International Journal of Applied Mathematics
& Statistical Sciences (IJAMSS)
ISSN(P): 2319-3972; ISSN(E): 2319-3980
Vol. 2, Issue 5, Nov 2013, 9-18
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

VERTEX AND EDGE HYPER WIENER INDEX OF SOME NANOTUBES USING NEW DISTANCE MATRIX ALGORITHM

K. THILAKAM & R. BHUVANESWARI

PG and Research, Department of Mathematics, Seethalakshmi Ramaswami College, Thiruchirappali, Tamil Nadu, India

ABSTRACT

One of the most widely known topological descriptors is the Wiener index or (Wiener number) named after American chemist Harold Wiener in 1947. Wiener number of a connected graph G is defined as the sum of the distances between distinct pairs of vertices of G.. The Hyper Wiener index is defined as WW (G) = $(\sum d^2(u\ ,v)+\sum d(u,v))/2$, where d(u,v) denotes the distance between the vertices u and v in the graph G and the summations run over all distinct pairs of vertices of G. Recently an edge version of Hyper Wiener Index was introduced by Ali Iranmanesh. Diudea was the first chemist who considered the problem of computing topological indices of nanostructures. In this paper we introduced a new algorithm to calculate the distance matrix of the given connected graph and using this we calculated edge and hyper Wiener index of some nanotubes.

KEYWORDS: Distance Sum, Nanotubes, Vertex and Edge Wiener Index, Vertex and Edge Hyper Wiener Index

INTRODUCTION

A Graph G is formally defined to be a pair [V(G,E(G)] where V(G) is a non empty finite set of elements called vertices and E(G) is a finite set of unordered pairs of elements of V(G) called edges. Molecular graphs represent the constitution of molecules[**1**]. They are generated using the following rule: Vertices stand for atoms and edges for bonds. A graph theoretical distance d (u,v) between the vertices u and v of the graph G is equal to the length of the shortest path that connects u and v.

An invariant of a graph G is a number associated with G that has the same value for any graph isomorphic to G. If G is a molecular graph then the corresponding invariants are called molecular descriptor or topological indices and they are used in theoretical chemistry for the design of so called Quantitative Structure Property Relations (QSPR) and Quantitative Structure Activity Relations (QSAR). One of the oldest topological index is Wiener index and is defined as the half of the sum of all the distances between every pair of vertices of G.[**2**] ie

$$\mathrm{W(G)} = \frac{1}{2}\sum_{u,v} d(u,v) \tag{1}$$

The hyper wiener index was introduced by Randic [**3**] and his definition is applicable to trees only. Klein, Lukovits and Gutman[**4**] introduced the formula for both trees and cycle containing structures and Hyper Wiener index WW of a graph G is defined as

$$WW(G) = \frac{1}{2}\Big(\sum d(u,v) + \sum d(u,v)^2\Big) = \frac{1}{2}(W(G) + \sum d(u,v)^2) \tag{2}$$

and summation runs over all distinct pairs of vertices. Equations (1) and (2) are called vertex version of Wiener and Hyper Wiener index respectively.

The edge versions of Wiener Index were introduced by A.Iranmanesh, I.Gutman, O.Khormali,A.Mahmiani in 2009 [5] and is defined as follows

The first edge- Wiener number is

$$W_{e0}(G) = \frac{1}{2} \sum_{\substack{e,f \in E(G) \\ e \neq f}} d_0(e,f) \tag{3}$$

Where $d_0(e,f) = d_1(e,f) + 1$ $e \neq f$

$= 0$ $e=f$

$d_1(e,f) = \min\{d(x,u), d(x,v), d(y,u), d(y,v)\}$ such that e=xy and f = uv and $W_{e0} = W(L(G))$.

The second edge - Wiener number is

$$W_{e4}(G) = \frac{1}{2} \sum_{\substack{e,f \in E(G) \\ e \neq f}} d_4(e,f) \tag{4}$$

Where $d_4(e,f) = \{ d_2(e,f)$ $e \neq f$

$= 0$ $e=f$

$d_2(e,f) = \max\{d(x,u), d(x,v), d(y,u), d(y,v)\}$ such that e=xy and f = uv. The edge version of Hyper Wiener indices are defined as[6]

$$WW_{ei}(G) = W_{ei}(G) + W_{ei}^{d^2}(G) \tag{5}$$

$$W_{ei}^{d^2}(G) = \frac{1}{2} \sum_{\substack{e,f \in E(G) \\ e \neq f}} d_i^2(e,f) \ , \ i = 0,4 \tag{6}$$

w here d_i are distances for i=0,4 .Since all these topological indices are depends on distance between every pair of vertices of a given graph G. So that to calculate these indices we need to calculate the distance matrix D = (d(i,j)) for a given graph G and it is defined as follows:

- d(i,i) = 0
- d(i,j) = the length of the shortest path between vertices i and j

W. R. Muller [7] gave an algorithm for distance matrix for a graph G with n vertices using a new matrix multiplication(square) namely a(i,j)=min{a(i,k)+a(k,j)/k=1,2,….N} ,for all (i,j) where $A_0=a_0(i,j)$ is the adjacency matrix in which all non-diagonal zeros are replaced by n .We square matrix A_0 for $L > \log_2(N-1)$ times. We encourage the reader to consult this paper for background material as well as basic computational techniques. To overcome the complexity in this algorithm we come up with new algorithm to calculate the distance matrix for the given connected graph G with n vertices. In this paper we introduced a new algorithm to calculate the distance matrix for the given graph and using this algorithm, Vertex and Edge version of HyperWiener index of Armchair Polyhex Nanotubes, A Zig-Zag Nanotube, TUC4C8(S) nanotorus and TUC4C8(R) Nanotubes are calculated by using MATLAB [12].

NEW DISTANCE MATRIX ALGORITHM

Let G be a given connected graph with n vertices.

Input: n, number of adjacency pairs and adjacency list of connected graph G.

Output

Distance matric D = (d(i,j))

Step 1: Determine the adjacency matrix using the adjacency list and denote it by A = a(i,j).

Step 2: Replace every non diagonal zero in A by n denote the new matrix by D.

Step 3: For c=1and for all i,j = 1,2,…..,n if d(i,j)=c, and if d(j,k) = 1 define d(i,k) = min(d(i,k),c+1)

and d(k,i)=d(i,k) for all k=1,2,…..n.

Step 4: Increase c by 1 and repeat step 3 until c < n gives the distance matrix D.

Advantage in this algorithm is that it takes less execution time than the algorithm given by W.R.Muller [**7**]. Using this distance matrix as input and by all above six equations we can calculate W ,WW ,W_{e0} ,WW_{e0} W_{e4} , WW_{e4}. Its MATLAB Program is given below.

PROGRAM USING MATLAB

```
%PROGRAM TO CLALCULATE WIENER INDICES USING NEW DISTANCE MATRIX ALGORITHM

n=input ('NUMBER OF VERTICES') % ADJACENCY MATIRX FORMATION BEGINS

t=input ('NUMBER OF EDGES')

A=[];
for i=1:t
a=input('ONE END OF AN EDGE')

b=input('OTHER END OF AN EDGE')

  if a~=b
    A(a,b)=1;A(b,a)=1;
  else
    A(a,b)=0;    end
end
A %GIVES THE ADJACENCY MATIRX

D=[];D=A; %DISTANCE  MATIRX CALCULATION BEGINS

for i=1:n
  for j=1:n
    if i~=j&A(i,j)==0
       D(i,j)=n;
    end
  end
```

```
end
c=1;
while c<n
    for i=1:n
        for j=1:n
            if D(i,j)==c
                for k=1:n
                    if D(j,k)==1
                        D(i,k)=min(D(i,k),c+1);D(k,i)=D(i,k);
                    end
                end
            end
        end
    end
    c=c+1;
end
D%GIVES THE DISTANCE MATRIX
w=0;ww=0; %COMPUTATION OF WIENER INDICES BEGINS
for i=1:n
for j=1:n
    w=D(i,j)+w;
    ww=(D(i,j)^2)+ww;
end
end
w0=0;ww0=0;f=0;w4=0;ww4=0;count=0;
    for i=1:n
        for j=1:n
            if A(i,j)==1&i<j
                for k=1:n
                    for l=1:n
                        if A(k,l)==1&k<l
                            if (k==i&l==j)|(k==j&l==i)
                                w01=0;w41=0;
                                w0=w01+w0;
                                ww0=w01^2+ww0;
                                f=2*w01+f;
                                w4=w41+w4;
                                ww4=w41^2+ww4;
                            else
                                x1=min(D(i,k),D(i,l));
                                y1=min(D(j,k),D(j,l));
                                w01=min(x1,y1);
                                w0=w01+w0;
                                ww0=w01^2+ww0;
                                f=2*w01+f;
                                x2=max(D(i,k),D(i,l));
                                y2=max(D(j,k),D(j,l));
                                w41=max(x2,y2);
                                w4=w41+w4;ww4=w41^2+ww4;count=count+1;
                            end
                        end
                    end
                end
            end
        end
    end
W0=(w0+count)/2;

WW0=((ww0+count+f)/2)+W0;

fprintf('Wiener index W = %d \n' , w/2) %FINAL OUTPUT

fprintf('Hyper Wiener index WW = %d \n' , (ww+w)/4)

fprintf('First edge Wiener index We0 = %d \n' , W0)

fprintf('First edge Hyper Wienr index WWe0 = %d \n' , WW0)

fprintf('Second edge Wiener index We4 = %d \n' , w4/2)

fprintf('Second edge Hyper Wiener index WWe4 = %d \n' , (ww4+w4)/2)
```

REMARK

The only complexity in this new distance matrix algorithm is that for graph G with more number of vertices it is hard to give the input adjacency list. To overcome this complexity we can define the general form of adjacency matrix of some graphs. By replacing adjacency matrix formation part of above program by this general form of adjacency matrix we can calculate vertex and edge Hyper Wiener index of corresponding graphs. In the following sections we define general form of adjacency matrix of some nanotubes using MATLAB.

VERTEX AND EDGE HYPER WIENER INDEX OF SOME NANOTUBES

In 1991 Iijima [**8**] discovered Carbon nanotubes as multi walled structures. Carbon nanotubes show remarkable mechanical properties. Experimental studies have shown that they belong to the stiffest and elastic known materials. These mechanical characteristics clearly predestinate nanotubes for advanced composites. Diudea was the first chemist who considered the problem of computing topological indices of nanostructures. Our notations of all nanotubes are mainly taken from[**9-11**].

ARMCHAIR POLYHEX NANOTUBE (TUV C_6[2p; q])

In this section G: = TUV C_6[2p, q], denotes an arbitrary armchair polyhex nanotube in terms of their circumference 2p and their length q

General form of Adjacency matrix of an Armchair polyhex is explained in the following MATLAB program and is illustrated in figure1.

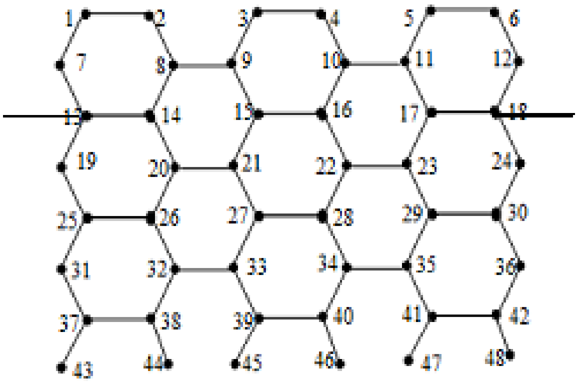

Figure 1: An Armchair Polyhex Lattice (p= 3, q=8)

```
%PROGRAM TO CALCULATE ADJACENCY MATRIX OF AIMCHAIR POLYHEX NANOTUBE

p=input('NUMBER OF ROWS')

q=input('NUMBER OF COLUMNS')

A=[];
j=1;n=2*p*q
for i=1:2*p*q-2*p
   A(i,i+2*p)=1;A(i+2*p,i)=1;
end
while j<=2*p*q-1
   A(j,j+1)=1;A(j+1,j)=1;
   if rem(j+1,2*p)==0|rem(j+2,2*p)==0
      if rem(j+2,4*p)==0
         A(j+2,j-2*p+3)=1;A(j-2*p+3,j+2)=1;
      end
      j=j+1;
      end
   j=j+2;
end
   A
```

A ZIG-ZAG NANOTUBE ($TUHC_6$)[2p,q]

General form of Adjacency matrix of Zig-Zag nano tube is explained in the following MATLAB program and is illustrated in figure 2.

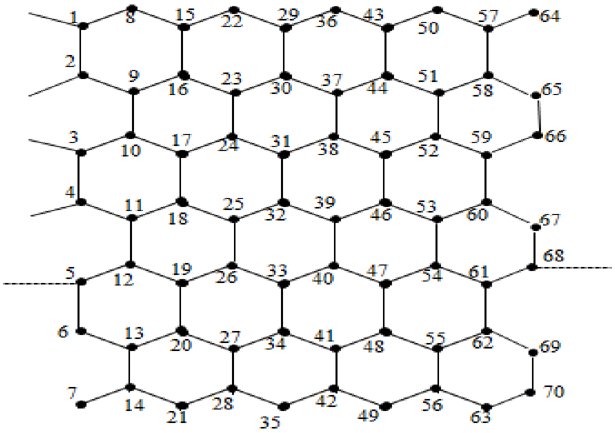

Figure 2: Zig-Zag Nano Tube (p = 5 q = 7)

TUC$_4$C$_8$ (R) NANOTUBE

General form of Adjacency matrix of T[p,q]=TUC$_4$C$_8$(R) nanotube is explained in the following program and is illustrated in figure 3.

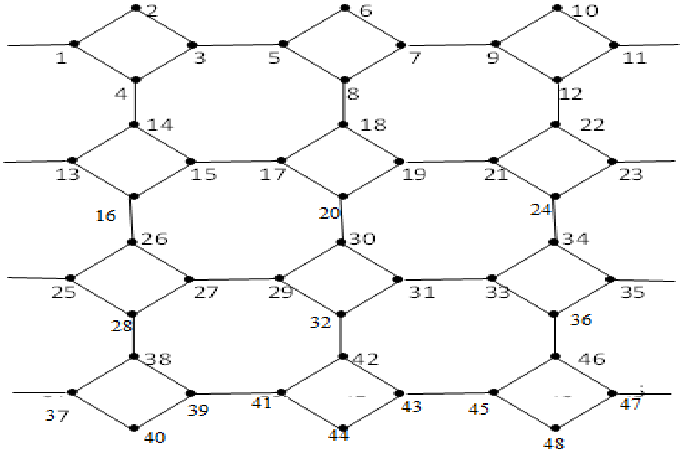

Figure 3: A TUC$_4$C$_8$(R) Lattice with p=3, q=4

```
%PROGRAM TO CALCULATE ADJACENCY MATRIX OF TUC4C8(R)

p=input('NUMBER OF ROWS')

q=input('NUMBER OF COLUMS')

A=[];
j=3;k1=4;
for i=1:4*p*q
  if rem(i,4)==0
      A(i,i-3)=1;A(i-3,i)=1;
      if i<=4*p*(q-1)
       A(i,4*p+i-2)=1;A(4*p+i-2,i)=1;
       end
      else
   A(i,i+1)=1;A(i+1,i)=1;
  end
```

```
    while j<4*p*q
        if j==k1*p-1
            A(j,j-4*p+2)=1;A(j-4*p+2,j)=1; k1=k1+4;
            else
            A(j,j+2)=1;A(j+2,j)=1;
       end
        j=j+4;
    end
end
```

TUC_4C_8(S) NANOTUBE

Adjacency matrix of T[p,q] = TUC_4C_8(R) nanotube is explained in MATLAB program.

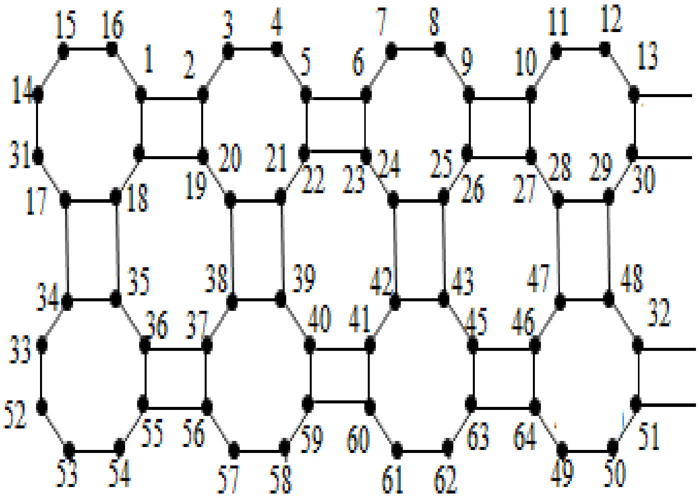

Figure 4: Fragment of TUC4C8(S) Nanotorus with p =4 q =4

```
%PROGRAM TO CALCULATE ADJACENCY MATRIX OF TUC4C8(S)
p=input('NUMBER OF ROWS')
q=input('NUMBER OF COLUMS')
A=[];n=4*p*q
for i=1:(4*p*q)
    if rem(i,(4*p))==0
        A(i,i-(4*p)+1)=1;A(i-(4*p)+1,i)=1;
        else
        A(i,i+1)=1;A(i+1,i)=1;
        end
if  (rem(i,4)==1|rem(i,4)==2)&i<4*(q-1)*p
    A(i,i+4*p+2)=1;A(i+4*p+2,i)=1;
    end
end
A
```

Illustration

Table 1: Vertex and Edge Hyper Wiener Index of Armchair Polyhex Nanotubes and Zig-Zag Nano Tubes

Nano Tubes	Wiener Indicies	p	3	6	5	7	9	10
		q	16	12	5	7	4	8
$TUVC_6$	W		32136	81216	6685	36183	23616	134400
	WW		160776	416592	25515	185892	152640	951840
	W_{e0}		64926	160500	11250	66255	36882	253100
	WW_{e0}		628776	1615080	85650	678734	476802	3581960
	W_{e4}		73650	179784	13140	74424	40698	276000
	WW_{e4}		766560	1947888	109070	815276	553482	4101600
$TUHC_6$	W		49836	91620	5625	30527	15264	104320
	WW		419370	580410	18125	131712	62100	555520
	W_{e0}		105657	191451	10855	61369	28656	213005
	WW_{e0}		1738896	2361936	67940	517062	230544	2231320
	W_{e4}		115086	212079	13005	70336	33147	237825
	WW_{e4}		1965570	2767152	90980	646380	291654	2678840

Table 2: Vertex and Edge Hyper Wiener Index of TUC_4C_8(R) and TUC_4C_8(S) for Same Set of p, q Values

TUC_4C_8(R)	W	308550	577908	34345	186165	88848	625100
	WW	3895932	5680158	161590	1193353	497124	4851770
	W_{e0}	675846	1252506	71425	395304	182682	1337550
	WW_{e0}	16909512	24291978	658590	4989572	2031480	20518580
	W_{e4}	710571	1339374	80465	432404	202077	1442730
	WW_{e4}	18150618	26897454	797160	5766810	2390238	23269760
TUC_4C_8(S)	W	226992	490896	35000	189336	107712	673280
	WW	2021688	3787152	163330	1200402	751176	5625760
	W_{e0}	482514	1029642	67740	382130	202644	1379970
	WW_{e0}	8432952	15593076	622460	4785284	2814642	22883800
	W_{e4}	520026	1114098	76840	419370	221184	1481810
	WW_{e4}	9426444	17738796	765620	5583424	3234078	25739800

CONCLUSIONS

Hence using Distance matrix algorithm we can calculate Wiener indices for any connected simple graph G(V,E) and by defining the general form the adjacency matrix of some nanotubes we obtained the vertex and edge version of Wiener and Hyper Wiener index of TUVC6 , TUHC6 , TUC_4C_8(R) , TUC_4C_8(S).

REFERENCES

1. Trinajsti N., "*Chemical Graph Theory*", CRC Press, Boca Raton, FL, 1983; 2nd revised edition, 1992.
2. Wiener. H., "*Structural determination of parafin boiling"*, J. Am. Chem. Soc. 69, (1947) 17-20.
3. Randic, M. Novel molecular descriptor for structure property studies. Chem. Phys. Lett.1993, 211, 478-483
4. Klein, D. J.; Lukovits, I.; Gutman, I. On the definition of the hyper –Wiener index for cycle – containing structures. J. Chem. Inf. Comput. Sci. 1995, 35, 50-52.
5. Ali Iranmanesh, Gutman. I, Omid Khormali, Mahmian. A, "*The edge version of Wiener Index*", MATCH Commun. Math. Comput. Chem., 61 (2009), 663 – 672.
6. Ali Iranmanexh, Soltani Kafrani. A., Omid Khormali, "*A New Version of Hyper – Wiener Index*", MATCH Commun. Math. Comput. Chem., 65 (2011), 113 – 122.

7. Muller. W. R, Szymanski. K., and Knop. J. V, "*An Algorithm for construction of the Molecular Distance Matrix*", Journal of Computational Chemistry, 8, No.2, (1987), 170-173.

8. S. Iijima, "*Helical microtubules of graphitic carbon, Nature"*, 354 (1991) 56-58.

9. Mircea V. Diudea, Monica Stefu, Basil Pârv and Peter E. Johnc *"Wiener Index of Armchair Polyhex Nanotubes",* CROATICA CHEMICA ACTA, 77(2004),111 -115.

10. S. Yousefi and A. R. Ashrafi, "*An Exact Expression for the Wiener Index of a TUC4C8(R) Nanotorus"*, J. Math. Chem. 1031-1039.

11. Ali Reza Ashrafi and Shahram Yousefi, "*Computing the Wiener Inde x of a TUC4C8(S) Nanotorus"*, MATCH Commun. Math. Comput. Chem. **57** (2007) 403-410.

12. Amos Gilat, "*MATLAB An Introduction with Applications*", John Wiley & Sons, Inc. U.K, 2004.