MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **79** (2018) 431-450

ISSN 0340 - 6253

Efficient Computation of Trees with Minimal Atom-Bond Connectivity Index Revisited

Darko Dimitrov[*]

Hochschule für Technik und Wirtschaft Berlin, Germany &
Faculty of Information Studies, Novo mesto, Slovenia
e-mail: `darko.dimitrov11@gmail.com`

Nikola Milosavljević

University of Niš, Faculty of Sciences and Mathematics, Serbia
e-mail: `nikola5000@gmail.com`

(Received June 29, 2017)

Abstract

The atom-bond connectivity (ABC) index is a vertex-degree-based graph invariant that found 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. Here, we show several new properties of the degree sequences of the trees with minimal ABC index. We exploit them and some recently proven results of the structure of the minimal-ABC trees to improve the algorithm based on the degree sequence [13,38]. The evaluation of the new algorithm shows that it is significantly faster than the known algorithms for identifying the trees with minimal ABC index.

[*]Corresponding author

1 Introduction

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. In 1998, Estrada et al. [24] proposed a new vertex-degree-based graph topological index, the *atom-bond connectivity (ABC) index*, defined as

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

In [24] it was shown that the ABC index can be a valuable predictive tool in the study of the heat of formation in alkanes. Several years later in [23] Estrada elaborated a novel quantum-theory-like justification for this topological index, which initiated a lot of interest both in mathematical and chemical research communities. The physico-chemical applicability of the ABC index was confirmed and extended also in several other studies [8, 32, 36, 47]. In addition, numerous results and structural properties of ABC index were established [5–7, 9–12, 17, 20, 21, 25, 28–31, 39, 40, 44–46].

In [10] it was proven that adding an edge in a graph strictly increases its ABC index (equivalently in [5] it was shown that deleting an edge in a graph strictly decreases its ABC index). This fact has two immediate consequences. Firstly, among all connected graphs with n vertices, the complete graph K_n has maximal value of ABC index. Secondly, among all connected graphs with n vertices, the graph with minimal ABC index is a tree.

In [25] it was shown that the star graph S_n is a tree with maximal ABC index. On the other hand, a complete characterization of trees with minimal ABC index (also refereed as minimal-ABC trees) is still an open problem. Computer supported search can be of enormous help towards a solution of that problem. Here, we present some new properties of trees with minimal *ABC* index. We combine them with some additional recent theoretical results to obtain a new algorithm based on degree sequences of trees, which is significantly faster than the known algorithms for identifying the trees with minimal ABC index [13, 26, 38, 41].

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.* A vertex is *big*, if its degree is at least 3 and it is not adjacent to a vertex of degree 2. As in [31], 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. If $d(v_k) > 2$, then S_k is an *internal path* of length $k-1$.

A B_1-branch is a path of length 2 with one end-vertex u that has at least one child of degree at least 3. A B_k-branch, for $k \geq 2$, is a (sub)graph comprised of a vertex v of degree $k+1$ and k pendant paths of length 2 that all have v as a common vertex. We call the vertex u (resp. the vertex v) also the center of the B_1-branch (resp. of the B_k-branch, $k \geq 2$). Moreover, a B_k^*-branch, for $k \geq 1$, is a (sub)graph obtainable from B_k by attaching an additional vertex to a pendant vertex of B_k-branch. Illustrations of B_k- and B_k^*-branches are given in Figure 1.

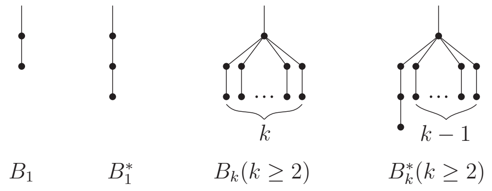


Figure 1. B_k and B_k^*-branches, $k \geq 1$.

A *k-terminal vertex* of a rooted tree is a vertex of degree $k+1 \geq 3$, which is a parent of only $B_{\geq 1}$-branches, such that at least one branch among them is a B_1-branch (or B_1^*-branch). The (sub)tree, induced by a *k-terminal vertex* and all its (direct and indirect) children vertices, is called a *k-terminal branch* or *T_k-branch.*

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.*

In [43] Wang defined a *greedy tree* as follows.

Definition 1.1 ([43]) *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 $v_2, v_3, \ldots$*

4. Repeat 3. for all newly labeled vertices, always starting with the neighbors of the labeled vertex with largest degree whose neighbors are not labeled yet.

The rest of the paper is structured as follows. In Section 2 we give an overview of the theoretical and computational results relevant to this work. The degree sequences of the minimal-ABC trees are considered in Section 3, where several new properties were shown. The new computational results and conclusional remarks are given in Section 4.

2 Related results

2.1 Related theoretical results

To determine the minimal-ABC tress of order less than 10 is an easy task, so to simplify the exposition in the rest of the paper, we assume that the trees of interest are of order at least 10.

In [31], Gutman, Furtula and Ivanović obtained the following results.

Theorem 2.1 *The n-vertex tree with minimal ABC index does not contain internal paths of any length $k \geq 1$.*

Theorem 2.2 *The n-vertex tree with minimal ABC index does not contain pendant paths of length $k \geq 4$.*

Theorem 2.3 ([31]) *The n-vertex tree with minimal ABC index contains at most one pendant path of length* 3.

An immediate, but important, consequence of Theorem 2.1 is the next corollary.

Corollary 2.4 *Let T be a tree with minimal ABC index. Then the subgraph induced by the vertices of T whose degrees are greater than two is also a tree.*

An improvement of Theorem 2.2 is the following result by Lin, Lin, Gao and Wu [40].

Theorem 2.5 *Each pendant vertex of an n-vertex tree with minimal ABC index belongs to a pendant path of length k, $2 \leq k \leq 3$.*

The following result by Gan, Liu and You [27] characterizes the trees with minimal ABC index with prescribed degree sequences. The same result, using slightly different notation and approach, was obtained by Xing and Zhou [44].

Theorem 2.6 *Given the degree sequence, the greedy tree minimizes the ABC index.*

Theorem 2.7 ([14]) *A minimal-ABC tree does not contain a B_k-branch, $k \geq 5$.*

Theorem 2.8 ([14]) *A minimal-ABC tree does not contain more than four B_4-branches.*

In the rest of this section, we mention several relevant results regarding the bounds on the number of B_k- and B_k^*-branches. Also we present some results about the forbidden configurations in the minimal-ABC trees.

Lemma 2.9 ([22]) *A minimal-ABC tree does not contain a B_k^*-branch, $k \geq 4$.*

Theorem 2.10 ([22]) *Suppose that T is a minimal-ABC tree of order $n > 18$. If T contains a pendent path of length 3, then two B_2-branches cannot be attached to the same vertex in T.*

Corollary 2.11 ([22]) *A minimal-ABC tree of order $n > 18$ that contains a pendent path of length 3 can contain at most two B_2-branches.*

Theorem 2.12 ([18]) *A minimal-ABC tree of order $n > 18$ with a pendant path of length 3 does not contain B_1-branch (B_1^*-branch).*

Theorem 2.13 ([18]) *A minimal-ABC tree of order $n > 18$ with a pendant path of length 3 may contain a B_2-branch if and only if it is of order 161 or 168. Moreover, in this case a minimal-ABC tree is comprised of single central vertex, B_3-branches and one B_2, including a pendant path of length 3 that may belong to a B_3^*-branch or B_2^*-branch.*

Theorem 2.14 ([14]) *A minimal-ABC tree does not contain more than four B_4-branches.*

Theorem 2.15 ([15]) *A minimal-ABC tree G can contain at most four B_1-branches. Moreover, if G is a T_k-branch itself, then it can contain at most three B_1-branches.*

Theorem 2.16 ([15]) *A minimal-ABC tree does not contain more than eleven B_2-branches.*

Theorem 2.17 ([16]) *A minimal ABC tree of order $n > 415$ does not contain a pendent path of length three.*

Theorem 2.18 ([19]) *A minimal-ABC tree cannot contain a B_4-branch and a B_1-branch simultaneously.*

Theorem 2.19 ([19]) *A minimal-ABC tree cannot contain a B_4-branch and a B_2-branch simultaneously.*

2.2 Related computational results

For complete characterization of the minimal-ABC trees, besides the theoretically proven properties, computer supported search can be of enormous help. Therefore, we would like to mention in the sequel few related results.

A first significant example of using computer search was done by Furtula, Gutman, Ivanović and Vukičević [26], where the trees with minimal ABC index of up to size of 31 were computed, and an initial conjecture of the general structure of the minimal-ABC trees was set. There, a brute-force approach of generating all trees of a given order, speeded up by using a distributed computing platform, was applied. The plausible structural computational model and its refined version presented there was 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 ≥ 3 induce a star graph. This assumption was shattered by counterexamples presented in [1–3, 13]. In this context, it is worth to mention that for a special class of trees, so-called *Kragujevac trees*, that are comprised of a central vertex and B_k-branches, $k \geq 1$ (see Figure 1 for an illustration), the minimal-ABC tress were fully characterized by Hosseini, Ahmadi and Gutman [35].

In [13] by considering only the degree sequences of trees and some known structural properties of the trees with minimal ABC index all trees with minimal ABC index of up to size of 300, within 15 days, were computed. The enumeration of the degree sequences of trees in [13] is related to the enumeration of the degree sequences of graphs by Ruskey et al. [42]. It is based on the Havel-Hakimi's recursive characterization of the degree sequences of grpahs [33, 34], and exploits the so called "reverse search", a term originated by Avis and Fukuda [4]. The algorithm of identifying trees with minimal ABC index (Algorithm 1), comprised of three consecutive steps is presented bellow.

Due to the nature of the recursive relation used in the first step of Algorithm 1, the same degree sequences were generated several times. That disadvantage was improved

Algorithm 1 MinABCTrees(n). Algorithm based on the degree sequences that identifies the minimal-ABC trees.

```
Input: An order $n$ of a tree
Output: A tree with the minimal ABC index
  1. Enumerate the degree sequences based on the Havel-Hakim recursive characterization,
satisfying in addition some known properties of the minimal-ABC trees.
  2. Find corresponding ‘greedy trees’ for each generated degree sequence applying
Theorem 2.6.
  3. Calculate the ABC index of each ‘greedy tree’ and select the tree with minimal value.
```

in [38], where the appropriate degree sequences were enumerated by applying an integer partitioning argument. For more information about the integer partitioning and its relation to degree sequences we refer the reader to the introduction of the next section. Together with combing the known properties of the minimal-ABC trees, the number and the length of the candidate degree sequences was reduced. Thus, in [38], using a similar single computer platform as in [13], all minimal-ABC trees of up to size of 350 within 8 days were identified.

Another advantage of applying integer partitioning for enumeration of the degree sequences is that such enumeration can be easily parallelized. In [41], the above variant of the degree sequences' based algorithm, was implemented with MPI + OpenMP, and minimal-ABC trees of up to size of 400 within 23 hours, on a workstation group with 36 CPU cores, were identified.

Here, by considering new structural properties of the minimal-ABC tree, we modify the initial degree sequence based algorithm [13] and its improvement [13, 38], obtaining a new version which is significantly faster than previous known algorithms mentioned above. We have implemented a not parallelized version and identify the minimal-ABC trees of up to size of 400 in less than 4 minutes on a single PC with 2 cores run at 2.3 GHz.

3 Degree sequences of minimal-ABC trees

The notation presented in this section is adopted from [37, 38].

A *partition* of a positive integer m is a representation of m as a sum of positive integers, say $m = d_1+d_2+\cdots+d_k$. The summands $d_1, d_2, \dots, d_k$ are called the *parts* of the partition. A partition $m = d_1 + d_2 + \cdots + d_k$ is said to be in *standard form* if $d_1 \geq d_2 \geq \cdots \geq d_k$, and can be written as a sequence $(d_1, d_2, \ldots, d_k)$. Further, we will assume here that the partitions are in standard form. By $P(m)$ we will denote the set of all partitions of m and by $P(m,k)$ the set of partitions of m having k parts. Obviously, $P(m) = \cup_{k=1}^{m} P(m,k)$. The partition numbers $p(m)$ and $p(m,k)$ denote the number of all partitions of m and the number of all partitions of m with k parts, respectively, i.e., $p(m) = |P(m)|$ and $p(m,k) = |P(m,k)|$. It will be convenient to define $p(0) = p(0,0) = 1$ and $p(m,0) = 0$, for all $m \geq 1$. Let

$$P_d(m,k) = \{(d_1, d_2, \ldots, d_k) \in P(m,k) \,|\, d_1 = d \geq d_i, i = 2, \ldots, k\}, \quad \textit{and}$$

$$P_{\geq d}(m,k) = \{(d_1, d_2, \ldots, d_k) \in P(m,k) \,|\, d_i \geq d, i = 1, \ldots, k\}.$$

Lemma 3.1 ([38]) *$(d_1, d_2, \ldots, d_k) \in P_{\geq d}(m,k) \Longleftrightarrow (d_1 - (d-1), d_2 - (d-1), \ldots, d_k - (d-1)) \in P(m - k(d-1), k), d \geq 2$.*

We say that a non-increasing positive integer sequence $D = (d_1, d_2, \ldots, d_t, d_{t+1}, \ldots, d_n)$ is *optimal*, if it is the degree sequence of a minimal-ABC tree with n vertices.

In the rest of this section, we present some new results about the degree sequences of minimal-ABC trees .

3.1 Degree sequences of minimal-ABC trees with only B_3-branches

Proposition 3.2 *Let $D = (d_1, d_2, \ldots, d_t, d_{t+1} \ldots, d_n)$ be a degree sequence of minimal-ABC tree G with n vertices that contains only B_3-branches, where $d_1, d_2, \ldots, d_t$ are degrees of big vertices and $d_{t+1} \ldots, d_n$ are degrees that belong to the vertices of the B_3-branches. Then,*

$$1 \leq t \leq \frac{n - 14 - i}{15},$$

with $i = 1$ if G contains a pendant path of length three, and $i = 0$ otherwise.

Proof: Since G is a tree, the lower bound on t is obvious. Let G has b_3 B_3-branches. Then, it holds that

$$n = t + 7b_3 + i, \qquad i = 0, 1. \tag{2}$$

By Theorem 2.6 it follows that the big vertices here have degree at least 4. Thus, we have that sum of degrees is at least $4t + 13b_3 + 2i$, or

$$2n - 2 \geq 4t + 13b_3 + 2i. \tag{3}$$

From (2) and (3) the upper bound on t follows. αa ■

Remark 3.1 *Equality (2) can be rewritten as* $b_3 = (n - t - i)/7$, $i = 0, 1$, *which for a fixed* n *has an integer solution for every 7th consecutive value of the parameter* t. *Thus, we need to consider every 7th value of* t *in the interval* $[1, (n - 14)/15]$.

Consider a degree sequence $D = (d_1, d_2, \dots, d_t, d_{t+1} \dots, d_n)$ defined as above. Let

$$D_t = (d_1, d_2, \dots, d_t),$$
$$S_t = \sum_{k=1}^{t} d_k = 2n - 2 - 13b_3 - 2i,$$

$i = 0, 1$ and $b_3 = (n-t-i)/7$, $i = 0, 1$. To identify the minimal-ABC tree(s) with n vertices and with only B_3-branches we need to generate the partition set $P(2n - 2)$, to calculate the corresponding ABC index of each partition and choose the minimal one. However, by Proposition 3.2 it suffices to consider the partitions set $P_{\geq 4}(S_t, t)$ or $P(S_t - 3t, t)$, for each t in $[1, (n - 14 - i)/15]$, that gives an integer solution of $b_3 = (n - t - i)/7$, $i = 0, 1$.

3.2 Degree sequences of minimal-ABC trees that contain B_4-branches

We exploit Lemma 2.9 and Theorems 2.14, 2.18 and 2.19 to obtain the following result.

Proposition 3.3 *Let* $D^{b_4} = (d_1, d_2, \dots, d_t, d_{t+1} \dots, d_n)$ *be a degree sequence of minimal-ABC tree* G *with* n *vertices that contains* B_3 *and* b_4 B_4*-branches, where* $d_1, d_2, \dots, d_t$ *are degrees of big vertices and* $d_{t+1} \dots, d_n$ *are degrees that belong to the vertices of the* B_3 *and* B_4*-branches. Then,*

$$1 \leq t \leq \frac{n - 14 - 2b_4}{22}, \qquad b_4 = 1, 2, 3, 4.$$

Proof: Similarly as the proof of Proposition 3.2. αa ■

Consider a degree sequence $D = (d_1, d_2, \dots, d_t, d_{t+1} \dots, d_n)$ defined as above. Let

$$D_t^{b_4} = (d_1, d_2, \dots, d_t),$$
$$S_t^{b_4} = \sum_{k=1}^{t} d_k = 2n - 2 - 17b_4 - 13b_3,$$

$b_4 = 1, 2, 3, 4$ and $b_3 = (n - t - 9b_4)/7$. To identify the minimal-ABC tree(s) with n vertices, with B_3-branches and b_4 B_4-branches, by Proposition 3.2 it suffices to consider the partitions set $P_{\geq 5}(S_t^{b_4}, t)$ or $P(S_t^{b_4} - 4t, t)$, for each t in $[1, (n-14-2b_4)/22]$, $b_4 = 1, 2, 3, 4$, that results in an integer solution of $b_3 = (n - t - 9b_4)/7$, $b_4 = 1, 2, 3, 4$.

3.3 Degree sequences of minimal-ABC trees that contain B_2 and B_1-branches

Proposition 3.4 *Let $D^{b_1,b_2} = (d_1, d_2, \ldots, d_t, d_{t+1} \ldots, d_n)$ be a degree sequence of minimal-ABC tree G with n vertices that contains B_3-branches, b_2 B_2-branches, and b_1 B_1-branches where $d_1, d_2, \ldots, d_t$ are degrees of big vertices and $d_{t+1} \ldots, d_n$ are degrees that belong to the vertices of the B_1, B_2, and B_3-branches. Then,*

$$1 \leq t \leq \frac{n - 14 + 2b_2 + 5b_1}{15}, \qquad b_1 = 0, 1, 2, 3, 4, \text{ and } b_2 = 0, 1, \ldots, 11.$$

Proof: Similarly as the proof of Proposition 3.2. αa ■

Consider a degree sequence $D^{b_1,b_2} = (d_1, d_2, \ldots, d_t, d_{t+1} \ldots, d_n)$ defined as above. Let

$$D_t^{b_1,b_2} = (d_1, d_2, \ldots, d_t),$$
$$S_t^{b_1,b_2} = \sum_{k=1}^{t} d_k = 2n - 2 - 13b_3 - 9b_2 - 3b_1,$$

$b_1 = 0, 1, 2, 3, 4$, $b_2 = 0, 1, \ldots, 11$ and $b_3 = (n - t - 5b_2 - 2b_1)/7$. To identify the minimal-ABC tree(s) with n vertices and with B_3-branches, b_2 B_2-branches, and b_1 B_1-branches, we need to consider the partitions set $P_{\geq 4}(S_t^{b_1,b_2}, t)$ or $P(S_t^{b_1,b_2} - 3t, t)$, for each t in $[1, (n - 14 + 2b_2 - 5b_1)/15]$, $b_1 = 0, 1, 2, 3, 4$, and $b_2 = 0, 1, \ldots, 11$, that gives an integer solution of $b_3 = (n - t - 5b_2 - 2b_1)/7$.

4 New computational results

The results and remarks from Subsections 2.1, 3.1, 3.2 and 3.3 lead to a modification of the first step of Algorithm 1, that significantly reduced the length and number of the possible degree sequences. The modification is presented in *EnumerateDegreeSequences(n)* procedure below.

We have implemented in C++ Algorithm 1 with EnumerateDegreeSequences(n) as its first step. The generation of partitions $P(m, k)$ was done with Algorithm 3.7:

Algorithm 2 ENUMERATEDEGREESEQUENCES(n). An algorithm that enumerate candidate degree sequences of a minimal-ABC tree of order n, based on the known properties of the minimal-ABC trees and integer partitioning.

Input: An order n of a tree
Output: Candidate degree sequences of a minimal-ABC tree of order n

```
1: if $n = 161$ or $169$ the minimal-ABC trees are given by Theorem 2.13
2: else
3:     if $n \leq 415$
4:         for $i = 0, 1$ do
5:             for $t = 1 \dots (n - 14 - i)/15$ and $(n - t - i)/7$ is integer do
6:                 $b_3 := (n - t - i)/7$
7:                 $S_t := 2n - 2 - 13b_3 - 2i$
8:                 Compute $P(S_t - 3t, t)$
9:     else
10:        for $t = 1 \dots (n - 14)/15$ and $(n - t)/7$ is integer do
11:            $b_3 := (n - t)/7$
12:            $S_t := 2n - 2 - 13b_3$
13:            Compute $P(S_t - 3t, t)$
14:    for $b_4 = 1 \dots 4$ do
15:        for $t = 1 \dots (n - 14 - 2b_4)/22$ and $(n - t - 9b_4)/7$ is integer do
16:            $b_3 := (n - t - 9b_4)/7$
17:            $S_t := 2n - 2 - 17b_4 - 13b_3$
18:            Compute $P(S_t - 4t, t)$
19:    for $b_1 = 0, 1 \dots 4$ do
20:        for $b_2 = 0, 1 \dots 11$ and $b_1 + b_2 \neq 0$ do
21:            for $t = 1 \dots (n - 14 + 2b_2 + 5b_1)/15$ and $(n - t - 5b_2 - 2b_1)/7$ is integer do
22:                $b_3 := (n - t - 5b_2 - 2b_1)/7$
23:                $S_t := 2n - 2 - 13b_3 - 9b_2 - 3b_1$
24:                Compute $P(S_t - 3t, t)$
```

PARTITIONLEXSUCCESSOR from [37]. Our implementation was run on the same platform as the initial algorithm from [13]: on 2.3 GHz Intel Core i5 processor with 4GB 1333 MHz DDR3 RAM. The comparative performance of the known search algorithms together with the algorithm presented here is given in Table 1.

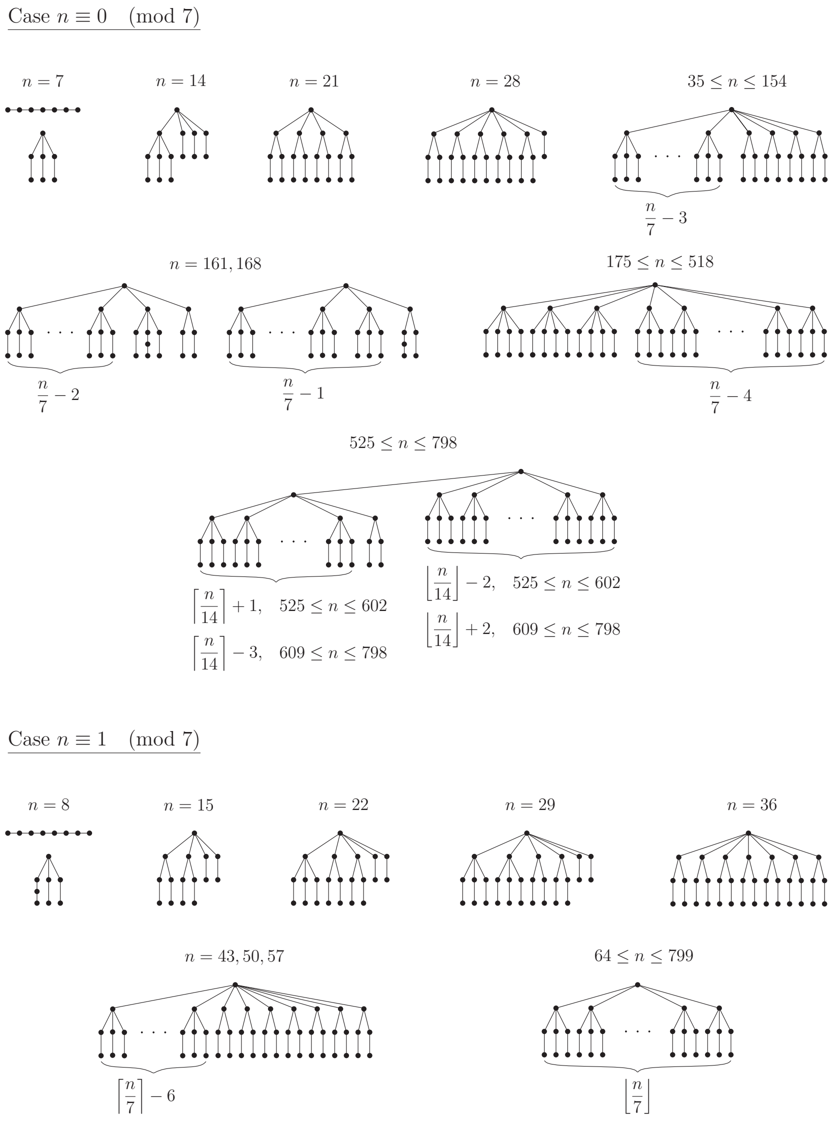
All obtained trees with minimal ABC index of order up to 800 are summarized in Figures 2, 3 and 4. For the sake of completeness, we include also the results for $7 \leq n \leq 400$, which were already obtained, partially or completely, in [13, 26, 38, 41]. For $n \leq 6$, the minimal ABC trees are paths P_n and they are omitted in the figures.

We would like to note that the algorithm presented here was run on a 6 year old personal computer. Using more modern computer configuration will certainly improve the running time. The algorithm presented here can be easily parallelized as in [41], which

Table 1. The comparison performance of the existing search algorithms. The abbreviation DS stands for degree sequence.

Algorithm	Range of n	Time (approx.)	Test platform
Brute-force search [26]	$1 \leq n \leq 31$	7 days for $n = 31$	Computer grid 400 CPUs
Original DS algorithm [13]	$1 \leq n \leq 300$	15 days	PC, 2 cores, 2.3 GHz
Modified DS algorithm [38]	$1 \leq n \leq 300$	75.5 hours	PC, 2 cores, 2.4 GHz
Parallelized modified DS algorithm [41]	$1 \leq n \leq 300$	0.21 hours	Workstation group, 36 cores
	$1 \leq n \leq 400$	23 hours	
Modified DS algorithm presented here	$n \leq 300$	13 seconds	PC, 2 cores, 2.3 GHz
	$1 \leq n \leq 400$	3.7 minutes	
	$1 \leq n \leq 700$	20 hours	
	$n = 800$	2.2 hours	
	$1 \leq n \leq 800$	7 days	

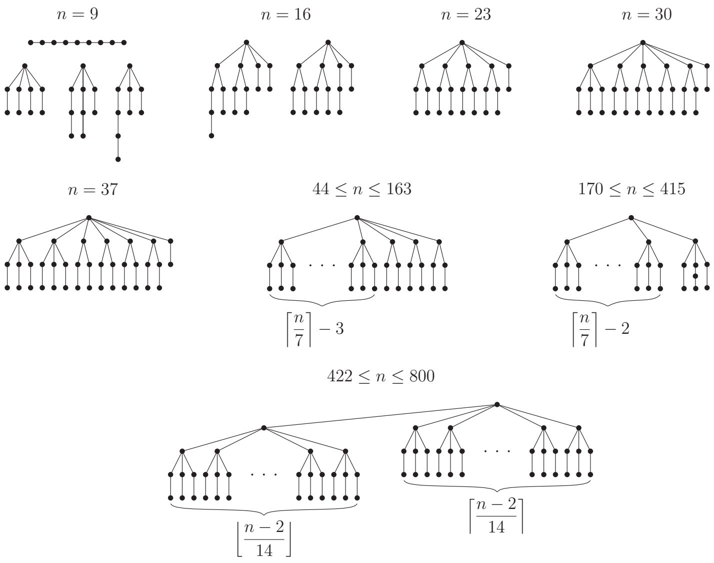
will bring more significant performance improvements. However, the new breakthrough in the computation one can expect only by incorporating further structural properties of the minimal-ABC trees and their degree sequences.



Case $n \equiv 0 \pmod{7}$
$n=7$
$n=14$
$n=21$
$n=28$
$35 \le n \le 154$
$\frac{n}{7}-3$
$n=161,168$
$\frac{n}{7}-2$
$\frac{n}{7}-1$
$175 \le n \le 518$
$\frac{n}{7}-4$
$525 \le n \le 798$
$\left\lceil\frac{n}{14}\right\rceil+1, \quad 525 \le n \le 602$
$\left\lceil\frac{n}{14}\right\rceil-3, \quad 609 \le n \le 798$
$\left\lfloor\frac{n}{14}\right\rfloor-2, \quad 525 \le n \le 602$
$\left\lfloor\frac{n}{14}\right\rfloor+2, \quad 609 \le n \le 798$
Case $n \equiv 1 \pmod{7}$
$n=8$
$n=15$
$n=22$
$n=29$
$n=36$
$n=43,50,57$
$\left\lceil\frac{n}{7}\right\rceil-6$
$64 \le n \le 799$
$\left\lfloor\frac{n}{7}\right\rfloor$

Figure 2. Trees of order n, $7 \le n \le 800$ (cases $n \equiv 0, 1 \pmod{7}$), with minimal ABC index obtained by computer search, where the candidate degree sequences were enumerated with Algorithm 2

Case $n \equiv 2 \pmod 7$


$n = 9$
$n = 16$
$n = 23$
$n = 30$
$n = 37$
$44 \le n \le 163$
$\left\lceil \frac{n}{7} \right\rceil - 3$
$170 \le n \le 415$
$\left\lceil \frac{n}{7} \right\rceil - 2$
$422 \le n \le 800$
$\left\lfloor \frac{n-2}{14} \right\rfloor$
$\left\lceil \frac{n-2}{14} \right\rceil$

Case $n \equiv 3 \pmod 7$

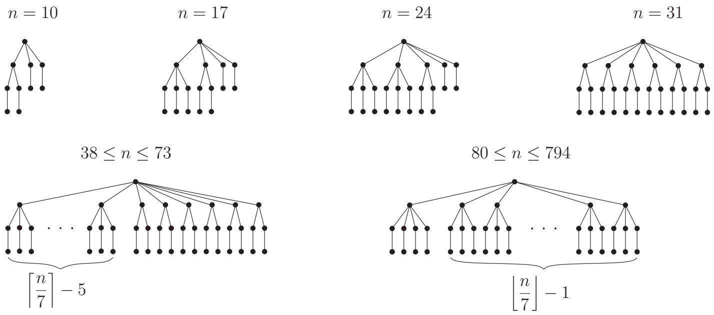
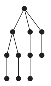

$n = 10$
$n = 17$
$n = 24$
$n = 31$
$38 \le n \le 73$
$\left\lceil \frac{n}{7} \right\rceil - 5$
$80 \le n \le 794$
$\left\lfloor \frac{n}{7} \right\rfloor - 1$

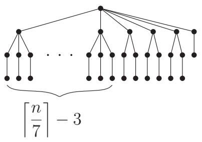
Figure 3. Trees of order n, $7 \le n \le 800$ (cases $n \equiv 2, 3 \pmod 7$), with minimal ABC index obtained by computer search, where the candidate degree sequences were enumerated with Algorithm 2.

Case $n \equiv 4 \pmod 7$

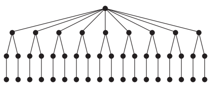
$n = 11$



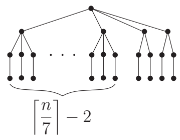
$n = 18, 25, 32, 39$



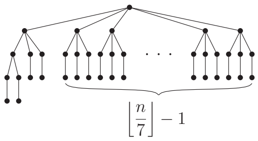
$n = 46$



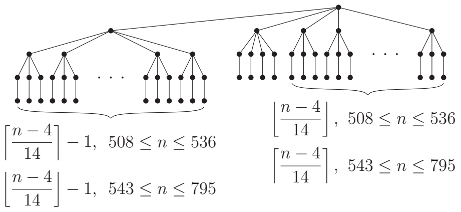
$53 \leq n \leq 305$



$312 \leq n \leq 501$

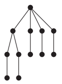


$508 \leq n \leq 795$

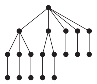

$\left\lceil \frac{n-4}{14} \right\rceil - 1,\ 508 \leq n \leq 536$
$\left\lfloor \frac{n-4}{14} \right\rfloor - 1,\ 543 \leq n \leq 795$
$\left\lfloor \frac{n-4}{14} \right\rfloor,\ 508 \leq n \leq 536$
$\left\lceil \frac{n-4}{14} \right\rceil,\ 543 \leq n \leq 795$

Case $n \equiv 5 \pmod 7$

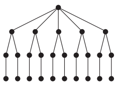
$n = 12$



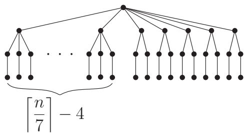
$n = 19$



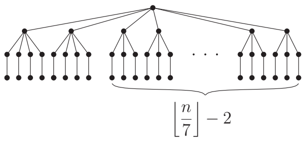
$n = 26$



$33 \leq n \leq 110$



$117 \leq n \leq 733$



$740 \leq n \leq 796$

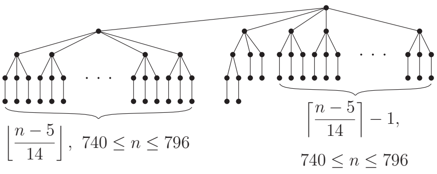
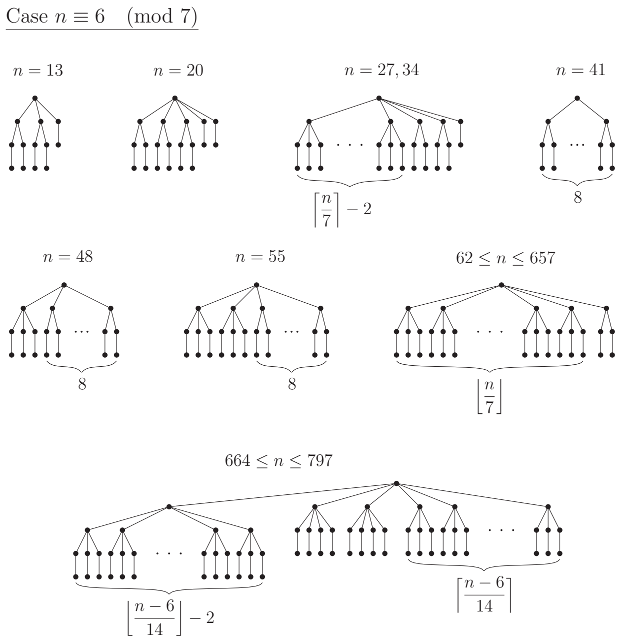

$\left\lfloor \frac{n-5}{14} \right\rfloor,\ 740 \leq n \leq 796$
$\left\lceil \frac{n-5}{14} \right\rceil - 1,$
$740 \leq n \leq 796$

Figure 4. Trees of order n, $7 \leq n \leq 800$ (cases $n \equiv 4, 5 \pmod 7$), with minimal ABC index obtained by computer search, where the candidate degree sequences were enumerated with Algorithm 2.



Case $n \equiv 6 \pmod 7$
$n = 13$
$n = 20$
$n = 27, 34$
$\left\lceil \frac{n}{7} \right\rceil - 2$
$n = 41$
8
$n = 48$
8
$n = 55$
8
$62 \leq n \leq 657$
$\left\lfloor \frac{n}{7} \right\rfloor$
$664 \leq n \leq 797$
$\left\lfloor \frac{n-6}{14} \right\rfloor - 2$
$\left\lceil \frac{n-6}{14} \right\rceil$

Figure 5. Trees of order n, $7 \leq n \leq 800$ (the case $n \equiv 6 \pmod 7$), with minimal ABC index obtained by computer search, where the candidate degree sequences were enumerated with Algorithm 2.

References

[1] M. B. Ahmadi, D. Dimitrov, I. Gutman, S. A. Hosseini, Disproving a conjecture on trees with minimal atom–bond connectivity index, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 685–698.

[2] M. B. Ahmadi, S. A. Hosseini, P. Salehi Nowbandegani, On trees with minimal atom bond connectivity index, *MATCH Commun. Math. Comput. Chem.* **69** (2013) 559–563.

[3] M. B. Ahmadi, S. A. Hosseini, M. Zarrinderakht, On large trees with minimal atom–bond connectivity index, *MATCH Commun. Math. Comput. Chem.* **69** (2013) 565–569.

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

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

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

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

[8] J. Chen, J. Liu, Q. Li, The atom–bond connectivity index of catacondensed polyomino graphs, *Discr. Dyn. Nat. Soc.* **2013** (2013) #598517.

[9] K. C. Das, Atom–bond connectivity index of graphs, *Discr. Appl. Math.* **158** (2010) 1181–1188.

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

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

[12] K. C. Das, M. A. Mohammed, I. Gutman, K. A. Atan, Comparison between atom–bond connectivity indices of graphs, *MATCH Commun. Math. Comput. Chem.* **76** (2016) 159–170.

[13] D. Dimitrov, Efficient computation of trees with minimal atom–bond connectivity index, *Appl. Math. Comput.* **224** (2013) 663–670.

[14] D. Dimitrov, On structural properties of trees with minimal atom–bond connectivity index, *Discr. Appl. Math.* **172** (2014) 28–44.

[15] D. Dimitrov, On structural properties of trees with minimal atom–bond connectivity index II: Bounds on B_1- and B_2-branches, *Discr. Appl. Math.* **204** (2016) 90–116.

[16] D. Dimitrov, On structural properties of trees with minimal atom–bond connectivity index IV: Solving a conjecture about the pendent paths of length three, *Appl. Math. Comput.* **313** (2017) 418–430.

[17] D. Dimitrov, Extremal trees with respect to the atom–bond connectivity index, in: I. Gutman, B. Furtula, K. C. Das, E. Milovanović, I. Milovanović (Eds.), *Bounds in Chemical Graph Theory – Mainstreams*, Univ. Kragujevac, Kragujevac, 2017, pp. 53–67.

[18] D. Dimitrov, Z. Du, C. M. da Fonseca, On structural properties of trees with minimal atom–bond connectivity index III: Trees with pendent paths of length three, *Appl. Math. Comput.* **282** (2016) 276–290.

[19] D. Dimitrov, Z. Du, C. M. da Fonseca, Forbidden branches in trees with minimal atom–bond connectivity index, submitted, also available at `https://arxiv.org/abs/1706.08680`.

[20] D. Dimitrov, B. Ikica, R. Škrekovski, Remarks on maximum atom–bond connectivity index with given graph parameters, *Discr. Appl. Math.* **222** (2017) 222–226.

[21] Z. Du, On the atom–bond connectivity index and radius of connected graphs, *J. Ineq. Appl.* (2015) #188.

[22] Z. Du, C. M. da Fonseca, On a family of trees with minimal atom–bond connectivity, *Discr. Appl. Math.* **202** (2016) 37–49.

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

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

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

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

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

[28] Y. Gao, Y. Shao, The smallest *ABC* index of trees with n pendent vertices, *MATCH Commun. Math. Comput. Chem.* **76** (2016) 141–158.

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

[30] I. Gutman, B. Furtula, M. B. Ahmadi, S. A. Hosseini, P. Salehi Nowbandegani, M. Zarrinderakht, The *ABC* index conundrum, *Filomat* **27** (2013) 1075–1083.

[31] I. Gutman, B. Furtula, M. Ivanović, Notes on trees with minimal atom–bond connectivity index, *MATCH Commun. Math. Comput. Chem.* **67** (2012) 467–482.

[32] I. Gutman, J. Tošović, S. Radenković, S. Marković, On atom–bond connectivity index and its chemical applicability, *Indian J. Chem.* **51A** (2012) 690–694.

[33] S. L. Hakimi, On the realizability of a set of integers as degrees of the vertices of a simple graph, *J. SIAM Appl. Math.* **10** (1962) 496–506.

[34] V. Havel, A remark on the existence of finite graphs, *Časopis Pěst. Mat.* **80** (1955) 477–480 in Czech.

[35] S. A. Hosseini, M. B. Ahmadi, I. Gutman, Kragujevac trees with minimal atom–bond connectivity index, *MATCH Commun. Math. Comput. Chem.* **71** (2014) 5–20.

[36] X. Ke, Atom–bond connectivity index of benzenoid systems and fluoranthene congeners, *Polycyc. Arom. Comp.* **32** (2012) 27–35.

[37] D. L. Kreher, D. R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press, Boca Raton, 1998.

[38] W. Lin, J. Chen, Q. Chen, T. Gao, X. Lin, B. Cai, Fast computer search for trees with minimal *ABC* index based on tree degree sequences, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 699–708.

[39] W. Lin, J. Chen, C. Ma, Y. Zhang, J. Chen, D. Zhang, F. Jia, On trees with minimal ABC index among trees with given number of leaves, *MATCH Commun. Math. Comput. Chem.* **76** (2016) 131–140.

[40] 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) 549–557.

[41] W. Lin, C. Ma, Q. Chen, J. Chen, T. Gao, B. Cai, Parallel search trees with minimal *ABC* index with MPI + OpenMP, *MATCH Commun. Math. Comput. Chem.* **73** (2015) 337–343.

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

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

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

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

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

[47] J. Yang, F. Xia, H. Cheng, The atom–bond connectivity index of benzenoid systems and phenylenes, *Int. Math. Forum* **6** (2011) 2001–2005.