyls (PCBs) in environment among the three possible pairs of indices selected from the PI index, the Szeged index, and the Wiener index [20]. For the Wiener and the Szeged indices such studies were previously reported in [21,22]. The Szeged and PI indices of some nanotubes were computed and reported in [23-26].

The computing of Szeged and PI indices seems to be straightforward, but this is not true. For computing the Szeged or PI index, we must obtain $n(e|G)$ or $m(e|G)$ for any edge in the graph [23-32] and it is time consuming. In the present paper, we provide an algorithm, implemented in the GAP program, which proved to be faster than the direct implementation of definitions. The program also enables the calculation of the Wiener and Schultz indices.

2. An algorithm for computing the Wiener and Schultz indices of a graph

In this section, we give an algorithm that enables one to compute the Wiener and Schultz indices of any graph. The algorithm is as follows:

1- Assign to any vertex one index number.

2- Determine all adjacent vertices of the vertex i, $i \in V$ and denote this set by $\mathrm{N(i)}$.

3- Start the program, by setting w=0, Sc=0, and at the end of run, the values of $\frac{1}{2}w$ and Sc will be the Wiener and Schultz indices of the graph G, respectively.

4- The set of vertices having their distance to vertex i equal to $t\ (t \geq 0)$ is denoted by $\mathrm{D}_{\mathrm{i,t}}$ and consider $D_{i,0} = \{i\}$. We have the following relations:

$$V = \bigcup_{t \geq 0} D_{i,t} \quad i \in V \tag{5}$$

$$\sum_{j \in V(G)} d(i,j) = \sum_{t \geq 1} t \times \left| D_{i,t} \right| \ , \forall i \in V(G) \tag{6}$$

$$W(G) = \frac{1}{2} \sum_{i \in V, t \geq 1} t \times \left| D_{i,t} \right| \tag{7}$$

$$\begin{aligned} MTI(G) &= \sum_{i \in V(G)} v(i) \times \sum_{j \in V(G)} (d(i,j) + A(i,j)) \\ &= \sum_{i \in V(G)} v(i) \times \left(\sum_{j \in N(i)} 2 + \sum_{j \in V(G) \setminus N(i)} d(i,j) \right) \\ &= \sum_{i \in V(G)} \left(2v(i)^2 + v(i) \sum_{j \in D_{i,t},\, t \geq 2} t \times \left| D_{i,t} \right| \right) \end{aligned} \tag{8}$$

According to (7) and (8), by determining these sets, we can obtain Wiener and Schultz indices of the graph.

The distance between the vertex i and its adjacent vertices is equal to 1, therefore $\mathrm{D}_{\mathrm{i,1}} = \mathrm{N(i)}$. For each $j \in D_{i,t}$ $t \geq 1$, the distance between each vertex of set $N(j) \setminus (D_{i,t} \cup D_{i,t-1})$ and the vertex i is equal to $t+1$, thus we have:

$$D_{i,t+1} = \bigcup_{j \in D_{i,t}} (N(j) \setminus (D_{i,t} \cup D_{i,t-1})), \quad t \geq 1 \tag{9}$$

According to (9), we can obtain $D_{i,t}$, $t \geq 2$ for each $i \in V$.

GAP stands for Groups, Algorithms and Programming [33]. The name was chosen to reflect the aim of the system, which is theoretical software for solving computational problems in group theory. The last years, a rapid spread of interest in the understanding, design and even implementation of group theoretical algorithms was recorded. GAP software was built by GAP's team in Aachen. We encourage the reader to consult Refs. [34] and [35] for background materials and computational techniques related to applications of GAP in solving some problems in chemistry and biology.

According to the above algorithm, we developed a GAP program to compute the Wiener and Schultz indices of $TUC_4C_8(S)$ nanotubes.

Example 2.1. In [36], the Wiener index of $TUC_4C_8(S)$ nanotubes was reported and we can compute this index by the above program.

A C_4C_8 net is a trivalent decoration made by alternating C_4 and C_8. It can cover either a cylinder or a torus.

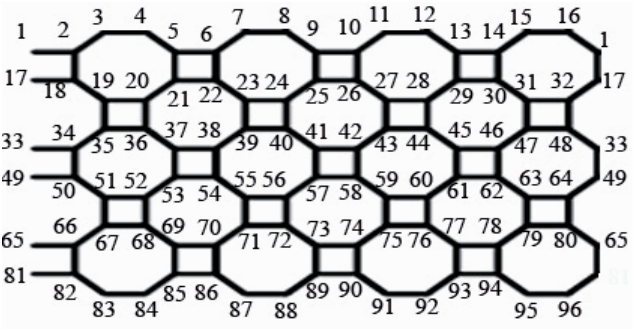

Figure1. $TUC_4C_8(S)$ nanotube, (p=4, q=6)

For computing of the Wiener and Schultz indices of $TUC_4C_8(S)$ nanotube by the above program, at first we assign to any vertex an index number (Figure 1). According to this numbering, the set of adjacent vertices to each vertex $1 \leq i \leq n$ is obtained by the following program (part 1). The Wiener index of the graph is computed by the 2nd part of the program.

We denote the number of octagons in one row by p and we denote the number of rows by q, $q \geq 2$. In each row there are $4p$ vertices and hence the number of vertices in this nanotube is equal to $4pq$.

We partition the vertices of this graph to the following sets:

K_1: The vertices of the first row.

K_2: The vertices of the rows 2m-1, $2 \leq m \leq \left\lceil \frac{q+1}{2} \right\rceil$.

K_3: The vertices of the rows 2m, $1 \leq m \leq \left\lceil \frac{q}{2} \right\rceil$.

K_4: The vertices of the final row.

We wrote a program to obtain the adjacent vertices set to each vertex in the sets K_i, i=1, …, 4. In this program, x is the index of vertex i in that row. The program computes the Wiener and Schultz indices of $TUC_4C_8(S)$ nanotube for arbitrary p and q.

```
p:=6; q:=10; # (For example)
n:=4*p*q;
N:=[];
k1:=[1..4*p];
v1:=[2..4*p-1];
for i in v1 do
 if i mod 4 in [1,2] then N[i]:=[i-1,i+1,i+4*p];
  else N[i]:=[i-1,i+1]; fi;
 N[1]:=[2,4*p,4*p+1]; N[4*p]:=[4*p-1,1];
od;
k:=[4*p+1..n-4*p];
k2:=Filtered(k,i->i mod (8*p) in [1..4*p]);
for i in k2 do
 x:=i mod (8*p);
 if (x mod 4) in [0,3] then N[i]:=[i-1,i+1,i-4*p];
  else  N[i]:=[i-1,i+1,i+4*p];fi;
 if x=1 then N[i]:=[i+1,i+4*p-1,i+4*p];fi;
 if x=4*p then N[i]:=[i-1,i-4*p,i-4*p+1]; fi;
od;
k3:=Filtered(k,i->i mod (8*p) in Union([4*p+1..8*p-1],[0]));
for i in k3 do
 x:=(i-4*p) mod (8*p);
 if x mod 4 in [0,3] then N[i]:=[i-1,i+1,i+4*p];
  else N[i]:=[i-1,i+1,i-4*p]; fi;
 if x=1 then N[i]:=[i+1,i+4*p-1,i-4*p]; fi;
```

```
 if x=4*p then N[i]:=[i-1,i-4*p+1,i+4*p]; fi;
od;
k4:=[n-4*p+1..n];
 for i in k4 do
  x:=(i-4*p) mod (8*p);
    if q mod 2=0 then
      if x mod 4 in [0,3] then N[i]:=[i-1,i+1];
        else N[i]:=[i-1,i+1,i-4*p]; fi;
       if x=1 then N[i]:=[i+1,i+4*p-1,i-4*p]; fi;
       if x=4*p then N[i]:=[i-1,i-4*p+1]; fi;
     else
  x:=i mod (8*p);
       if x mod 4 in [0,3] then N[i]:=[i-1,i+1,i-4*p];
         else  N[i]:=[i-1,i+1];fi;
       if x=1 then N[i]:=[i+1,i+4*p-1];fi;
       if x=4*p then N[i]:=[i-1,i-4*p,i-4*p+1]; fi;
     fi;
  od;

w:=0;
Sc:=0;
v:=[];
D:=[];
for i in [1..n] do
 D[i]:=[];
 u:=[i];
 D[i][1]:=N[i];
 v[i]:=Size(N[i]);
 u:=Union(u,D[i][1]);
 w:=w+Size(D[i][1]);
 Sc:=Sc+v[i]*2*Size(D[i][1]);
 r:=1;
 t:=1;
 while r<>0 do
  D[i][t+1]:=[];
  for j in D[i][t] do
   for m in Difference (N[j],u) do
    AddSet(D[i][t+1],m);
   od;
  od;
  u:=Union(u,D[i][t+1]);
  w:=w+(t+1)*Size(D[i][t+1]);
  Sc:=Sc+v[i]*(t+1)*Size(D[i][t+1]);
  if D[i][t+1]=[] then r:=0;fi;
  t:=t+1;
  od;
 od;
w:=w/2; # This value is equal to the Wiener index of the graph
Sc; # This value is equal to the Schultz index of the graph
```

Examples for various p and q, are given in Table 1.

Table 1. Wiener and Schultz indices of $TUC_4C_8(S)$ nanotube for some p and q

p	q	W(G)	MTI(G)
2	2	336	1768
3	2	1032	5292
3	3	2664	14328
4	2	2336	11856
4	3	5824	31200
5	4	20800	114220
6	3	18144	96912
7	4	52864	290276
7	7	189336	1076488
3	5	9456	52704
4	6	30720	172944
4	8	64384	367280
5	7	79200	449560
6	8	175056	1000200
7	8	260288	1488340
7	9	346528	1990352
8	8	369664	2115168
6	10	306480	1765560

3. An algorithm for computing the Szeged and PI indices

In this section, we give an algorithm that enables one to compute the Szeged and PI indices of any graph. The algorithm is as follows:

1- Assign to any vertex one index number.

2- Determine all adjacent vertices of the vertex i, $i \in V$ and denote this set by N(i). The set of vertices that their distance to vertex i is equal to (

= be an edge connecting the vertices i and j, then we have the following results:

a) $V = \bigcup_{t \geq 0} D_{i,t}$, $i \in V$ (G).

b) $(D_{i,t} \setminus D_{j,t}) \subseteq (D_{j,t-1} \bigcup D_{j,t+1})$, $t \geq 1$.

c) $(D_{i,t} \bigcap D_{j,t-1}) \subseteq N_2(e|G)$ *and* $D_{i,t} \bigcap D_{j,t+1} \subseteq N_1(e|G)$ $t \geq 1$.

d) $(D_{i,1} \bigcup \{i\}) \setminus (D_{j,1} \bigcup \{j\}) \subseteq N_1(e|G)$ *and* $(D_{j,1} \bigcup \{j\}) \setminus (D_{i,1} \bigcup \{i\}) \subseteq N_2(e|G)$.

According to the above relations, by determining $D_{i,t}$, $t \geq 1$, we can obtain $N_1(e|G)$ and $N_2(e|G)$ for each edge e and therefore the Szeged and PI indices of the graph G is computed.

3- The distance between vertex i and its adjacent vertices is equal to 1, therefore $D_{i,1} = N(i)$. For each $j \in D_{i,t}$, $t \geq 1$, the distance between each vertex of set $N(j) \setminus (D_{i,t} \bigcup D_{i,t-1})$ and the vertex i is equal to $t+1$, thus we have

$$D_{i,t+1} = \bigcup_{j \in D_{i,t}} (N(j) \setminus (D_{i,t} \bigcup D_{i,t-1})) , \ t \geq 1.$$

According to the above equation, $D_{i,t}$ $t \geq 2$ for each $i \in V$ (G) can be obtained.

4- At the start of program, set SZ and PI equal to zero and T equal to empty set. At the end of program, the values SZ and PI are equal to the Szeged and PI indices of the graph G, respectively. For each vertex i, $1 \leq i \leq n$, and each vertex j in $N(i)$, calculate $N_1(e|G)$ and $N_2(e|G)$ for edge $e = ij$, then add the values of $n_1(e|G).n_2(e|G)$ and $n_1(e|G) + n_2(e|G)$ to SZ and PI, respectively. Since the edge ji is equal to ij,

the vertex i is added to T and continue this step for the vertex i+1 and for each vertex in $N(i+1)\setminus T$. According to the above algorithm, a GAP program to compute the Szeged and PI indices of a $HAC_5C_7[p,q]$ nanotube was provided.

Example 3.1. A C_5C_7 net is a trivalent decoration made by alternating C_5 and C_7. It can cover either a cylinder or a torus. In this example, the Szeged and PI indices of $HAC_5C_7[p,q]$ nanotube is computed by the GAP program.

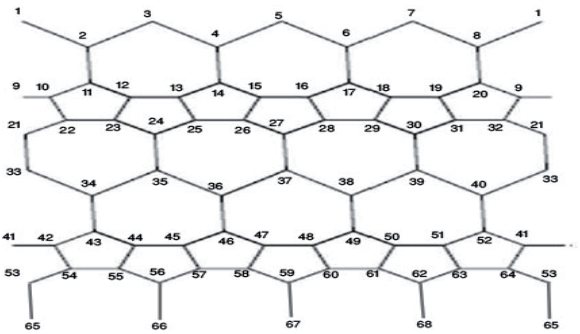

Figure2. $HAC_5C_7[4, 2]$

Let's denote the number of heptagons in one row by p. In this nanotube, the three first rows of vertices and edges are repeated alternatively, and the number of this repetition is denoted by q. In each period there are $8p$ vertices and p vertices, which are joined to the end of the graph and hence the number of vertices in this nanotube is equal to $8pq + p$.

Let's partition the vertices of this graph to following sets:

K_1: The vertices of the first row, which number is $2p$.

K_2: The vertices of the first row in each period (except the first one) which number is $2p(q-1)$.

K_3: The vertices of the second rows in each period, which number is $3pq$.

K_4 : The vertices of the third row in each period, which number is $3pq$.

K_5 : The last vertices of the graph, which number is p .

Figure 3 shows the rows of m[th] period.

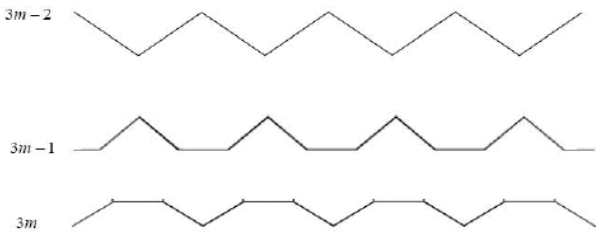

Figure3

A program to obtain the adjacent vertices set to each vertex in the sets K_i, i=1…5 was written. The set of adjacent vertices to each vertex is obtained by joining these programs. The values of x denote the index number of vertex i in a given period.

The following program computes the Szeged and PI indices of a $HAC_5C_7[p,q]$ nanotube, for arbitrary p and q .

```
p:=3; q:=7;# (For example)
n:=8*p*q +p;
N:=[];
K1:=[1..2*p];
V1:=[2..2*p-1];
N[1]:=[2,2*p];
N[2*p]:=[2*p-1,5*p,1];
for i in V1 do
 if i mod 2=0 then N[i]:=[i-1,i+1,3/2 *i+2*p];
  else N[i]:=[i-1,i+1];fi;
od;
k:=[2*p+1..8*p*q];
k2:=Filtered(k,i->i mod (8*p)in [1..2*p]);;
for i in k2 do
 x:= i mod (8*p);
 if x mod 2 =1 then N[i]:=[i-1,i+1,(x-1)*(3/2) +1+i-x-3*p];
  else N[i]:=[i-1,i+1,x*(3/2) +2*p+i-x];fi;
 if x=1 then N[i]:=[i+1,i-1+2*p,i-3*p];fi;
 if x=2*p then N[i]:=[i-1,i+3*p,i-2*p+1];fi;
od;

k3:=Filtered(k,i->i mod (8*p) in[2*p+1..5*p]);;
for i in k3 do
 x:=i mod (8*p);
```

```
 if (x-(2*p)) mod 3 =1 then N[i]:=[i-1,i+1,i+3*p-1];
  elif (x-(2*p)) mod 3 =2 then N[i]:=[i-1,i+1,i+3*p];
   elif (x-(2*p)) mod 3 =0 then N[i]:=[i-1,i+1,(2/3) *(x-2*p)+i-x];fi;
 if x=2*p+1 then N[i]:=[i-1+3*p,i-1+6*p,i+1];fi;
 if x=5*p   then N[i]:=[i-3*p,i-3*p+1,i-1];fi;
od;

k4:=Filtered(k,i->i mod (8*p) in Union([5*p+1..8*p-1],[0]) );;
for i in k4 do
 x:=i mod (8*p);
 if (x-(5*p)) mod 3 =1 then N[i]:=[i-1,i+1,(x-(5*p)-1)*(2/3) +1+(i-x)+8*p];
  elif (x-(5*p)) mod 3 =2 then N[i]:=[i-1,i+1,i-3*p];
   elif (x-(5*p)) mod 3 =0 then N[i]:=[i-1,i+1,i-3*p+1];fi;
 if x=5*p+1 then N[i]:=[i+3*p-1,i+1,i+3*p];fi;
 if x=0 then N[i]:=[i-1,i-3*p+1,i-6*p+1];fi;
od;
K5:=[8*p*q+1 ..8*p*q+p];

for i in K5 do
 x:=i-8*p*q;
 y:=8*p*(q-1)+5*p+3*x-2;
 N[i]:=[y];
 N[y][3]:=i;
od;
D:=[];
for i in [1..n] do
   D[i]:=[];
   u:=[i];
   D[i][1]:=N[i];
   u:=Union(u,D[i][1]);
   s:=1;
   t:=1;
   while s<>0 do
     D[i][t+1]:=[];
     for j in D[i][t] do
       for m in Difference(N[j],u) do
         AddSet(D[i][t+1],m);
       od;
     od;
   u:=Union(u,D[i][t+1]);
      if D[i][t+1]=[] then
     s:=0;
   fi;
   t:=t+1;
   od;
od;
A:=[];
T:=[];
sz:=0;
pi:=0;
e:=[];
for i in [1..n-1] do
N1:=[];
  for j in Difference(N[i],T) do
N2:=[];
```

```
   N1[j]:=Union(Difference(N[i],Union([j],N[j])),[i]);
   N2[i]:=Union(Difference(N[j],Union([i],N[i])),[j]);
     for t in [2..Size(D[i])-1] do
       for x in Difference(D[i][t],Union(D[j][t],[j])) do
         if not x in D[j][t-1] then
            AddSet(N1[j],x);
          elif x  in D[j][t-1] then
           AddSet(N2[i],x);
        fi;
       od;
     od;
   sz:=sz+Size(N1[j])*Size(N2[i]);
   pi:=pi+Size(N1[j])+Size(N2[i]);
  od;
  Add(T,i);
od;
sz;# The value of sz is equal to szeged index of the graph
pi;# The value of pi index of the graph.
```

References

1. Wiener, H. Structural determination of paraffin boiling points, *J. Am. Chem. Soc.* **69** (1947) 17-20.
2. Khadikar, P. V.; Karmarkar, S. On the Estimation of PI index of Polyacenes. *Acta Chim. Slov*. **49** (2002) 755-771.
3. Nikolić, S.; Trinajstić, N.; Mihalić, Z. The Wiener index: developments and applications. *Croat. Chem. Acta* **68** (1995) 105–129.
4. Gutman, I.; Yeh, Y. N.; Lee, S. L.; Luo, Y. L.: Some recent results in the theory of the Wiener number, *Indian J. Chem.* **32A** (1993) 651–661.
5. Gutman, I.; Potgieter, J. H. Wiener index and intermolecular forces. *J. Serb. Chem. Soc.* **62** (1997) 185–192.
6. Schultz, H. P. Topological organic chemistry. 1. Graph theory and topological indices of alkanes, *J. Chem. Inf. Comput. Sci.* **29** (1989) 227-228.
7. Schultz, H. P.; Topological organic chemistry.13.Transformation of graph adjacency matrixes to distance matrixes, *J. Chem. Inf. Comput. Sci.* **40** (2000) 1158-1159.
8. Klavžar, S.; Gutman, I.; Wiener number of vertex-weighted graphs and a. chemical applications, *Disc. Appl. Math.* **80** (1997) 73-81.
9. Gutman, I.; Selected properties of the Schultz molecular topological index, *J. Chem. Inf. Comput. Sci.* **34** (1994) 1087-1089.
10. Dobrynin, A. A.; Explicit Relation between the Wiener Index and the Schultz Index of Catacondensed Benzenoid Graphs, *Croat. Chem. Acta* **72** (1999) 869-874.
11. Deng, H.; The trees on $n \geq 9$ vertices with the first to seventeenth greatest Wiener indices are chemical trees, *MATCH Commun. Math. Comput. Chem.* **57** (2007) 393-402.
12. Diudea, M. V.; Stefu, M.; Pary, B.; John, P. E.; Wiener index of armchair polyhex nanotubes, *Croat. Chem. Acta* 77 (2004) 111–115.
13. John, P. E.; Diudea, M. V.; Wiener index of Zig-zag polyhex nanotubes, *Croat. Chem. Acta* **77** (2004) 127-132.
14. Randić, M.; On generalization of Wiener index for cyclic structures, *Acta Chim. Slov.* **49** (2002) 483-496.

15. Stefu, M.; Diudea, M. V.; Wiener Index of C4C8 Nanotubes, *MATCH Commun. Math. Comput. Chem.* **50** (2004) 133-144.
16. Yousefi, S.; Ashrafi, A. R.; An exact expression for the Wiener index of a polyhex nanotorus, *MATCH Commun. Math. Comput. Chem.* **56** (2006) 169-178.
17. Zhang, H.; Xu, S.; Yang, Y.; Wiener index of toroidal polyhexes, *MATCH Commun. Math. Comput. Chem.* **56** (2006) 153-168.
18. Khadikar, P.V. On a novel structural descriptor, *PL. Nat. Acad. Sci. Lett* **23** (2000) 113-118.
19. Khadikar, P.V; Karmarkar, S; Agrawal,V. K.; Relationships and relative correlation potential of the Wiener, Szeged and PI indices, *Nat. Acad. Sci. Lett.* **23** (2000) 165-170.
20. Khadikar, P.V.; Karmarkar, S.; Agrawal, V. K. A novel PI index and its applications to QSPR/QSAR studies, *J. Chem. Inf. Comput. Sci*. **41** (2001) 934–949.
21. Khadikar, P. P.; Deshpande, N. V.; Kale, P. P.; Dobrynin, A. A.; Gutman, I.; Domotor, G.; The Szeged index and an analogy with the Wiener index, *J. Chem. Inf. Comput. Sci* **35** (1995) 545-550.
22. Mathur, K. C.; Singh, S.; Mathur S.; Khadikar, P. V.; Modeling polychlorinated biphenyls (PBCs) in environment, *Poll. Res*. **18** (1999) 405-409.
23. Iranmanesh, A.; Khormali, O.; PI index of HAC_5C_7 nanotube, *J. Comput. Theor. Nanosci.* (In press)
24. Ashrafi, A. R.; Saati, H.; PI and Szeged indices of one−pentagonal carbon nanocones. *J. Comput. Theor. Nanosci.*, **4** (2007) 761-763.
25. Iranmanesh, A.; Khormali, O.; Szeged index of HAC_5C_7 nanotubes*, J. Comput. Theor. Nanosci.* (In press)
26. Iranmanesh, A.; Soleimani, B.; PI Index of $TUC_4C_8(R)$ nanotubes, *MATCH Commun. Math. Comput. Chem.* **57** (2007) 251-262.
27. Iranmanesh, A.; Soleimani, B.; Ahmadi, A.; Szeged index of $TUC_4C_8(R)$ nanotube, *J. Comput. Theor. Nanosci*. **4** (2004) 147-151.
28. Ashrafi A. R.; Loghman, A.; PI index of zig-zag polyhex nanotubes, *MATCH Commun. Math. Comput. Chem.* **55** (2006) 447-452.
29. Ashrafi A. R.; Loghman, A.; Padmakar-Ivan index of TUC4C8(S) nanotubes, *J. Comput. Theor. Nanosci.* **3** (2006) 378-381.
30. Ashrafi A. R.; Rezaei, F.; PI index of polyhex nanotori, *MATCH Commun. Math. Comput. Chem.* **57** (2006) 243-250.
31. Ashrafi A. R.; Loghman, A.; PI index of armchair polyhex nanotubes, *Ars Combin.* **80** (2006) 193-196.
32. Yousefi, H.; Bahrami, A.; Yazdani, J.; Ashrafi, A. R.; PI index of V-phenylenic nanotubes and nanotori, *J. Comput. Theor. Nanosci.* **3** (2007) 604-605.
33. Schonert, M. et al., *GAP, Groups, Algorithms and Programming*, Lehrstuhl De fur Mathematik, RWTH, Achen, **1992.**
34. Dabirian, M.; Iranmanesh, A.; The full non-rigid group theory for trimethylamine-BH3 complex, *MATCH Commun. Math. Comput. Chem.* **54** (2005) 75-88.
35. Trinajstić, N.; *Chemical Group Theory*, CRC Press, Boca Roton, FL. **1992**.
36. Ashrafi A. R.; Yousefi S.; Computing the Wiener index of a TUC4C8(S) nanotorus, *MATCH Commun. Math. Comput. Chem.* **57** (2007) 403–410.