Efficient computation of trees
with minimal atom-bond connectivity index

Darko Dimitrov
Institut für Informatik, Freie Universität Berlin,
Takustraße 9, D–14195 Berlin, Germany
E-mail: darko@mi.fu-berlin.de

Abstract
The atom-bond connectivity (ABC) index is one of the recently most investigated degree-based molecular structure descriptors, that have applications in chemistry. For a graph $G$, the ABC index is defined as $\sum_{uv \in E(G)} \sqrt{\frac{(d(u)+d(v)-2)}{d(u)d(v)}}$, where $d(u)$ is the degree of vertex $u$ in $G$ and $E(G)$ is the set of edges of $G$. Despite many attempts in the last few years, it is still an open problem to characterize trees with minimal ABC index. In this paper, we present an efficient approach of computing trees with minimal ABC index, by considering the degree sequences of trees and some known properties of the graphs with minimal ABC index. The obtained results disprove some existing conjectures and suggest new ones to be set.

1 Introduction and some related results

Molecular descriptors [26] are mathematical quantities that describe the structure or shape of molecules, helping to predict the activity and properties of molecules in complex experiments. Among them, the graph topological indices [10] play a significant role. In 1998, Estrada et al. [13] proposed a new vertex-degree-based graph topological index, the atom-bond connectivity (ABC) index, and showed that it can be a valuable predictive tool in the study of the heat of formation in alkenes. Later, the physico-chemical applicability of the ABC index was confirmed and extended in several studies [9, 12, 17, 20].

Let $G = (V, E)$ be a simple undirected graph of order $n = |V|$ and size $m = |E|$. For $v \in V(G)$, the degree of $v$, denoted by $d(v)$, is the number of edges incident to $v$. Then the atom-bond connectivity index of $G$ is defined as

$$ABC(G) = \sum_{uv \in E(G)} \sqrt{\frac{(d(u)+d(v)-2)}{d(u)d(v)}}. \quad (1)$$

As a new and well motivated graph invariant, the ABC index has attracted a lot of interest in the last several years both in mathematical and chemical research communities.
and numerous results and structural properties of ABC index were established [3, 4, 5, 6, 7, 8, 14, 18, 19, 22, 27, 29, 30, 31].

The fact that adding an edge in a graph strictly increases its ABC index [7] (or equivalently that deleting an edge in a graph strictly decreases its ABC index [3]) has the following two immediate consequences.

**Corollary 1.1.** Among all connected graphs with \( n \) vertices, the complete graph \( K_n \) has maximal value of ABC index.

**Corollary 1.2.** Among all connected graphs with \( n \) vertices, the graph with minimal ABC index is a tree.

Although it is fairly easy to show that the star graph \( S_n \) is a tree with maximal ABC index [14], despite many attempts in the last years, it is still an open problem the characterization of trees with minimal ABC index (also refereed as minimal-ABC trees). To accomplish that task, besides the theoretically proven properties of the trees with minimal ABC index, computer supported search can be of enormous help. A good example of that is the work done by Furtula et al. [15], where the trees with minimal ABC index of up to size of 31 were computed. There, a brute-force approach of generating all trees of a given order, speeded up by using a distributed computing platform, was applied.

Here, we improve the computer search in two ways. Firstly, we consider only the degree sequences of trees. We would like to stress that the number of degree sequences of a given length \( n \) is significantly smaller than the number of all trees of order \( n \). For example, the number of trees with 32 vertices is 109,972,410,221 [25], while the number of degree sequences of length 32 is 5604 (see Table 1). Secondly, to speed up the computation, we generate only degree sequences of trees that correspond to some known structural properties of the trees with minimal ABC index (Propositions 3.1, 3.2 and 3.3 from Section 3). Thus, using a single PC, we have identified all trees with minimal ABC of order up to 2000. The obtained results strengthen the believe that some conjectures are true and reject other conjectures.

In the sequel, we present some additional results and notation that will be used in the rest of the paper. A vertex of degree one is a *pendant vertex*. As in [19], a sequence of vertices of a graph \( G, S_k = v_0 v_1 \ldots v_k \), will be called a *pendant path* if each two consecutive vertices in \( S_k \) are adjacent in \( G \), \( d(v_0) > 2, d(v_i) = 2, \) for \( i = 1, \ldots, k - 1 \), and \( d(v_k) = 1 \). The length of the pendant path \( S_k \) is \( k \). A sequence \( D = (d_1, d_2, \ldots, d_n) \) is *graphical* if there is a graph whose vertex degrees are \( d_i, i = 1, \ldots, n \). If in addition \( d_1 \geq d_2 \geq \cdots \geq d_n \), then \( D \) is a *degree sequence*. Let \( D_n \) be the set of all degree sequences of trees of length \( n \).

In [28] Wang defined a *greedy tree* as follows.

**Definition 1.1** ([28]). *Suppose the degrees of the non-leaf vertices are given, the greedy tree is achieved by the following ‘greedy algorithm’:

1. Label the vertex with the largest degree as \( v \) (the root).

2. Label the neighbors of \( v \) as \( v_1, v_2, \ldots \), assign the largest degree available to them such that \( d(v_1) \geq d(v_2) \geq \ldots \)

3. Label the neighbors of \( v_1 \) (except \( v \)) as \( v_{11}, v_{12}, \ldots \) such that they take all the largest degrees available and that \( d(v_{11}) \geq d(v_{12}) \geq \ldots \) then do the same for \( v_2, v_3, \ldots \)
The following result by Gan, Liu and You [16] characterizes the trees with minimal ABC index with prescribed degree sequences.

**Theorem 1.3.** Given the degree sequence, the greedy tree minimizes the ABC index.

The same result as in Theorem 1.3, using slightly different notation and approach, was obtained by Xing and Zhou [29]. Since the Theorem 1.3 plays a crucial role in our computation, the first important issue is how efficiently to enumerate degree sequences of trees. This problem is considered in the next section.

## 2 Enumerating degree sequences of trees

There exist several algorithms for enumerating degree sequences of graphs. A comprehensive source of references of such algorithms can be found in [21]. Clearly, each of those algorithms can be used for enumerating degree sequences of trees just by considering only the degree sequences with sum of degrees equals to \(2n - 2\), where \(n\) is the length of the degree sequences. However, this is not an efficient approach, because most of the generated degree sequences are not degree sequences of trees. For an illustration, the number of all degree sequences of length 29 is \(2022337118015338\) [24], while the number of degree sequences that correspond to trees of order 29 is 3010 (see Table 1). Thus, it is not a surprise that the largest reported enumerated degree sequences of graphs was only of length 29, with running time of 6733 days, distributed to 200 PCs containing about 700 cores [21]. Since we are not aware of an algorithm specialized only for enumerating degree sequences of trees, we present such an algorithm in this section. Our algorithm is related to the algorithm of enumerating degree sequences of graphs presented by Ruskey et al [23], and exploit the so called “reverse search”, a term originated by Avis and Fukuda [1]. Therefore, in the sequel, we will adopt the notation used in [23]. The main result, on which our algorithm is based, is the following characterization of a degree sequence of a tree.

**Theorem 2.1.** A sequence of integers \(D = (d_1, d_2, \cdots, d_n)\), with \(n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_{n-m} > d_{n-m+1} = \cdots = d_n = 1\), is the degree sequence of a tree if and only if \(C = (c_1, c_2, \cdots, c_{n-d_{n-m}+1})\) is the degree sequence of a tree, where

\[
c_i = \begin{cases} d_i & i \leq n - m - 1; \\ 1 & \text{otherwise}. \end{cases}
\]  

**Proof.** Let \(T_C\) be a tree with degree sequence \(C = (c_1, c_2, \cdots, c_{n-d_{n-m}+1})\), with \(c_1 \geq c_2 \geq \cdots \geq c_{n-m-1} > c_{n-m} = \cdots = c_{n-d_{n-m}+1} = 1\) and \(c_{n-m-1} \geq d_{n-m} \geq 2\), satisfying (2).

To prove the easier direction of the equivalence, just add \(d_{n-m} - 1\) pendant vertices to a pendant vertex of \(T_C\), obtaining a tree \(T_D\). The degree sequence that corresponds to \(T_D\) is \(D = (d_1, d_2, \cdots, d_n)\), with \(n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_{n-m} > d_{n-m+1} = \cdots = d_n = 1\).

The other direction of the equivalence, we prove as follows. Let \(D = (d_1, d_2, \cdots, d_n)\), with \(n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_{n-m} > d_{n-m+1} = \cdots = d_n = 1\), be a degree sequence of a tree.
tree $T_D$. Let $v_{n-m}$ be the vertex with degree $d_{n-m}$. If $v_{n-m}$ has $d_{n-m} - 1$ pendant vertices, then delete them obtaining the tree $T_C$. If this is not a case, i.e., $v_{n-m}$ has $d > 1$ adjacent vertices of degree bigger than one that comprised a set $U = \{u_1, u_2, \ldots, u_d\}$. Let $U_1$ be a set of adjacent vertices to $u_1$. First, delete all edges between $u_1$ and vertices in $U_1 \setminus \{v_{n-m}\}$ and add edges between vertices in $U_1 \setminus \{v_{n-m}\}$ and a pendant vertex whose distance to $u_1$ is bigger than its distance to any other vertex in $U$. Notice that $T_D$ has more than $d_{n-m}$ pendant vertices, therefore such pendant vertex must exists. Repeat the same as for $u_1$, for the rest of the vertices $u_2, u_3, \ldots, u_d$, considering one vertex per step until $v_{n-m}$ has $d_{n-m} - 1$ pendant vertices, obtaining a tree $T_D'$. Observe that $T_D'$ has the same degree sequence as $T_D$. Finally, in $T_D'$ delete all $d_{n-m} - 1$ pendant vertices adjacent to $v_{n-m}$, obtaining the tree $T_C$.

Let $S_i$ be the set of all sequences $D_i = (d_1, d_2, \ldots, d_i)$, with fixed length $i$, where $1 < i \leq n$ and $n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_i$ if and only if $D_i = (D_{i-d_i+1}, d_i) \in D_{i-d_i+1}$. Define a function $f_i : S_i \times d_i \to S_i - d_i + 1 \times d_i - 1$ such that for a given $D_i \in S_i$ and $C = (c_1, c_2, \cdots, c_{i-c_{i+1}})$, it holds that $(C, c_{i+1}) = f_i(D_i, d_i)$ if

$$c_k = \begin{cases} \frac{d_k}{1} & \text{if } k \leq h_i - 1; \\ \frac{1}{1} & \text{otherwise.} \end{cases}$$

By Theorem 2.1 and definition of the function $f_i$, we have the following two corollaries.

**Corollary 2.2.** For $i > 0$ and $D_i \in S_i$, the sequence $D_i \in D_i$ if and only if $f_i(D_i, d_i) = (D_{i-d_i+1}, d_i) \in D_{i-d_i+1}$.

**Corollary 2.3.** Let $C = (c_1, c_2, \ldots, c_{i-k}, c_{i+1}) \in D_{i-k+1}$, with $c_{i-k}$ the smallest degree bigger than one, and $2 \leq z \leq c_{i-k}$. The sequence $D_i = (d_1, d_2, \ldots, d_i) \in f_i^{-1}(C, c_{i-k})$ if and only if

$$d_k = \begin{cases} \frac{c_k}{1} & \text{if } k \leq h_i; \\ \frac{z}{1} & \text{if } k = h_i + 1; \\ \frac{1}{1} & \text{otherwise.} \end{cases}$$

The following example illustrates Corollary 2.3:

$$f^{-1}(651111111111111, 654111111111111, 653111111111111, 652111111111111) \supseteq \{655111111111111, 655111111111111, 655111111111111, 652111111111111\}.$$
Table 1: Performance of the algorithm for enumerating degree sequences of trees. For degree sequences of length \( n \), \( S(n) \) denotes the number of degree sequences, \( T(n) \) denotes the total running time and \( S(n)/T(n) \) denotes the amortized running time for generating a sequence.

| \( n \) | \( S(n) \) | \( T(n) \) | \( T(n)/S(n) \) [ms] | \( n \) | \( S(n) \) | \( T(n) \) | \( T(n)/S(n) \) [ms] |
|-----|-----|-----|------------------|-----|-----|-----|------------------|
| 21  | 490 | 3.031ms | 0.00618571 | 34  | 8349 | 0.041s | 0.00618571 |
| 22  | 627 | 3.989ms | 0.00636204 | 35  | 10143| 0.051s | 0.00601538 |
| 23  | 792 | 4.875ms | 0.00615530 | 40  | 26015| 0.098s | 0.00502455 |
| 24  | 1002| 6.082ms | 0.00606986 | 50  | 14727| 0.612s | 0.00415907 |
| 25  | 1255| 7.884ms | 0.00628207 | 60  | 715220| 3.250s | 0.00451627 |
| 26  | 1575| 9.996ms | 0.00634667 | 70  | 3087735| 15.300s| 0.00495518 |
| 27  | 1958| 13.083ms| 0.00668182 | 80  | 12132164| 1m3s  | 0.00520250 |
| 28  | 2436| 14.434ms| 0.00592529 | 90  | 44108109| 4m5s  | 0.00556759 |
| 29  | 3010| 20.086ms| 0.00667309 | 100 | 150198136| 14m27s| 0.00577865 |
| 30  | 3718| 18.821ms| 0.00506213 | 110 | 483502844| 51m26s| 0.00635496 |
| 31  | 4565| 23.031ms| 0.00504513 | 120 | 1482074143| 2h39m8s| 0.00647126 |
| 32  | 5604| 29.523ms| 0.00526820 | 130 | 4351076800| 7h36m43s| 0.00629813 |
| 33  | 6842| 33.430ms| 0.00488600 | 140 | 12292341831| 21h7m44s| 0.00618793 |

To improve the computation by considering the distribution of the graphical sequences according to their first element. That can help to design an algorithm that computes the new values of \( S(n) \) by slicing of the computations belonging to a given value of \( n \), similarly as it was done in [21].

Having all degree sequences of a particular length, in the next section we proceed to determine the trees with minimal ABC index.

3 Trees with minimal atom-bond connectivity index

Our algorithm of identifying the trees with minimal ABC index is comprised of the following steps:

1. Enumerate all degree sequences as described in Section 2.
2. Find corresponding ‘greedy trees’ for each generated degree sequence applying Theorem 1.3.
3. Calculate the ABC index of each ‘greedy tree’ and select the tree with minimal value.

Computationally, step 1. is the most expensive one, and as pointed in Section 2, it can be parallelized to some extent to run in distributed framework. Steps 2. and 3. can be implemented straightforwardly and there computational cost is linear with the respect to the length of the degree sequence. Although this approach is computationally superior to the computer assisted search presented in [15], it can be considerably improved by enumerating
only the degree sequences that satisfy the following structural properties of the minimal ABC trees.

**Proposition 3.1** ([19]). If \( n \geq 10 \), then the \( n \)-vertex tree with minimal ABC index does not contain pendent paths of length \( k \geq 4 \).

**Proposition 3.2** ([19]). If \( n \geq 10 \), then the \( n \)-vertex tree with minimal ABC index contains at most one pendent path of length \( k = 3 \).

**Proposition 3.3** ([22]). If \( n \geq 10 \), then each pendent vertex of the \( n \)-vertex tree \( G \) with minimal ABC index belongs to a pendent path of length \( 2 \leq k \leq 3 \).

Considering all these results, that reduce significantly the number of degree sequences, we have implemented an algorithm that identifies trees with minimal ABC index. On our single processor platform, we have calculated all trees with minimal ABC index of order up to 2000 in about 13 days. All obtained trees with minimal ABC index are summarized in Figures 1 and 2. For the sake of completeness, we include also the results for \( 7 \leq n \leq 31 \), which were already obtained in [15, 19]. For \( n \leq 6 \), the minimal ABC trees are paths \( P_n \) and they are omitted in the figures.

The obtained results lead to disprovement of some existing conjectures. The plausible structural computational model and its refined version in [15], is based on the main assumption that the minimal ABC tree posses a single central vertex, or said with other words, it is based on the assumption that the vertices of a minimal ABC tree of degree \( \geq 3 \) induce a star graph. The configuration \( T_4 \) in Figure 3, for \( n \equiv 4 \) (mod 7) and \( n \geq 312 \), is an counterexample to that assumption.

Also the configuration \( T_2 \) in Figure 3 indicates different structure of the minimal ABC trees, in the case when \( n \equiv 2 \) (mod 7) and \( n \geq 1185 \), than the structure suggested for this case in [15, 18].

As a consequence to these counterexamples, we present a revised version of the conjecture by Gutman and Furtula [18] about the trees with minimal ABC index. Our computations support the new conjecture in the cases when the order of a tree is up to 2000.

**Conjecture 3.1.** Let \( G \) be a tree with minimal ABC index among all trees of size \( n \).

(i) If \( n \equiv 0 \) (mod 7) and \( n \geq 175 \), then \( G \) has the structure \( T_0 \) depicted in Figure 3.

(ii) If \( n \equiv 1 \) (mod 7) and \( n \geq 64 \), then \( G \) has the structure \( T_1 \) depicted in Figure 3.

(iii) If \( n \equiv 2 \) (mod 7) and \( n \geq 1185 \), then \( G \) has the structure \( T_2 \) depicted in Figure 3.

(iv) If \( n \equiv 3 \) (mod 7) and \( n \geq 80 \), then \( G \) has the structure \( T_3 \) depicted in Figure 3.

(v) If \( n \equiv 4 \) (mod 7) and \( n \geq 312 \), then \( G \) has the structure \( T_4 \) depicted in Figure 3.

(vi) If \( n \equiv 5 \) (mod 7) and \( n \geq 117 \), then \( G \) has the structure \( T_5 \) depicted in Figure 3.

(vii) If \( n \equiv 6 \) (mod 7) and \( n \geq 62 \), then \( G \) has the structure \( T_6 \) depicted in Figure 3.

The obtained computational results also indicate the following extension of the Proposition 3.2.
Case $n \equiv 0 \pmod{7}$

$n = 7, 14, 21, 28, 161, 168$

$n = 7 - 1$

$35 \leq n \leq 154$

$n = 161, 168$

$n = \left\lceil \frac{n}{7} \right\rceil - 3$

for $175 \leq n \leq 1995$

see the configuration $T_0$ in Figure 3

Case $n \equiv 1 \pmod{7}$

$n = 8, 15, 22, 29, 36$

$n = 8 - 2$

$44 \leq n \leq 163$

$n = 43, 50, 57$

$n = \left\lceil \frac{n}{7} \right\rceil - 6$

for $64 \leq n \leq 1996$

see the configuration $T_1$ in Figure 3

Case $n \equiv 2 \pmod{7}$

$n = 9, 16, 23, 30$

$n = 9 - 2$

$44 \leq n \leq 163$

$n = 37$

$n = \left\lceil \frac{n}{7} \right\rceil - 3$

for $1185 \leq n \leq 1997$

see the configuration $T_2$ in Figure 3

$n = 170 \leq n \leq 1178$

$n = \left\lceil \frac{n}{7} \right\rceil - 2$

Figure 1: Trees of order $n$, $7 \leq n \leq 2001$, with minimal ABC index obtained by computer search - cases $n \equiv 0, 1, 2 \pmod{7}$.
Case $n \equiv 3 \pmod{7}$

$n = 10$ \hspace{0.5cm} $n = 17$ \hspace{0.5cm} $n = 24$ \hspace{0.5cm} $n = 31$ \hspace{0.5cm} $38 \leq n \leq 73$

for $80 \leq n \leq 1998$ see the configuration $T_3$ in Figure 3

Case $n \equiv 4 \pmod{7}$

$n = 11$ \hspace{0.5cm} $n = 18, 25, 32, 39$ \hspace{0.5cm} $n = 46$ \hspace{0.5cm} $53 \leq n \leq 305$

for $312 \leq n \leq 1999$ see the configuration $T_4$ in Figure 3

Case $n \equiv 5 \pmod{7}$

$n = 12$ \hspace{0.5cm} $n = 19$ \hspace{0.5cm} $n = 26$ \hspace{0.5cm} $33 \leq n \leq 110$

for $117 \leq n \leq 2000$ see the configuration $T_5$ in Figure 3

Case $n \equiv 6 \pmod{7}$

$n = 13$ \hspace{0.5cm} $n = 20$ \hspace{0.5cm} $n = 27, 34$ \hspace{0.5cm} $n = 41$ \hspace{0.5cm} $n = 48$

$n = 55$

for $62 \leq n \leq 2001$ see the configuration $T_6$ in Figure 3

Figure 2: Trees of order $n$, $7 \leq n \leq 2001$, with minimal ABC index obtained by computer search - cases $n \equiv 3, 4, 5, 6 \pmod{7}$. 
Conjecture 3.2. A minimal ABC tree of order $n > 1178$ does not contain a pendant path of length three.

We would like to note that our computations show only minor violation of the assumption about the central vertex. This strengthen the already existing believe, supported also by the computational model in [15], that the minimal ABC tree is unique, for trees of order larger than 168.

4 Acknowledgment

The author would like to thank Boris Furtula for presenting the problem of characterizing graphs with minimal atom-bond connectivity index and sharing initial information about it.
References

[1] D. Avis, K. Fukuda, Reverse search for enumeration, Discrete Appl. Math. 2 (1996) 21–46.

[2] A. Bondy, U. S. R. Murty, Graph Theory, Springer Verlag, Berlin, 2008.

[3] J. Chen, X. Guo, Extreme atom-bond connectivity index of graphs, MATCH Commun. Math. Comput. Chem. 65 (2011) 713–722.

[4] J. Chen, X. Guo, The atom-bond connectivity index of chemical bicyclic graphs, Appl. Math. J. Chinese Univ. 27 (2012) 243–252.

[5] J. Chen, J. Liu, X. Guo, Some upper bounds for the atom-bond connectivity index of graphs, Appl. Math. Lett. 25 (2012) 1077–1081.

[6] K. C. Das, Atom-bond connectivity index of graphs, Discrete Appl. Math. 158 (2010) 1181–1188.

[7] K. C. Das, I. Gutman, B. Furtula, On atom-bond connectivity index, Chem. Phys. Lett. 511 (2011) 452–454.

[8] K. C. Das, I. Gutman, B. Furtula, On atom-bond connectivity index, Filomat 26 (2012) 733–738.

[9] K. C. Das, N. Trinajstić, Comparison between first geometric-arithmetic index and atom-bond connectivity index, Chem. Phys. Lett. 497 (2010) 149–151.

[10] J. Devillers, A T. Balaban (Eds.), Topological Indices and Related Descriptors in QSAR and QSPR, Wiley–VCH, Gordon and Breach, Amsterdam, 1999.

[11] P. Erdös, T. Gallai, Graphs with prescribed degrees of vertices, (in Hungarian) Mat. Lapok. 11 (1960) 264–274.

[12] E. Estrada, Atom-bond connectivity and the energetic of branched alkanes, Chem. Phys. Lett. 463 (2008) 422–425.

[13] E. Estrada, L. Torres, L. Rodriguez, I. Gutman, An atom-bond connectivity index: Modelling the enthalpy of formation of alkanes, Indian J. Chem. 37A (1998) 849–855.

[14] B. Furtula, A. Graovac, D. Vukičević, Atom-bond connectivity index of trees, Discrete Appl. Math. 157 (2009) 2828–2835.

[15] B. Furtula, I. Gutman, M. Ivanović, D. Vukičević, Computer search for trees with minimal ABC index, Appl. Math. Comput. 219 (2012) 767–772.

[16] L. Gao, B. Liu, Z. You, The ABC index of trees with given degree sequence, MATCH Commun. Math. Comput. Chem. 68 (2012) 137–145.

[17] A. Graovac, M. Ghorbani, A new version of the atom-bond connectivity index, Acta Chim. Slov. 57 (2010) 609–612.

[18] I. Gutman, B. Furtula, Trees with smallest atom-bond connectivity index, MATCH Commun. Math. Comput. Chem. 68 (2012) 131–136.

[19] I. Gutman, B. Furtula, M. Ivanović, Notes on trees with minimal atom-bond connectivity index, MATCH Commun. Math. Comput. Chem. 67 (2012) 467–482.
[20] I. Gutman, J. Tošović, S. Radenković, S. Marković, *On atom-bond connectivity index and its chemical applicability*, Indian J. Chem. **51**A (2012) 690–694.

[21] A. Iványi, L. Lucz, T. F. Móri, P. Só tér, *On Erdős-Gallai and Havel-Hakimi algorithms*, Acta Univ. Sapientiae, Inform. **3** (2011) 230–268.

[22] W. Lin, X. Lin, T. Gao, X. Wu, *Proving a conjecture of Gutman concerning trees with minimal ABC index*, MATCH Commun. Math. Comput. Chem. **69** (2013), in press.

[23] F. Ruskey, R. Cohen, P. Eades, A. Scott, *Alley CATs in search of good homes*, Congr. Numer. **102** (1994) 97–110.

[24] N. J. A. Sloane, *Sequence A004251/M1250*, The on-line encyclopedia of integer sequences.

[25] N. J. A. Sloane, *Sequence A000055*, The on-line encyclopedia of integer sequences.

[26] R. Todeschini, V. Consonni, *Molecular Descriptors for Chemoinformatics*, Wiley–VCH, Weinheim, 2009.

[27] T. S. Vassilev, L. J. Huntington, *On the minimum ABC index of chemical trees*, Appl. Math. **2** (2012) 8–16.

[28] H. Wang, *Extremal trees with given degree sequence for the Randić index*, Discrete Math. **308** (2008) 3407–3411.

[29] R. Xing, B. Zhou, *Extremal trees with fixed degree sequence for atom-bond connectivity index*, Filomat **26** (2012) 683–688.

[30] R. Xing, B. Zhou, F. Dong, *On atom-bond connectivity index of connected graphs*, Discrete Appl. Math. **159** (2011) 1617–1630.

[31] R. Xing, B. Zhou, Z. Du, *Further results on atom-bond connectivity index of trees*, Discrete Appl. Math. **158** (2011) 1536–1545.