MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **74** (2015) 521-532

ISSN 0340 - 6253

The Edge–Wiener Index of Benzenoid Systems in Linear Time

Aleksander Kelenc[a], **Sandi Klavžar**[b,a,c], **Niko Tratnik**[a]

[a] *Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia*
aleksander.kelenc@um.si, niko.tratnik1@um.si

[b] *Faculty of Mathematics and Physics, University of Ljubljana, Slovenia*
sandi.klavzar@fmf.uni-lj.si

[c] *Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia*

(Received April 26, 2015)

Abstract

The edge-Wiener index of a graph G is defined as the Wiener index of the line graph of G. In this paper an algorithm is developed that, for a given benzenoid system G with m edges, computes the edge-Wiener index of G in $O(m)$ time. The key to the algorithm is a reduction of the problem to three different weighted trees. In addition to the previously used weighted vertex- and edge-Wiener indices, the so-called weighted vertex-edge-Wiener index is introduced and essentially used in the algorithm.

1 Introduction

The edge-Wiener index of a graph was independently introduced in [19, 20]. In [19] several possible variations of the concept were discussed and suggested that the edge-Wiener index of a graph G should be defined as

$$W_e(G) = \frac{1}{2} \sum_{e \in E(G)} \sum_{f \in E(G)} d_G(e, f)\,, \tag{1}$$

where $d_G(e,f)$ is the usual shortest-path distance between vertices e and f of the line graph $L(G)$ of G. In other words, $W_e(G)$ is just the Wiener index of the line graph of G, that is,

$$W_e(G) = W(L(G)). \tag{2}$$

Here we follow this convention because in this way the pair $(E(G), d)$ forms a metric space. On the other hand, for edges $e = ab$ and $f = xy$ of a graph G it is also legitimate to set

$$\widehat{d}_G(e,f) = \min\{d_G(a,x), d_G(a,y), d_G(b,x), d_G(b,y)\}. \tag{3}$$

Replacing d with $\widehat{d}$ in (1), a variant of the edge-Wiener index from [20] is obtained, let us denote it with $\widehat{W}_e(G)$ (it was denoted $W_{e1}(G)$ in [19]). It is easy to observe that $W_e(G)$ and $\widehat{W}_e(G)$ are connected in the following way (cf. [19, Corollary 8] and [20, Theorem 2.4]):

$$\widehat{W}_e(G) = W_e(G) - \binom{|E(G)|}{2}. \tag{4}$$

Because of (2) it is not surprising that the edge-Wiener index was investigated long before it was formally introduced, see [9, 10, 15, 16, 17]. Recent studies of the edge-Wiener index include [1, 2, 3, 8, 12, 31] as well as [24, 26, 29] where relations with related invariants are obtained. We point out that in [33] the so-called classical cut-method [22] was developed for the edge-Wiener index and that in the paper [23] (which rounds off several earlier papers) a complete solution of the equation $W(L^3(T)) = W(T)$ is given. For a recent survey on the edge-Wiener descriptors in chemical graph theory see [18].

In this paper we develop a linear time algorithm for the edge-Wiener index of benzenoid systems. These graphs are also know as the hexagonal systems (see [32, 34]) and form one of the most extensively studied family of chemical graphs. Papers [4, 7, 11, 13, 14, 25, 27, 28, 30] present a sample of relevant recent investigations.

The algorithm of this paper is parallel to the linear algorithm for the Wiener index that was developed in [6]. However, the present algorithm is a bit more involved and requires some additional insights. We proceed as follows. In the next section we give definitions and recall or introduce concepts needed later. In Section 3 we prove that the edge-Wiener index of a benzenoid system can be expressed as the sum of weighted Wiener-type indices of related weighted trees and also show how the vertex-edge Wiener index of a weighted tree can be computed. These results form the basis of the algorithm that is presented and analyzed in Section 4. In the final section an example is presented to illustrate the performance of the algorithm as well as to show that the method used can be applicable to determine the edge-Wiener index by hand.

2 Preliminaries

Unless stated otherwise, the graphs considered in this paper are connected. We have already defined $d_G(x,y)$ to be the distance between vertices $u,v \in V(G)$. From technical reasons we also set $\widehat{d}_G(x,y) = d_G(x,y)$. In addition, to be consistent with (3), for a vertex $x \in V(G)$ and an edge $e = ab \in E(G)$ we set

$$\widehat{d}_G(x,e) = \min\{d_G(x,a), d_G(x,b)\}\,.$$

The *Wiener index* of a graph G is defined as $W(G) = \frac{1}{2}\sum_{x\in V(G)}\sum_{y\in V(G)} d_G(x,y)$. To emphasize that it is the vertex-Wiener index, we will also write $W_v(G)$ for $W(G)$. The edge-Wiener index was already defined in (1), while the *vertex-edge Wiener index* is

$$W_{ve}(G) = \sum_{x\in V(G)}\sum_{e\in E(G)} d_G(x,e)\,.$$

In [20] the vertex-edge Wiener index is defined with the additional factor $1/2$, but for our purposes the present definition is more suitable. Moreover, in this way we consider all vertex-edge pairs exactly once, just like W considers all (unordered) vertex-vertex pairs and W_e all edge-edge pairs.

The above definitions of the Wiener indices extend to weighted graphs as follows. Let G be a graph and let $w : V(G) \to \mathbb{R}^+$ and $w' : E(G) \to \mathbb{R}^+$ be given functions. Then (G,w), (G,w'), and (G,w,w') are a *vertex-weighted graph*, an *edge-weighted graph*, and a *vertex-edge weighted graph*, respectively. The corresponding weighted Wiener indices are defined as

$$\begin{aligned}
W(G,w) &= \frac{1}{2}\sum_{x\in V(G)}\sum_{y\in V(G)} w(x)w(y)d_G(x,y)\,,\\
W_e(G,w') &= \frac{1}{2}\sum_{e\in E(G)}\sum_{f\in E(G)} w'(e)w'(f)d_G(e,f)\,,\\
W_{ve}(G,w,w') &= \sum_{x\in V(G)}\sum_{e\in E(G)} w(x)w'(e)\widehat{d}_G(x,e)\,.
\end{aligned}$$

The vertex version was for the first time introduced in [21]; we will again also write $W_v(G,w)$ for $W(G,w)$. $\widehat{W}_e(G,w')$ is defined analogously as $W_e(G,w')$, that is, by replacing $d_G(e,f)$ with $\widehat{d}_G(e,f)$ in the definition.

Let $\mathcal{H}$ be the hexagonal (graphite) lattice and let Z be a cricuit on it. Then a *benzenoid system* is induced by the vertices and edges of $\mathcal{H}$, lying on Z and in its interior. The edge set of a benzenoid system G can be naturally partitioned into sets E_1, E_2, and E_3 of edges of the same direction. For $i \in \{1,2,3\}$, set $G_i = G - E_i$. Then the connected components of the graph G_i are paths. The quotient graph T_i, $1 \le i \le 3$, has these paths as vertices, two such paths (i.e. components of G_i) P' and P'' being adjacent in T_i if some edge in E_i joins a vertex of P' to a vertex of P''. It is known that T_1, T_2 and T_3 are trees [5, 6]. Let (T_i, w_i) be the vertex-weighted tree T_i, where $w_i(x)$ is the number of vertices in the component (the path) of $G - E_i$ corresponding to x. Now we can recall the following fundamental result:

Proposition 2.1 [6, Proposition 2] *If G is a benzenoid system and (T_i, w_i), $1 \le i \le 3$, are the corresponding weighted quotient trees, then $W_v(G) = W_v(T_1, w_1) + W_v(T_2, w_2) + W_v(T_3, w_3)$.*

3 Preparation for the algorithm

Let G be a benzenoid system and let T_1, T_2, T_3 be its quotient trees as defined in the preliminaries. Then for every $i \in \{1,2,3\}$ we define $\alpha_i : E(G) \longrightarrow V(T_i) \cup E(T_i)$ by

$$\alpha_i(e) = \begin{cases} C \in V(T_i); & e \in E(C)\,, \\ C_1C_2 \in E(T_i); & e = ab \text{ and } a \in V(C_1), b \in V(C_2)\,. \end{cases} \tag{5}$$

Now we can state:

Theorem 3.1 *If G is a benzenoid system, then for every $e, f \in E(G)$,*

$$\widehat{d}_G(e,f) = \sum_{i=1}^{3} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f))\,.$$

Proof. Let $e = ab$, $f = xy$ and assume that $\widehat{d}_G(e,f) = \widehat{d}_G(a,x)$. Select a shortest path P from a to x in G and for every $i \in \{1,2,3\}$, set $F_i = E(P) \cap E_i$. As P is a shortest path, no two edges of F_i belong to the same cut. Since $\widehat{d}_G(e,f) = |F_1| + |F_2| + |F_3|$ it suffices to show that for $i \in \{1,2,3\}$ it holds $|F_i| = \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f))$. Let $C_1, C_2 \in V(T_i)$ be connected components of $G - E_i$ such that $a \in V(C_1)$ and $x \in V(C_2)$. It follows that $\widehat{d}_{T_i}(C_1, C_2) = |F_i|$. In order to show that $\widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) = \widehat{d}_{T_i}(C_1, C_2)$, we consider the following cases.

Case 1. $e \notin E_i$ and $f \notin E_i$.

In this case we have $\alpha_i(e) = C_1$ and $\alpha_i(f) = C_2$ and the desired conclusion is clear.

Case 2. One of e and f is in E_i but not the other.

We may without loss of generality assume that $e \in E_i$ and $f \notin E_i$. Then $\alpha_i(e) = C'C_1 \in E(T_i)$ for some $C' \in V(T_i)$ and $\alpha_i(f) = C_2$. Since $\widehat{d}_{T_i}(C_1, C_2) \leq \widehat{d}_{T_i}(C', C_2)$, it follows that $\widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) = \widehat{d}_{T_i}(C_1, C_2)$ (see Fig. 1).

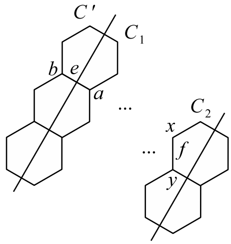

Figure 1: Edges e and f as in Case 2

Case 3. $e \in E_i$ and $f \in E_i$.

Now $\alpha_i(e) = C'C_1 \in E(T_i)$ for some $C' \in V(T_i)$ and $\alpha_i(f) = C''C_2$ for some $C'' \in V(T_i)$. We thus infer (cf. Fig. 2) that $\widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) = \widehat{d}_{T_i}(C_1, C_2)$.

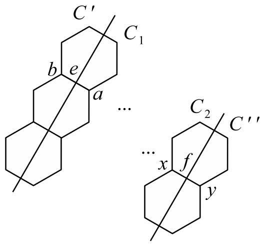

Figure 2: Edges e and f in Case 3

Since in any case $\widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) = \widehat{d}_{T_i}(C_1, C_2)$ holds, the proof is complete. ■

We next extend the quotient trees T_1, T_2, T_3 to weighted trees (T_i, w_i), (T_i, w_i'), (T_i, w_i, w_i') as follows:

- for $C \in V(T_i)$, let $w_i(C)$ be the number of edges in the component C of G_i;

- for $E = C_1C_2 \in E(T_i)$, let $w_i'(E)$ be the number of edges between components C_1 and C_2.

The next theorem is the key to our algorithm.

Theorem 3.2 *If G is a benzenoid system, then*

$$\widehat{W}_e(G) = \sum_{i=1}^{3} \left(\widehat{W}_e(T_i, w_i') + W_v(T_i, w_i) + W_{ve}(T_i, w_i, w_i') \right).$$

Proof. Applying Theorem 3.1 we obtain:

$$\begin{aligned}
\widehat{W}_e(G) &= \frac{1}{2} \sum_{e \in E(G)} \sum_{f \in E(G)} \widehat{d}_G(e,f) = \frac{1}{2} \sum_{e \in E(G)} \sum_{f \in E(G)} \left(\sum_{i=1}^{3} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) \right) \\
&= \sum_{i=1}^{3} \left(\frac{1}{2} \sum_{e \in E(G)} \sum_{f \in E(G)} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) \right).
\end{aligned}$$

Depending on the map α_i defined in (5), the inner sums can be partitioned into three sums such that

$$\begin{aligned}
\widehat{W}_e(G) &= \sum_{i=1}^{3} \left(\frac{1}{2} \sum_{\substack{e \in E(G) \\ \alpha_i(e) \in E(T_i)}} \sum_{\substack{f \in E(G) \\ \alpha_i(f) \in E(T_i)}} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) \right. \\
&\quad + \frac{1}{2} \sum_{\substack{e \in E(G) \\ \alpha_i(e) \in V(T_i)}} \sum_{\substack{f \in E(G) \\ \alpha_i(f) \in V(T_i)}} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) \\
&\quad \left. + \sum_{\substack{e \in E(G) \\ \alpha_i(e) \in V(T_i)}} \sum_{\substack{f \in E(G) \\ \alpha_i(f) \in E(T_i)}} \widehat{d}_{T_i}(\alpha_i(e), \alpha_i(f)) \right).
\end{aligned}$$

Taking into account the definition of the corresponding weighted trees we conclude that

$$\begin{aligned}
\widehat{W}_e(G) &= \sum_{i=1}^{3} \left(\frac{1}{2} \sum_{E \in E(T_i)} \sum_{F \in E(T_i)} w_i'(E) w_i'(F) \widehat{d}_{T_i}(E,F) \right. \\
&\quad + \frac{1}{2} \sum_{C_1 \in V(T_i)} \sum_{C_2 \in V(T_i)} w_i(C_1) w_i(C_2) \widehat{d}_{T_i}(C_1, C_2) \\
&\quad \left. + \sum_{C \in V(T_i)} \sum_{E \in E(T_i)} w_i(C) w_i'(E) \widehat{d}_{T_i}(C, E) \right) \\
&= \sum_{i=1}^{3} \left(\widehat{W}_e(T_i, w_i') + W_v(T_i, w_i) + W_{ve}(T_i, w_i, w_i') \right).
\end{aligned}$$

■

For a fast computation of the Wiener indices of weighted trees we need some additional notation. If T is a tree and $e \in E(T)$, then the graph $T - e$ consists of two components that will be denoted by $C_1(e)$ and $C_2(e)$. For a vertex-edge weighted tree (T, w, w') and $e \in E(T)$ set

$$n_i(e) = \sum_{u \in V(C_i(e))} w_i(u) \qquad \text{and} \qquad m_i(e) = \sum_{e \in E(C_i(e))} w'_i(e)\,.$$

Using this notation we recall the following results:

$$W(T, w) = \sum_{e \in E(T)} n_1(e) n_2(e) \tag{6}$$

and

$$\widehat{W}_e(T, w') = \sum_{e \in E(T)} m_1(e) m_2(e)\,, \tag{7}$$

where (6) was proved in [21], while (7) is a result from [33]. We next derive a related result for the vertex-edge Wiener index.

Proposition 3.3 *If (T, w, w') is a vertex-edge weighted tree, then*

$$W_{ve}(T, w, w') = \sum_{e \in E(T)} \big(n_1(e) m_2(e) + n_2(e) m_1(e)\big).$$

Proof. Let e be an edge of T. If $x \in V(T)$, $f \in E(T)$, and $f \neq e$, then set $\delta_e(x, f) = 0$ if x and f are in the same connected component of $T - e$, and $\delta_e(x, f) = 1$ otherwise. In addition, if $e = f$, then set $\delta_e(x, f) = 0$. Let $f = ab$ and assume that $\widehat{d}_T(x, f) = \widehat{d}_T(x, a)$ $(= d_T(x, a))$. Considering the (unique) shortest a, x-path in T we see that the edges of P are precisely the edges e, for which $\delta_e(x, f) \neq 0$ holds. It follows that

$$\widehat{d}_T(x, f) = \sum_{e \in E(T)} \delta_e(x, f).$$

With this equality we have

$$W_{ve}(T, w, w') = \sum_{x \in V(T)} \sum_{f \in E(T)} w(x) w'(f) \left(\sum_{e \in E(T)} \delta_e(x, f) \right).$$

Therefore,

$$W_{ve}(T, w, w') = \sum_{e \in E(T)} \left(\sum_{x \in V(T)} \sum_{f \in E(T)} w(x) w'(f) \delta_e(x, f) \right).$$

Because $\delta_e(x, f) = 1$ if and only if x is in one and f is in another connected component of $T - e$, the inner sum equals $n_1(e) m_2(e) + n_2(e) m_1(e)$. ■

4 The algorithm

We are now ready to design the announced algorithm. For a given benzenoid system G we first compute the quotient trees T_i using a procedure called `calculateQuotientTrees`. For each T_i we first compute the vertex weights w and the edge weights w' using a procedure `calculateWeights`. These weights are then updated in the algorithm and new weights w'' are computed using a procedure named `UpdateWeights`. (See the proof of Theorem 4.1 for details on this.) For a given i, the three terms from Theorem 3.2 are finally computed into $X_{i,1}$, $X_{i,2}$, and $X_{i,3}$ and (4) is used to obtain the final result. The algorithm reads as follows:

Algorithm 1: Edge-Wiener Index of Benzenoid Systems

Input : Benzenoid system G with m edges
Output: $W_e(G)$

$(T_1, T_2, T_3) \leftarrow$ `calculateQuotientTrees` (G)
for $i = 1$ to 3 **do**
  $(w_i, w_i') \leftarrow$ `calculateWeights` (T_i, G)
  `updateWeights` (T_i, w_i, w_i', w_i'')
  $X_{i,1} \leftarrow W_v(T_i, w_i, w_i', w_i'')$
  $X_{i,2} \leftarrow W_e(T_i, w_i, w_i', w_i'')$
  $X_{i,3} \leftarrow W_{ve}(T_i, w_i, w_i', w_i'')$
  $Y_i \leftarrow X_{i,1} + X_{i,2} + X_{i,3}$
end
$W_e(G) \leftarrow Y_1 + Y_2 + Y_3 + \binom{m}{2}$

Theorem 4.1 *If G is a benzenoid system with m edges, then Algorithm 1 correctly computes $W_e(G)$ and can be implemented in $O(m)$ time.*

Proof. The correctness of the algorithm follows from Theorem 3.2 and Equality (4).

For the time complexity we first recall from [5] that the quotient trees T_i, $1 \leq i \leq 3$, can be computed in linear time. It is also straightforward to determine the corresponding weights w_i and w_i'. The fact that the Wiener index of a weighted tree can be computed within the same time was proved in [6] (see Lemma 3 from the paper and the text afterwards).

Using a method parallel to the above method from [6] to compute the Wiener index of a weighted tree, the for loop of the algorithm can be implemented to run in $O(m)$ time. More precisely, consider T_i as a rooted tree with a root x and label the vertices of T_i such that, if a vertex y is labelled ℓ, then all vertices in the subtree rooted at y have labels smaller than ℓ. For instance, such a labeling can be obtained by ordering the vertices of T_i with respect to the distance levels from x, the most distant vertices receiving the smallest labels. Using the standard BFS algorithm this can be done in linear time. When traversing the tree T_i we visit vertices according to this labeling.

In Line 4, we update weights w_i and calculate new weights w_i'' as follows. For each vertex y, the new weight $w_i(y)$ is computed as the sum of all the weights of vertices in the subtree rooted at y, while the weight $w_i''(y)$ is obtained as the sum of all the weights of the edges in the subtree rooted at y. To calculate these weights we traverse T_i and proceed as follows. Suppose that y is a vertex just visited. If y is a leaf, then $w_i(y)$ is left unchanged and $w_i''(y)$ is set to 0. Otherwise update $w_i(y)$ by adding to it $w_i(z)$ for all down-neighbours z of y, and compute $w_i''(y)$ as the sum of $w_i''(z)$ and $w_i'(e)$ for all down-neighbors z of y and all the corresponding edges e. Since each of the weights in the computations is used a constant number of times (actually at most two times), the time complexity of the for loop is also $O(m)$.

Note that for every vertex y of the tree T_i, we can consider the subtree rooted at y as a connected component of the graph $T_i - e$, where e is the up-edge of y. Therefore, $n_1(e) = w_i(y)$ and $m_1(e) = w_i''(y)$. Denote with $n_{T_i} = \sum_{v \in V(T_i)} w_i(v)$ and $m_{T_i} = \sum_{f \in E(T_i)} w_i'(f)$ the sum of the weights of the vertices and edges of T_i, respectively. Then $n_2(e) = n_{T_i} - n_1(e)$ and $m_2(e) = m_{T_i} - m_1(e) - w_i'(e)$. Due to Equations (6), (7), and Proposition 3.3, Lines 5 to 7 can be computed in linear time. We conclude that Algorithm 1 can be implemented in $O(m)$ time. ■

5 An example

We conclude the paper with an example that demonstrates how the designed algorithm performs and how the method can also be used by hand. Consider the benzenoid system G from Fig. 3 with $m = 30$ edges.

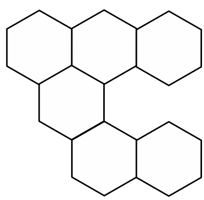

Figure 3: Benzenoid system G

First, for $i = 1, 2, 3$, we determine the graphs $G - E_i$, where E_i is the set of edges of G of the same direction. These graphs are represented in Fig. 4 with thick edges. Afterwards we determine the weighted quotient trees (T_i, w_i, w'_i), $1 \le i \le 3$, see Fig. 5. We next compute (note that this can also be done by hand!) the quantities:

$$W_v(T_1, w_1) = 250, \quad \widehat{W}_e(T_1, w'_1) = 26, \quad W_{ve}(T_1, w_1, w'_1) = 179,$$
$$W_v(T_2, w_2) = 266, \quad \widehat{W}_e(T_2, w'_2) = 12, \quad W_{ve}(T_2, w_2, w'_2) = 126,$$
$$W_v(T_3, w_3) = 231, \quad \widehat{W}_e(T_3, w'_3) = 12, \quad W_{ve}(T_3, w_3, w'_3) = 128.$$

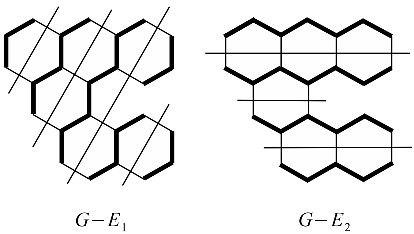

Figure 4: Graphs $G - E_1$, $G - E_2$, and $G - E_3$

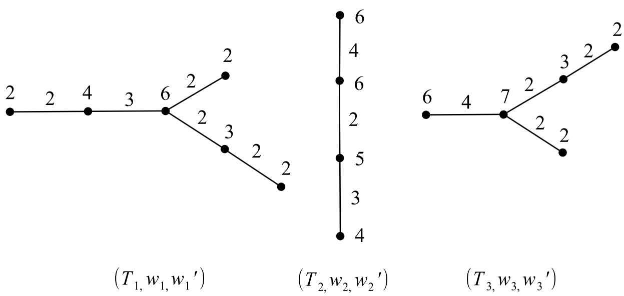

Figure 5: Weighted quotient trees

By Theorem 3.2 and Equation (4) we then conclude that

$$W_e(G) = (250 + 26 + 179) + (266 + 12 + 126) + (231 + 12 + 128) + \binom{30}{2} = 1665.$$

Acknowledgment: S. K. supported in part by the Ministry of Science of Slovenia under the grant P1-0297.

References

[1] Y. Alizadeh, A. Iranmanesh, T. Došlić, M. Azari, The edge Wiener index of suspensions, bottlenecks, and thorny graphs, *Glas. Mat. Ser. III* **49(69)** (2014) 1–12.

[2] M. Azari, A. Iranmanesh, A. Tehranian, A method for calculating an edge version of the Wiener number of a graph operation, *Util. Math.* **87** (2012) 151–164.

[3] M. Azari, A. Iranmanesh, A. Tehranian, Two topological indices of three chemical structures, *MATCH Commun. Math. Comput. Chem.* **69** (2013) 69–86.

[4] M. Berlič, N. Tratnik, P. Žigert Pleteršek, Equivalence of Zhang-Zhang polynomial and cube polynomial for spherical benzenoid systems, *MATCH Commun. Math. Comput. Chem.* **73** (2015) 443–456.

[5] V. Chepoi, On distances in benzenoid systems, *J. Chem. Inf. Comput. Sci.* **36** (1996) 1169–1172.

[6] V. Chepoi, S. Klavžar, The Wiener index and the Szeged index of benzenoid systems in linear time, *J. Chem. Inf. Comput. Sci.* **37** (1997) 752–755.

[7] C.-P. Chou, H. A. Witek, ZZDecomposer: a graphical toolkit for analyzing the Zhang–Zhang polynomials of benzenoid structures, *MATCH Commun. Math. Comput. Chem.* **71** (2014) 741–764.

[8] P. Dankelmann, I. Gutman, S. Mukwembi, H. Swart, The edge-Wiener index of a graph, *Discr. Math.* **309** (2009) 3452–3457.

[9] A. A. Dobrynin, Distance of iterated line graphs, *Graph Theory Notes N. Y.* **37** (1999) 8–9.

[10] A .A. Dobrynin, L. S. Mel'nikov, Wiener index of graphs and their line graphs with arbitrary large cyclomatic number, *Appl. Math. Lett.* **18** (2005) 307–312.

[11] T. Došlić, I. Zubac, Saturation number of benzenoid graphs *MATCH Commun. Math. Comput. Chem.* **73** (2015) 491–500.

[12] Y. Dou, H. Bian, H. Gao, H. Yu, The polyphenyl chains with extremal edge-Wiener indices, *MATCH Commun. Math. Comput. Chem.* **64** (2010) 757–766.

[13] V. Gineityte, Perturbative analogue for the concept of conjugated circuits in benzenoid hydrocarbons, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 39–73.

[14] J. E. Graver, E. J. Hartung, Kekuléan benzenoids, *J. Math. Chem.* 52 (2014) 977–989.

[15] I. Gutman, Distance of line graphs, *Graph Theory Notes N. Y.* **31** (1996) 49–52.

[16] I. Gutman, V. Jovašević, A. A. Dobrynin, Smallest graphs for which the distance of the graph is equal to the distance of its line graph, *Graph Theory Notes N. Y.* **33** (1997) 19–19.

[17] I. Gutman, L. Pavlović, More on distance of line graphs, *Graph Theory Notes N. Y.* **33** (1997) 14–18.

[18] A. Iranmanesh, M. Azari, Edge–Wiener descriptors in chemical graph theory: a survey, *Curr. Org. Chem.* **19** (2015) 219–239.

[19] A. Iranmanesh, I. Gutman, O. Khormali, A. Mahmiani, The edge versions of Wiener index, *MATCH Commun. Math. Comput. Chem.* **61** (2009) 663–672.

[20] M. H. Khalifeh, H. Yousefi Azari, A. R. Ashrafi, S. G. Wagner, Some new results on distance–based graph invariants, *Eur. J. Comb.* **30** (2009) 1149–1163.

[21] S. Klavžar, I. Gutman, Wiener number of vertex–weighted graphs and a chemical application, *Discr. Appl. Math.* **80** (1997) 73–81.

[22] S. Klavžar, M. J. Nadjafi–Arani, Cut method: update on recent developments and equivalence of independent approaches, *Curr. Org. Chem.* **19** (2015) 348–358.

[23] M. Knor, M. Mačaj, P. Potočnik, R. Škrekovski, Complete solution of equation $W(L^3(T)) = W(T)$ for the Wiener index of iterated line graphs of trees, *Discr. Appl. Math.* **171** (2014) 90–103.

[24] M. Knor, P. Potočnik, R. Škrekovski, Relationship between the edge–Wiener index and the Gutman index of a graph, *Discr. Appl. Math.* **167** (2014) 197–201.

[25] J. Kovič, T. Pisanski, A. T. Balaban, P. W. Fowler, On symmetries of benzenoid systems, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 3–26.

[26] M. J. Nadjafi–Arani, H. Khodashenas, A. R. Ashrafi, Relationship between edge Szeged and edge Wiener indices of graphs, *Glas. Mat. Ser. III* **47(67)** (2012) 21–29.

[27] J. Rada, R. Cruz, I. Gutman, Benzenoid systems with extremal vertex–degree–based topological indices, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 125–136.

[28] A. Taranenko, P. Žigert, Resonant sets of benzenoid graphs and hypercubes of their resonance graphs, *MATCH Commun. Math. Comput. Chem.* **68** (2012) 65–77.

[29] A. Soltani, A. Iranmanesh, Z. A. Majid, The multiplicative version of the edge Wiener index, *MATCH Commun. Math. Comput. Chem.* **71** (2014) 407–416.

[30] A. Vesel, Fast computation of Clar formula for benzenoid graphs without nice coronenes, *MATCH Commun. Math. Comput. Chem.* **71** (2014) 717–740.

[31] B. Wu, Wiener index of line graphs, *MATCH Commun. Math. Comput. Chem.* **64** (2010) 699–706.

[32] S.-J. Xu, H. Zhang, J. Cai, Complete forcing numbers of catacondensed hexagonal systems, *J. Comb. Optim.* **29** (2015) 803–814.

[33] H. Yousefi–Azari, M. H. Khalifeh, A. R. Ashrafi, Calculating the edge Wiener and edge Szeged indices of graphs, *J. Comput. Appl. Math.* **235** (2011) 4866–4870.

[34] H. Zhang, J. Cai, On the global forcing number of hexagonal systems, *Discr. Appl. Math.* **162** (2014) 334–347.