Iranian Journal of Mathematical Chemistry, Vol. 3, No. 1, February 2012, pp. 81 – 94 **IJMC**

Automatic Graph Construction of Periodic Open Tubulene ((5,6,7)3) and Computation of its Wiener, PI, and Szeged Indices

AHMAD YOOSOFAN• AND MOSTAFA NAMAZI–FARD

Department of Computer Engineering, Faculty of Engineering, University of Kashan, Kashan, Iran

(*Received March 13, 2011*)

ABSTRACT

The mathematical properties of nano molecules are an interesting branch of nanoscience for researches nowadays. The periodic open single wall tubulene is one of the nano molecules which is built up from two caps and a distancing nanotube/neck. We discuss how to automatically construct the graph of this molecule and plot the graph by spring layout algorithm in graphviz and netwrokx packages. The similarity between the shape of this molecule and the plotted graph is a consequence of our work. Furthermore, the Wiener, Szeged and PI indices of this molecule are computed.

Keywords: Open tubulene, topological index, Szeged index, Wiener index, PI index.

1 INTRODUCTION

Carbon nanotubes exhibit a large number of new interesting phenomena, therefore many researches of different areas attracted to work on nanotubes [1]. They are crucial in all sorts of ways because of the manifold utilities they provide. One interesting feature of carbon nanotubes is their use as catalyst for improving the hydrogen absorption and desorption [2]. Some researchers are trying to use single-wall nanotubes as reservoir for storing hydrogen which may use as fuel by penetrating more Hydrogen atoms in the structure of the molecules[3,4]. One major element of energy research activities of some countries is reducing or eliminating the dependency on petroleum of transportation systems by replacing it with new fuels. Hydrogen fuel have the potential to offer cleaner, more efficient alternatives to today's technology [5]. Therefore, many fuel molecules, including

•Corresponding Author (Tel/fax +98 361 591 2495 E–mail: yoosofan@kashanu.ac.ir)

nanotubes, with different features have been found and studied up to now. Tubulenes are types of nanotubes molecules which are studied as fuel, too [3,6,7].

Some physiochemical properties of these molecules depend on their structures. Simple indirect graph is used to model the structure of these molecules. Despite its simplicity and lack of some structural characteristics of a molecule, graph of a molecule comprises important topological information. Numerical values calculated based on a molecular graph are called topological indices. Several different topological indices with various applications have been proposed up to now.

In this article, the graph of open periodic single-wall tubulene is generated by an iterative method. The graph is plotted by spring layout algorithm [8]. In addition, some topological indices of the graph of this molecule are computed.

2 AUTOMATIC GRAPH CONSTRUCTION OF PERIODIC OPEN TUBULENE

Periodic open single wall tubulene is one of the nano molecules which is built up from two caps and a distancing nanotube/neck [9]. Periodic closed tubulene is derived from C_{60} by cutting off polar hexagons with the repeat spherical moiety [9]. The figure of periodic closed tubulene $C_{204(6(56)^3(665)^3(656)^3 7^6 - Z[12,0] - r)}$; r=4, is depicted in [10]. The periodic open tubulene is open counterpart of closed tubulene which is focused in this article. The precise figure of open periodic tubulene(((5,6,7)3)VA) is plotted in [10]. The Figure 1 shows a close plot of this molecule which is one of the consequences of this paper.

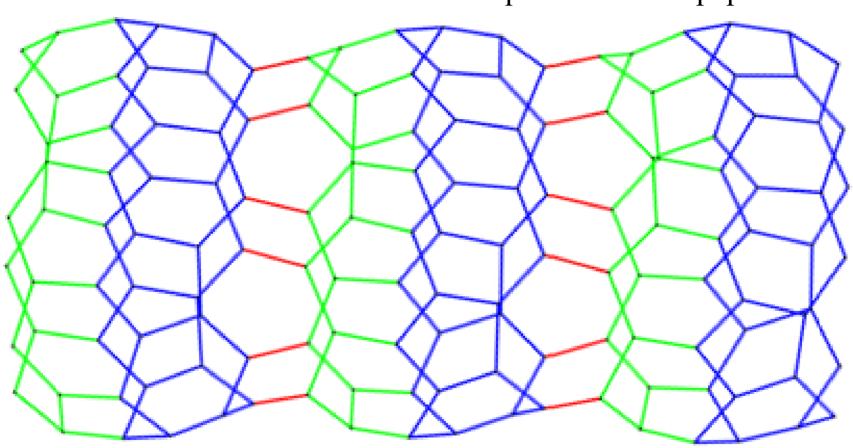

Figure 1. Periodic Open Tubulene.

It is difficult to construct the graph of periodic open tubulene based on its physical characterizations. Researchers who want to calculate topological indices of these sort of molecules often try to find the mathematical relations based on the graph of them [11–13]. It seems that finding the adjacency matrix of a nanotube is more useful than finding only one or two indices of it. One of the most interesting advantages of constructing graph of a molecule is the straightforward calculation of the various topological indices. A novel algorithm for constructing the graph of carbon nanohorn is proposed in [14]. In this article, an analogous approach has been proposed for constructing the graph of periodic open tubulene which is discussed in detail in the following paragraphs.

The repetition of three connected sections of pentagons, hexagons and heptagons constructs the graph of the open tubulene. These sections are shown in the Figure 1 by blue, green and red colors, respectively. The first and the second sections connect to each other to construct levels of this molecule. The third section uses for conjunction of the current level to the previous level, if any exist. Based on this definition of levels, the Figure 1 has three levels, three first sections, three second sections and two third sections. Each section rolls different number of pentagons, hexagons and heptagons.

The first section consists of six hexagons and three pentagons, Figure 2. Each consecutive pair of hexagons interleaves by a pentagon. The Figure 3 has more details on construction of the first section of the first level. It shows the node number of this section and how a node connects to other nodes, too. The construction of the first sections of the subsequent levels is similar to the first section of the first level.

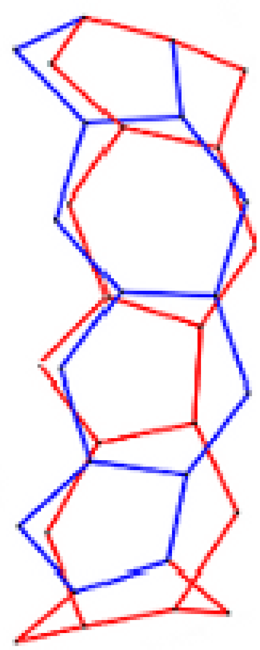

Figure 2. The First Section.

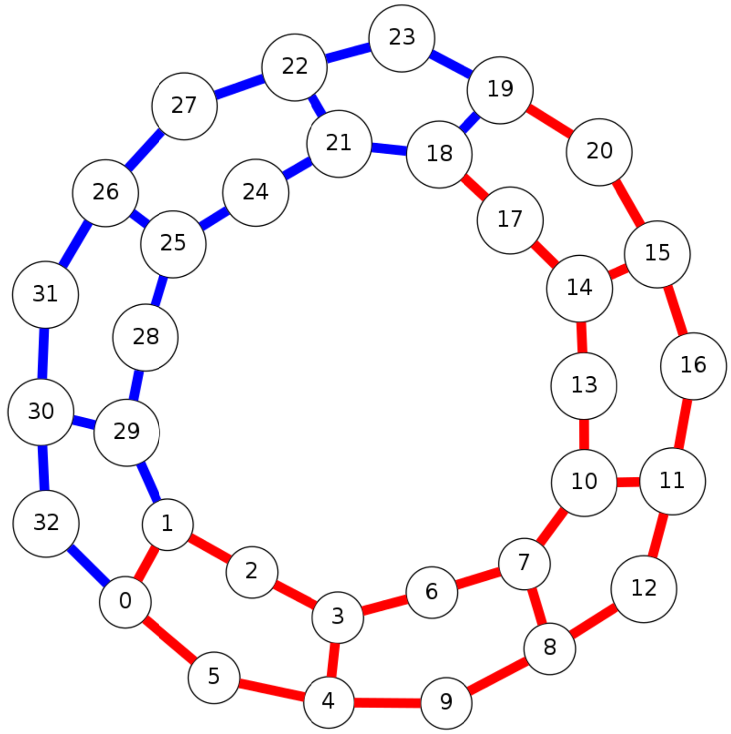

Figure 3. The Section One.

The second section also consists of six hexagons and three pentagons with different arrangement in respect to the first section. The second section is connected to the first one which is shown in the Figure 4.

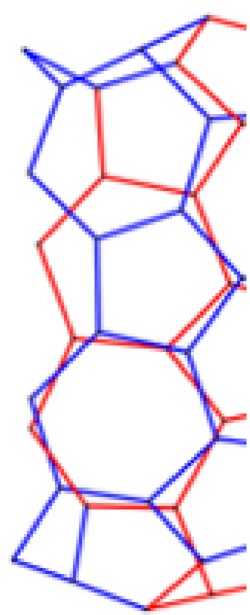

Figure 4. The Second Section.

The third section is too simple. It consists of six connected heptagons. The Figure 5 shows the third section.

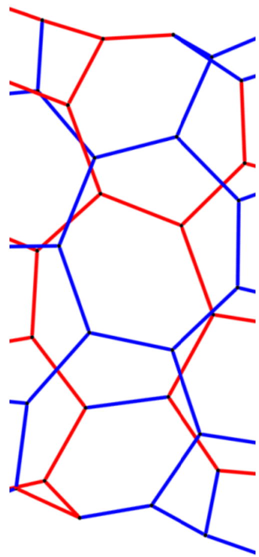

Figure 5. The Third Section.

Suppose NL denotes the number of levels has been created up to now, and ML denotes the maximum number of levels which we need to construct the graph. The algorithm of constructing open periodic tubulene has the following steps:

1. Construction of the first section of the first level
2. Construction of the second section of the first level
3. NL = 1
4. WHILE NL < ML DO
    4.1. Construction of the first section of the $(NL+1)^{th}$ level
    4.2. Construction of the third section of the $(NL+1)^{th}$ level
    4.3. Construction of second section of the $(NL+1)^{th}$ level
    4.4. NL = NL + 1
5. END

The Python programming language is used to implement the algorithms discussed in this paper. Python is a powerful open source and free scripting language enriched by many open source modules for wide variety of purposes. It has a very concise, clear, readable and consistent syntax, yet it has a lot of capabilities and advanced features such as dynamic typing, generators, exceptions, very high-level dynamic data types and classes. Several open source and free libraries are developed for working with graphs in Python, such as python–graph, NetworkX, py_graph, graph–tool, igraph, etc. The NetworkX is used for creating and manipulating graph objects in this paper. Many types of graphs, including simple graphs, directed graphs, and graphs with parallel edges and self-loops are implemented in NetworkX [15]. The following code is the python implementation of the preceding algorithm for creating open tubulene. The sections and levels are connected together by pre, nextin, nextst. Variable nv contains the number of nodes created up to now.

```
Pre=[0,0,0,0,0,0]
nextin=[0,0,0,0,0,0,0,0,0]
nextst=[0,0,0,0,0,0]
nv = 0
nv=FirstSection(nv,pre,nextin)
nv=SecondSection(nv,nextin,nextst)
for i in range(0,numberOfLevels-1):
   nv=FirstSection(nv,pre,nextin)
   ThirdSection(nextst,pre)
   nv=SecondSection(nv,nextin,nextst)
```

3 DRAWING THE GRAPH

During development of the preceding algorithm, plotting the resulting graph as shown in Figures 4 and 5 was used to adjust the algorithm. Therefore, several free and open source graph drawing tools were tried out. Neato was selected for graph drawing because the plotted graph by Neato is close to the shape of the molecule. Neato is a part of Graphviz package that make layouts of undirected graphs. Graphviz is free and open source graph visualization which is widely used in many areas. The tools in Graphviz take description of a graph in a simple text language, and create diagrams in different formats, such as images and SVG for web pages, PDF or Postscript for inclusion in other documents. Many useful features for concrete diagrams have been added to Graphviz, such as options for colors, fonts, tabular node layouts, line styles, hyperlinks, roll, and custom shapes.

Graphviz offers both graphical and command line tools. There exist several ways for using Graphviz from python, but in most cases the Graphviz command line tools are called to parse files containing a graph definition and render a rasterized image of the graph. Therefore, Neato can be either run in command line, or invoking it in python by “os.system” function. This is unsatisfactory for our purposes, and a more direct interface to the layout algorithms is desirable. There are several python interface libraries to the Graphviz (e.g. PyGraphviz, pydot, etc.). PyGraphviz have been chosen for this purpose in this research.

Neato draws undirected graph using a variation of spring algorithm proposed by Kamada and Kawai [8]. The proposed algorithm places an ideal spring between every pair of nodes such that its length is set to the shortest path distance between the endpoints. The first spring algorithm was proposed by Eades [16]. Since then several researchers have proposed variations of the spring algorithm [17–19]. These algorithms are also known as multidimensional scaling, in statistics. Kruskal and Seery noted their application to graph drawing in the late 1970s [19]. Spring algorithms are the most simplest and popular algorithms in force-directed placement graph drawing methods. The graph drawing problem is modeled by a force-directed algorithm models through a physical system of bodies with forces acting between them which minimizing the energy of the system by finding a good placement of the bodies [20]. Force-directed based graph drawing algorithms yield reasonable layouts in respect to symmetry, structure, clustering and vertex distribution [21].

The plotted graph of this molecule with 3 levels is shown in the Figure 1. The plotted graph by the spring algorithm is surprisingly close to the shape of this molecule and proves the usefulness of this algorithm for drawing real molecule as previously noted in [14]. Neato supports several shapes for nodes of a graph including circle, point, etc. The shape of nodes of Figures 3 is circle, but point used in Figures 1, 2, 4 and 5 for achieving the better figure. Neato also supports more dimensions for drawing a graph. The Figure 6 shows the three dimensional image of plotted graph of the molecule with four levels. It is difficult to realized the sections and levels in a two dimensional figure which holds a three dimensional plotted graph.

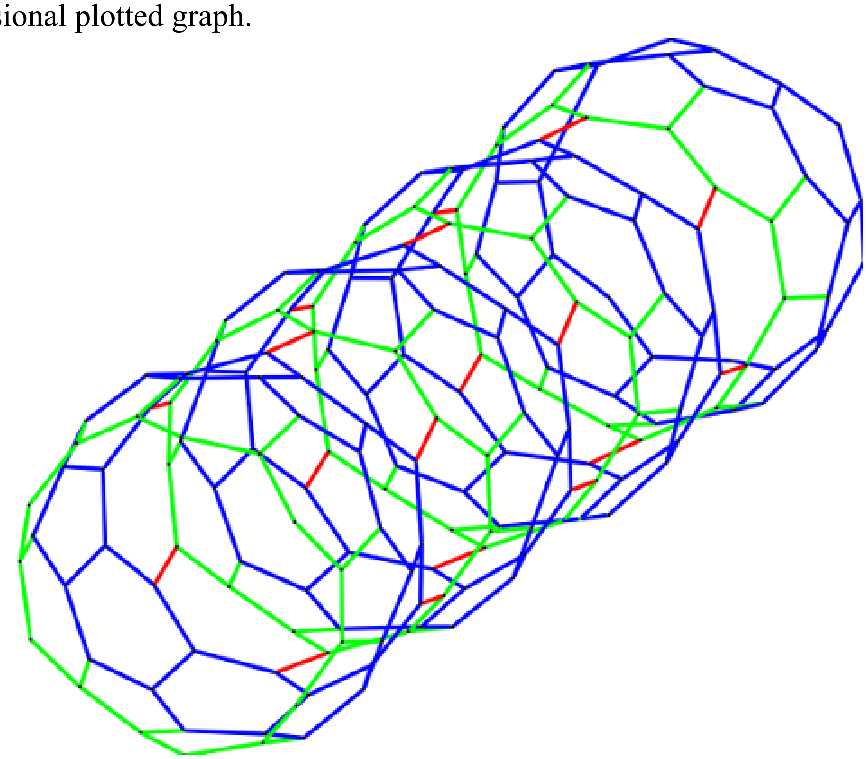

Figure 6. 3D View of Open Periodic Tubulene Plotted by Neato.

4 TOPOLOGICAL INDICES

A numerical invariant related to molecular graph of a chemical compound is called a topological index [22, 23]. Also there is a semi-empirical index which is discussed in [24,25]. Several different topological indices have been proposed to encode chemical properties of molecules [26]. These indices are calculated based on graphs of molecules or graphs of different kinds of networks such as social network [27]. So, by finding the graph of a molecule (or its adjacency matrix), computing the topological indices of that molecule is straightforward.

4.1 THE WIENER INDEX

The first topological index which is used in chemistry is the Wiener index. Harold Wiener developed and used the Wiener index to determine physicochemical properties of types of alkanes known as paraffin in 1947 [28]. To define, we assume that G is an indirect simple graph. The Wiener index, W(G), of G with n vertices is the sum of the lengths of shortest paths between all pairs of vertices of G.

$$W(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \quad ,d_{ii} = 0. \tag{1}$$

4.2 SZEGED INDEX

Ivan Gutman introduced the Szeged index in 1994 [29]. Let E(G) be the set of all edges of G and e = uv be an edge of G. Define $W(e) = n_u(e) \times n_v(e)$, where $n_u(e)$ is the number of vertices of G closer to u than v, and $n_v(e)$ is defined analogously. The Szeged index of G is the sum of W(e) over all edges of G [12]. So,

$$S_Z(G) = \sum_{e\in E(G)} W(e) \tag{2}$$

4.3 PI INDEX

The PI index was introduced by P. V. Khadikar in 2000 [30]. The summation over all edges uv of G which are not equidistant to u and v is PI index [31]. Based on the notations introduced in Szeged index, the PI index is defined as follows:

$$PI(G) = \sum_{e\in E(G)} [n_u(e) + n_v(e)]. \tag{3}$$

5 RESULTS

Finding the shortest paths between all nodes in a graph is the most time consuming part of computing of the preceding topological indices. The well known Dijkstra algorithm finds these shortest paths, but it is impossible to compute these topological indices based on this algorithm in a reasonable time. Therefore, the Floyd-Warshal algorithm is more suitable for this problem. This algorithm is basically equivalent to the transitive closure algorithm independently proposed by Roy [32] in 1959. Current version of this algorithm was proposed by Ingerman which used three nested for-loop. This algorithm is faster at the expense of memory. Therefore, the pitfall of this algorithm is the order of memory usage. The preceding results are computed on a computer with 12 GB main memory running Ubuntu 64-bit. The Table 1 shows the Wiener index, PI index and Szeged index of open tubulene with different level numbers.

Table 1: Values of some computed topological indices of diffrenet open Tubulene with different number of vertex

Level Number	Number of Vertices	Wiener Index	PI Index	Szeged Index
1	48	5496	2652	26622
10	480	2373096	303108	22289022
20	960	18575496	1238388	187132182
30	1440	62425896	2807268	646311342
40	1920	147748296	5009748	1551890502
50	2400	288366696	7845828	3055933662
70	3360	790787496	15418788	8467667982
100	4800	2304738696	31530228	24875349462
120	5760	3982199496	45439188	43114603782
140	6720	6323212296	61882548	68613762102
160	7680	9438369096	80860308	102589336422
180	8640	13438261896	102372468	146257838742
200	9600	18433482696	126419028	200835781062

The plot of these indices versus vertex numbers is more readable and clear than numbers in Table1. The Matplotlib visualization library is a standard package for curve plotting in Python. The Figures 7, 8 and 9 have been obtained using this library. The Matplotlib automatically regulates the axes ratio. It shows the axes ratio on the plot. The diagram of the Wiener index versus vertex number is shown in Figure 7. The axes ratio of this plot is 10^{10}.

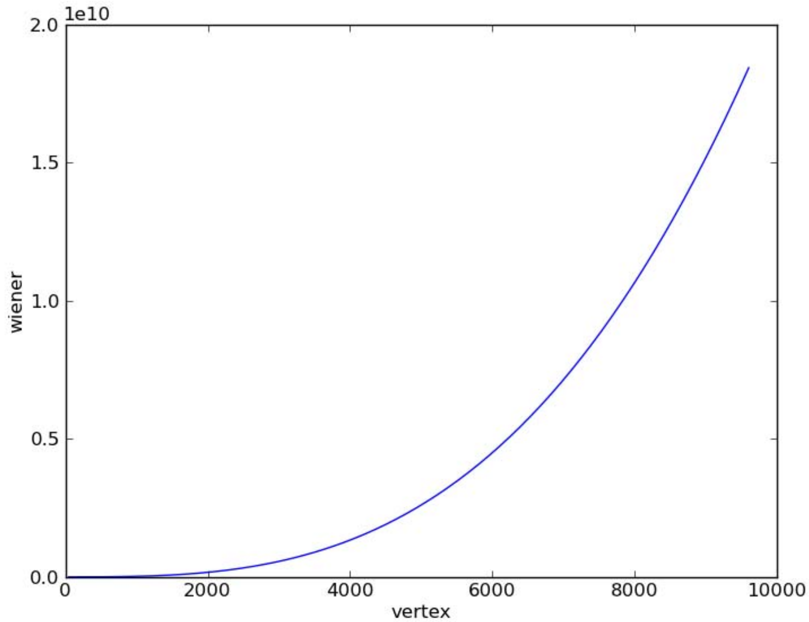

Figure 7. The Wiener Index Versus Vertex.

The diagram of the Szeged index versus vertex number is shown in Figure 8.

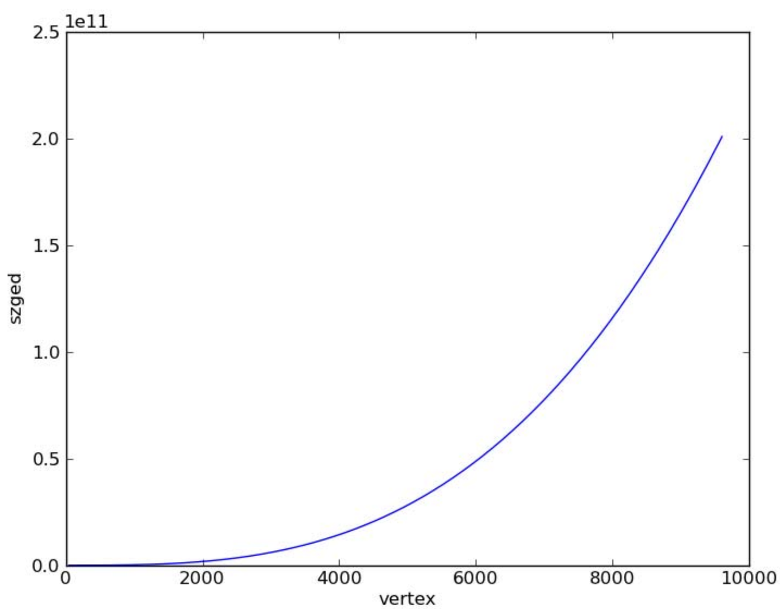

Figure 8. The Szeged Index Versus Vertex.

The diagram of the PI index versus vertex number is shown in Figure 9.

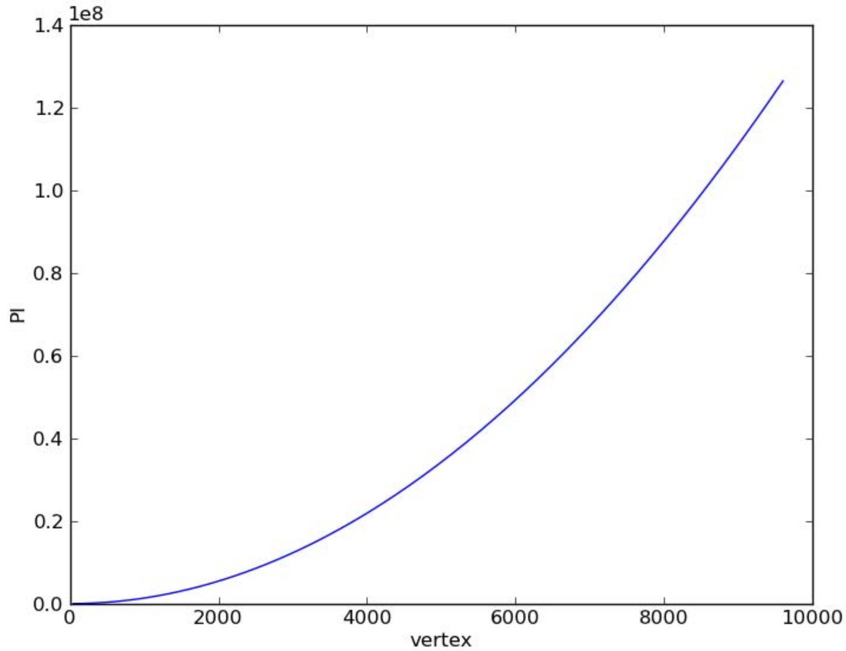

Figure 9. The PI Index Versus Vertex.

Table 2: Polynomial for the Wiener Index.

n	RMS	0	1	2	3	4	5
1	1886833697	–5214293688	1965227				
2	262437466	1771561032	–1487965	328.5			
3	**1.69458e–06**	**–5304**	**155**	**–7e–13**	**0.02**		
4	1.8528e–06	–5304	154.99	4e–12	0.02	2e–20	
5	1.026e–05	–5304	155	–7e–13	0.02	–1e–20	5e–25

Table 3: Polynomial for the PI Index.

n	RMS	0	1	2	3	4	5
1	7820939	–29239212	14422.5				
2	**1.04e–08**	**1428**	**–31.5**	**1.375**			
3	3.22e–08	1428	–31.5	1.375	3.4e–19		
4	2.27e–08	1428	–31.5	1.375	–1e–18	6e–23	
5	9.4e-08	1427.99	-31.5	1.375	-2e-18	2.e-22	

Table 4: Polynomial for The Szeged Index.

n	RMS	0	1	2	3	4	5
1	20635483310	–56905553562	21396835.25				
2	2886812122	19486947558	–16364897	3592.25			
3	**2.802e–05**	**–282138**	**4423.25**	**–21.25**	**0.22917**		
4	1.66e–05	–282137.9	4423.25	–21.25	0.22917	2e–19	
5	9.0298e–05	–282138	4423.25	–21.25	0.22917	8.7e–19	–3e–23

In Tables 2–4, RMS denotes the root mean square error of curve fitting. The polynomials recorded in Tables 2–4, when n = 3, 2, 3, are the best polynomials that is fitted to the Wiener, PI and Szeged indices of this molecule, respectively.

6 CONCLUSIONS

A new intuitive method for constructing the graph of open tubulene is proposed and discussed in this article. Several packages and tools based on Python programming language are used to implement the algorithm. The spring method is used to plot the constructed graph. A consequence of using this method is the similarity between the picture of open periodic tubulene and plotted graph. Three major topological indices, namely the Wiener, PI and Szeged indices of this molecule are calculated based on the constructed graph. Memory and time cost is the problems against calculating these indices unbounded.

Acknowledgement. This research has been supported by the research affair of the University of Kashan, I R Iran. I am very pleased from professor A R Ashrafi for helping me to correct the first draft of this paper.

REFERENCES

1. S. Berber, Y.-K. Kwon, and D. Tománek, “Electronic and structural properties of carbon nanohorns,” *Phys. Rev. B*, vol. **62**, no. 4, pp. 2291–2294, 2000.
2. Z. Dehouche, L. Lafi, N. Grimard, J. Goyette and R Chahine, “The catalytic effect of single–wall carbon nanotubes on the hydrogen sorption properties of sodium alanates,” *Nanotechnology*, vol. **16**, no. 4, pp. 402–409, 2005.
3. I. Zaporotskova, E. Prokofyeva and T. Ermakova, “Influence of bordering functional groups on processes of capillary filling of single-wall tubulenes with elemental hydrogen,” in *Fullerenes and Atomic Clusters IWFAC’2007*, St Petersburg, Russia, 2007.

4. Q. Wang and J. K. Johnson, “Optimization of Carbon nanotube arrays for Hydrogen adsorption,” *J. Phys. Chem. B*, vol. **103**, no. 23, pp. 4809–4813, 1999.
5. S. G. Chalk and J. F. Miller, “Key challenges and recent progress in batteries, fuel cells, and hydrogen storage for clean energy systems,” *J. Power Sources*, vol. **159**, no. 1, pp. 73–80, 2006.
6. I. Zaporotskova and E. Prokof’eva, “Internal investigation of saturation carbon nanotubes molecular hydrogen,” *Russ. J. Phys. Chem. B*, vol. **5**, no. 3, pp. 530–536, 2011.
7. A. Enyashin, S. Gemming, and G. Seifert, “Nanosized allotropes of molybdenum disulfide,” *Eur Phys J-Spec Top*, vol. 149, no. **1**, pp. 103–125, 2007.
8. T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Inform Process Lett*, vol. 31, no. **1**, pp. 7–15, 1989.
9. M. V. Diudea, D. Janezic, and A. Graovac, “Topology counting in nanostructures,” *Carpathian J. Math.*, vol. **20**, no. 2, pp. 223–234, 2004.
10. M. V. Diudea, “Tabular Nanostructures” Available: http://chem.ubbcluj.ro/~diudea/cursuri si referate/tubes1s.pdf . [Accessed: 04-Jun-2012].
11. A. R. Ashrafi and A. Loghman, “Computing Padmakar-Ivan Index of a TC4C8(R) Nanotorus,” *J. Comput. Theor. Nanosci.,* vol. **5**, no. 7, pp. 1431–1434, 2008.
12. G. H. Fath−Tabar, T. Došlić, and A. R. Ashrafi, “On the Szeged and the Laplacian Szeged spectrum of a graph,” *Linear Algebra Appl.*, vol. **433**, no. 3, pp. 662–671, 2010.
13. A. R. Ashrafi and A. Loghman, “PI index of zig-zag polyhex nanotubes,” *MATCH Commun. Math. Comput. Chem.*, vol. **55**, no. 2, pp. 447–452, 2006.
14. A. A. Yoosofan and A. R. Ashrafi, “Automatic generation of adjacency matrix of single−wall carbon nanohorn,” *Optoelect. Adv. Mat. Rapid Commun.,* vol. **4**, no. 6, pp. 900–901, 2010.
15. A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, 2008, pp. 11–15.
16. P. Eades, “A Heuristic for Graph Drawing,” *Congressus Numerantium*, vol. 42, pp. 149–160, 1984.
17. F. Brandenburg, M. Himsolt, and C. Rohrer, “An experimental comparison of force-directed and randomized graph drawing algorithms,” in *Graph Drawing*, 1996, pp. 76–87.
18. T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement ,” *Softw: Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
19. S. Cheng, “Graph drawing with spring algorithm,” MSc Thesis, Concordia University, 2002.
20. Y. F. Hu, “Efficient and high quality force−directed graph drawing,” *The Math. J.*, vol. 10, pp. 37–71, 2005.
21. E. Gansner and S. North, “Improved Force-Directed Layouts,” in *GD ’98: Proceedings of the 6th International Symposium on Graph Drawing*, 1998, pp. 364–373.

22. R. Natarajan, I. Nirdosh, and S. Muthuswami, “Application of topological indices to froth–flotation of an uranium ore,” *Curr. Sci India*, vol. **77**, no. 9, 10, pp. 1170–1174, 1999.
23. K. L. Lakshmi, “On Fr indices of phenylenes,” *Int. J. Comput. Cog.*, vol. **8**, no. 3, pp. 48–50, 2010.
24. B. da S. Junkes, R. D. de M. C. Amboni, R. A. Yunes and V. E. F. Heinzen, “Application of the semi-empirical topological index in quantitative structure-chromatographic retention relationship (QSRR) studies of aliphatic ketones and aldehydes on stationary phases of different polarity,” *J. Brazil. Chem. Soc.,* vol. **15**, pp. 183–189, 2004.
25. L. C. Porto, É. S. Souza, B. da S. Junkes, R. A. Yunes and V. E. F. Heinzen, “Semi-empirical topological index: Development of QSPR/QSRR and optimization for alkylbenzenes,” *Talanta*, vol. **76**, no. 2, pp. 407–412, 2008.
26. S. Nikolic and N. Raos, “Estimation of Stablity Constants of Mixed Amino Acid Complexes with Copper(II) from Topological Indices,” *Croat. Chem. Acta*, vol. **74**, no. 3, pp. 621–631, 2001.
27. H. González–Díaz and C. R. Munteanu, *Topological indices for medicinal chemistry, biology, parasitology, neurological and social networks*. Kerala, India: Transworld Research Network, 2010.
28. H. Wiener, “Structural determination of paraffin boiling points,” *J. Am. Chem. Soc.*, vol. **69**, no. 1, pp. 17–20, 1947.
29. I. Gutman, “A formula for the Wiener number of trees and its extension to graphs containing cycles,” *Graph Theory Notes New York*, vol. **27**, pp. 9–15, 1994.
30. P. Khadikar, “On a novel structural descriptor PI,” *Natl. Acad. Sci. Lett.*, vol. **23**, no. 7, 8, pp. 113–118, 2000.
31. H. Sabaghian–Bidgoli and A. R. Ashrafi, “A numerical method for computing PI index of fullerene molecules containing carbon atoms,” *J. Comput. Theor. Nanosci.*, vol. **6**, no. 7, pp. 1706–1708, 2009.
32. B. Roy, “Transitivité et connexité,” *C. R. Acad. Sci. Paris*, vol. **249**, pp. 216–218, 1959.