Open J. Discret. Appl. Math., Vol. 1(2018), No. 2, pp. 01 - 09 Website: https://pisrt.org/psr-press/journals/odam/ ISSN: 2617-9687 (Online) 2617-9679 (Print) http://dx.doi.org/10.30538/psrp-odam2018.0005

# CHARACTERIZING TREES WITH MINIMAL ABC INDEX WITH COMPUTER SEARCH: A SHORT SURVEY

YIMING ZHENG, WENSHUI LIN, QI'AN CHEN<sup>1</sup>, LINSHAN HUANG, ZHIXI WU

ABSTRACT. The atom-bond connectivity (ABC) index of a graph G = (V, E) is defined as  $ABC(G) = \sum_{v_i v_j \in E} \sqrt{(d_i + d_j - 2)/(d_i d_j)}$ , where  $d_i$  denotes the degree of vertex  $v_i$  of G. Due to its interesting applications in chemistry, this molecular structure descriptor has become one of the most actively studied vertex-degree-based graph invariants. Many efforts were made towards the elementary problem of characterizing tree(s) with minimal ABC index, which remains open and was coined as the "ABC index conundrum". Up to date, quite a few significant results have been obtained. In the course of research computer search plays a non-negligible role. In the present paper we review the state of the art of the problem. In addition we intend to demonstrate that, repeating the procedure "searching - conjecturing - proving" can be an applicable paradigm to cope with elusive problems of extremal graph characterization.

Mathematics Subject Classification: Primary: 05C07; Secondary: 05C90. *Key words and phrases:* Atom-bond connectivity index; trees; xxtremal graph; computer search.

#### 1. Introduction

We consider non-trivial connected simple graphs only. Such a graph will be denoted by G = (V, E), where  $V = \{v_0, v_1, \dots, v_{n-1}\}$  and are the vertex set and edge set of, respectively. If  $v_i v_j \in E$ , then  $G - v_i v_j$  will denote the graph obtained from G by deleting the edge  $v_i v_j$ . If, in turn,  $v_i v_j \notin E$ , then  $G + v_i v_j$  will denote the graph obtained from G by adding the edge  $v_i v_j$ .

Received 16-08-2108. Accepted 10-11-2018.

<sup>&</sup>lt;sup>1</sup> Corresponding Author

<sup>© 2018</sup> Yiming Zheng, Wenshui Lin, Qi'an Chen, Linshan Huang, Zhixi Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Let  $P = u_0 u_1 \cdots u_{k-1} u_k$  be a path of length k in graph G with  $k \ge 1$ ,  $d(u_0) \ge 3$ ,  $d(u_k) \ne 2$ , and  $d(u_1) = d(u_2) = \cdots = d(u_{k-1}) = 2$ . If  $d(u_k) \ge 3$  then P is said to be an internal path of G. If  $d(u_k) = 1$  then P is said to be a pendent path of G.

Let  $d_i = d(v_i)$  be the degree of  $v_i$ , and  $\Delta = \Delta(G)$  the maximum degree of G. A vertex of degree 1 is called a pendent vertex.  $\pi(G) = (d_0, d_1, \dots, d_{n-1})$  is called the degree sequence of G. Given a positive integer sequence  $\pi = (d_0, d_1, \dots, d_{n-1})$ , if there exists a connected graph G with  $\pi(G) = \pi$ , then  $\pi$  is said to be a (graphic) degree sequence. In particular, if G is a tree, then  $\pi$  is called a tree degree sequence. Let  $\mathcal{C}(\pi) = \{G|G \text{ is connected and } \pi(G) = \pi\}$ , and  $\mathcal{T}(\pi) = \{T|T \text{ is a tree and } \pi(T) = \pi\}$ .

The ABC index of graph G = (V, E) is defined [1] as

$$ABC(G) = \sum_{v_i v_j \in E} \sqrt{(d_i + d_j - 2)/(d_i d_j)}.$$

This vertex-degree-based topological index turned out to be closely correlated with the heat of formation of alkanes [1], and a quantum-chemical explanation for its descriptive ability was provided in [2]. Gutman *et al.* [3] later confirmed that the *ABC* index could reproduce the heat of formation with accuracy comparable to that of high-level ab initio and DFT (MP2, B3LYP) quantum chemical calculations. Recently, a probabilistic interpretation of the generalized *ABC* index is provided by Estrada [4]. Due to these applications, there is an increased interest in the mathematical properties of the *ABC* index in the last few years (See [5–36]).

From a mathematical point of view, the first question that needs to answer is for which graphs this index assumes minimal and maximal values. It was proved [5,6] that, the ABC index of a graph strictly increases with addition of edges. Hence among *n*-vertex connected graphs, the complete graph  $K_n$  uniquely maximizes the ABC index, and a graph with minimal ABC index is a tree. In [7] Furtula proved that, among *n*-vertex trees the star  $S_n$  uniquely maximizes the ABC index. Xing *et al.* [8] found some upper bounds for the ABC index in some classes of trees. However, the problem of characterizing n-vertex tree(s) with minimal ABC index is more elusive and remains open. In [9] Gutman et al. summarized the known results and coined the problem as the "ABC index conundrum". After [9] there are quite a few significant developments, in both mathematical and computational aspects. In the course of research computer search plays a non-negligible role. Namely, the research paradigm of repeating the procedure "Searching - Conjecturing - Proving" was applied: (a) Computer search for trees with minimal ABC index of order as large as possible by using their known properties; (b) Conjecture their properties based on the search results; (c) Prove or disprove the conjectures; (d) Go to (a). In the present paper, we will review the state of the art of the problem, as an update of [9]. In addition we intend to demonstrate that, the paradigm may be applicable to cope with other elusive problems of extremal graph characterization.



FIGURE 1. The branches in a min-ABC tree.

For convenience, in the rest of the paper, we refer a tree with minimal ABC index as a min-ABC tree, and the problem of characterizing *n*-vertex tree(s) with minimal ABC index as min-ABC tree problem. We also assume  $n \ge 10$ .

### 2. A brute-force and a heuristic search, and the modulo 7 conjecture

In order to guess the general structure of min-ABC trees, Furtula *et al.* [10] firstly conducted a brute-force computer search. Their algorithm consists of two successive steps: (1) Generating the trees recursively; (2) Computing the ABC index for each generated tree and find its minimum value. Since the number of *n*-vertex trees increases rapidly with *n*, though a computer grid with 400 CPUs was employed, the computation was just performed up to n = 31. One can refer to [11] for the search results.

Few as the search results obtained in [10] are, some structural properties of min-ABC trees were observed and proved by Gutman *et al.* in [11].

### **Lemma 2.1.** [12] Let T be an n-vertex min-ABC tree. Then

- (1) T has no internal paths of length  $\geq 2$ ;
- (2) T has no pendent paths of length  $\geq 4$ ;
- (3) T has at most one pendent path of length 3.

It was also conjectured [12] that, such a tree has no pendent paths of length 1. Soon this was confirmed by Lin *et al.* [13].

**Lemma 2.2.** [13] Let T be an n-vertex min-ABC tree. Then each pendent path of T is of length 2 or 3.

Lemma 2.2 reveals that, each vertex of degree 1 or 2 of an *n*-vertex min-ABC tree T is contained in a so-called  $B_k$ - or  $B_k^*$ -branch (shown in Figure 1). From Lemma 2.1 (3) T has a unique  $B_k^*$ -branch if T has a pendent path of length 3. Based on these facts Gutman and Furtula [14] implicitly made the following conjecture.

Conjecture 2.3. Let T be an n-vertex min-ABC tree. Then

- (1) T has a single high-degree vertex  $v_0$ ;
- (2) To  $v_0$  only  $B_k$  or  $B_k^*$ -branches,  $1 \le k \le 5$ , are attached.



FIGURE 2. The counterexample from [15].

Actually, Conjecture 2.3 is somehow natural. Let T be an n-vertex min-ABC tree, and  $T[\delta]$  the subgraph of T induced by the vertices of degree at least  $\delta$ . From Lemma 2.1 (2), T[3] is connected and thus is a tree. Moreover, based on the known search results at that time, T[3] is conjectured to be a star (see [13]). With the priori assumptions in Conjecture 2.3, Gutman and Furtula [14] conducted a heuristic incomplete computer search for n-vertex min-ABC tree(s) up to n = 700. This is an easy task, because the key is to find the solution set of the Diophantine equation  $n = 1 + 2x_1 + 5x_2 + 7x_3 + 9x_4 + 11x_5 + x_6$ , where  $x_i \ge 0$  denotes the number of  $B_k$ -branches,  $1 \le i \le 5$ , and  $x_6 \in \{0, 1\}$  is the number of pendent paths of length 3.

The output of this heuristic search shows that, when n is sufficiently large (e.g.,  $n \ge 175$ ) the structure of n-vertex min-ABC trees present a peculiar modulo 7 regularity. Therefore, the so-called "modulo 7 conjecture" was proposed [14]. Unfortunately, soon later this plausible conjecture was disproved by Ahmad *et al.* [15,16]. In particular, the counterexample (shown in Figure 2) provided in [15] violates Conjecture 2.3, and indicates the existence of the so-called  $B_3^{**}$ -branches (see Figure 1) in a min-ABC tree.

On the other hand, since the trees considered in [14] as candidates with minimal ABC index, have an interesting structure, they were named Kragujevac trees in [17]. The modulo 7 conjecture with slight corrections, was shown to be valid for Kragujevac trees.

### 3. Greedy trees and search based on tree degree sequence

Gan *et al.* [18] and Xing *et al.* [19] independently proved that, the so-called "greedy tree" minimizes the *ABC* index in  $T(\pi)$ . Soon later, Lin *et al.* [20] generalized this result to connected graphs. One can refer to [20] for the definitions of greedy trees and BFS-graphs.

**Lemma 3.1.** [20] Let  $\pi$  be a degree sequence. Then there exists a BFS-graph  $G^*$  with minimal ABC index in  $C(\pi)$ .

Note that, given the degree sequence of a greedy tree (or equivalently, BFS-tree) T, ABC(T) can be easily computed. With Lemma 3.1 computer search for min-ABC trees can be done by enumerating degree sequences of trees. Dimitrov [21] presented the first such algorithm, which consists of three successive steps.

Characterizing trees with minimal ABC index with computer search: A short survey

5

- 1. Generate all tree degree sequences (recursively);
- 2. Find corresponding greedy trees for each generated degree sequence;

3. Calculate the ABC index of each greedy tree and select the tree with minimal value.

Comparing with the brute-force algorithm presented in [10], Dimitrov's is much more superior. It avoids generating and storing all *n*-vertex trees, and just needs to generate degree sequences of *n*-vertex trees. The search space is far less. For example, for n = 31, there are 40,330,829,030 trees, but only 4565 tree degree sequences. Hence Dimitrov's algorithm is able to find all *n*-vertex min-*ABC* trees for  $n \leq 300$  on a single processor platform in about 15 days.

Later Dimitrov's algorithm was improved by Lin *et al.* [22,23]. For convenience, we say a non-increasing positive integer sequence  $\pi = (d_0, d_1, \dots, d_{n-1})$  is optimal, if it is the degree sequence of a min-*ABC* tree. [22] and [23] obtained some features of an optimal tree degree sequence, which can be used to significantly narrow the search space. As reported in [22], for n = 31 only 49 (about 1%) tree degree sequences have to be generated. The MPI+OpenMP implement in [23] founds all *n*-vertex min-*ABC* tree(s) up to n = 400 in 23 hours on a workstation group with 36 CPU cores. It is worth to remark that, the search results in [22] for the first time disprove Conjecture 2.3, and confirm the existence of  $B_3^{**}$ -branches in a min-*ABC* tree.

Based on his search result, Dimitrov modified the modulo 7 conjecture initially proposed in [14]. The modified conjecture is valid for  $n \leq 400$ . However, this plausible conjecture was shown to be completely false for sufficiently large n by Ahmadi *et al.* [24]. Hence a much more efficient search algorithm or implementation is still desired to identify large min-*ABC* trees.

## 4. On branches and search up to order 1100

Let  $\pi = (\Delta = d_0, d_1, \dots, d_t, d_{t+1}, \dots, d_{n-1})$   $(d_t \ge 3 \text{ and } d_{t+1} \le 2)$  be the (non-increasing) degree sequence of a min-*ABC* tree *T*,  $n_k$  denotes the number of *k*'s among  $\{d_1, d_2, \dots, d_{n-1}\}$ , and  $d = \sum_{i=1}^t d_i/t$ .  $\#P_3 = (n-t-1) \mod 2$  indicates the number of paths of length 3 in *T*.

Since  $B_k^{(*)}$ -branches ( $B_k$ - or  $B_k^*$ -branches) are the main structure of a min-*ABC* tree T, it is meaningful to pay attentions to the number  $b_k$  of  $B_k^{(*)}$ -branches in T. Note please, here a  $B_3^{**}$ -branch is regarded as one  $B_2$ -branch and two  $B_1$ -branches. In fact, in 2014 Hosseini *et al.* [17] have considered this task for Kragujevac trees. For general trees, most works were done by Dimitrov *et al.* and Du *et al.* in [25–30]. We summarize the main results in Theorem 4.1.

**Theorem 4.1.** [25–30]

(1)  $b_1 \leq 2, b_2 \leq 11, b_4 \leq 4, and b_k = 0 \text{ for } k \geq 5;$ 

- (2)  $b_1b_4 = b_2b_4 = 0;$
- (3) If n > 18 and  $\#P_3 = 1$ , then  $b_1 = 0$ ,  $b_2 \le 2$ , and  $b_k = 0$  for  $k \ge 4$ ;
- (4) If n > 415, then  $\#P_3 = 0$ .

Algorithm	Range of $n$	Time	Test platform
Brute-force search [10]	$n \leq 31$		PC grid, 400 CPUs
Dimitrovs algorithm [21]	$30 \le n \le 300$	15 days	PC, 2 cores, $2.3 \text{ GHz}$
Algorithm in [22]	$30 \le n \le 350$	107.8 hours	PC, 8 cores, 1.8 GHz
Algorithm in [23]	$30 \le n \le 400$	23 hours	PC Group, 36 cores
Dimitrovs algorithm [31]	$30 \le n \le 800$		PC, 2 cores, $2.3 \text{ GHz}$
Algorithm in [11]	$30 \le n \le 1100$	207.1 hours	PC, 4 cores, $2.2 \text{ GHz}$

TABLE 1. The performance of some search algorithms.

From Theorem 4.1 the features of an optimal tree degree sequence obtained in [22] and [23] were significantly refined in [11] as following.

**Theorem 4.2.** [11]

 $\begin{array}{l} (1) \ n_1 = \lfloor (n-t-1)/2 \rfloor, \ n_2 = \lceil (n-t-1)/2 \rceil, \ n_3 = b_2 \leq 11, \ and \ n_4 \geq b_3 \geq (2t-31)/3; \\ (2) \ \#P_3 = 0 \ if \ n \geq 416, \ and \ so \ n_1 = n_2 = (n-t-1)/2; \\ (3) \ (n-9)/7 \leq t \leq (n+13)/5; \\ (4) \ 4 \leq \Delta \leq t \ and \ \Delta \leq n/7 + 3 \ if \ n \geq 40; \\ (5) \ 2\Delta + 5t \leq n+21; \\ (6) \ 4 - 77/(n-9) \leq d < 5; \end{array}$ 

By applying Theorem 4.1 Dimitrov [31] conducted a computer search for *n*-vertex min-*ABC* tree(s) up to n = 800. Soon later by Theorem 4.2 Lin *et al.* [11] presented the fastest algorithm so far. The test was performed up to n = 1100 within 9 days on a single PC. Table 1 shows the performance of the main computer search algorithms.

### 5. Further discussions

As pointed out in [11] that, the fastest algorithm is not yet polynomial time one, and still too powerless for large n (e.g., n = 5000), even a large workstation group is involved. Hence towards the complete solution of the min-ABC tree problem, at this point the main task is to find more properties of (sufficiently large) min-ABC trees. The following are some possible directions in further investigation.

(1) Refine the upper bounds of  $b_1$ ,  $b_2$ , and  $b_4$ . We guess  $b_1 = b_2 = 0$  and  $b_4 \leq 2$ .

(2) Obtain a non-trivial lower bound of  $\Delta$  on n and/or t.

(3) Investigate the behavior of  $d = \sum_{i=1}^{t} d_i/t$  with *n* increases, so as to get better bounds of *d* for large *n*. Better bounds of *d* can help us refine Theorem 4.2. On the other hand, d < 5 implies that, the number of high-degree vertices should be small. In fact, we guess the high-degree vertices induce a star.

### Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 11771362).

#### **Competing Interests**

The authors declare that they have no competing interests.

#### References

- Estrada, E., Torres, L., Rodriguez, L., & Gutman, I. (1998). An atom-bond connectivity index: modelling the enthalpy of formation of alkanes. *Indian journal of chemistry. Sect.* A: Inorganic, physical, theoretical & analytical, 37(10), 849-855.
- Estrada, E. (2008). Atombond connectivity and the energetic of branched alkanes. Chemical Physics Letters, 463(4-6), 422-425.
- Gutman, I., Tošović, J., Radenković, S., & Marković, S. (2012). On atom-bond connectivity index and its chemical applicability. *Indian J. Chem.*, 51A, 690-694.
- Estrada, E. (2017). The ABC matrix. Journal of Mathematical Chemistry, 55(4), 1021-1033.
- Chen, J., & Guo, X. (2011). Extreme atom-bond connectivity index of graphs. MATCH Commun. Math. Comput. Chem, 65(3), 713-722.
- Das, K. C., Gutman, I., & Furtula, B. (2011). On atom-bond connectivity index. Chemical Physics Letters, 511(4-6), 452-454.
- Furtula, B., Graovac, A., & Vukičević, D. (2009). Atombond connectivity index of trees. Discrete Applied Mathematics, 157(13), 2828-2835.
- Xing, R., Zhou, B., & Du, Z. (2010). Further results on atom-bond connectivity index of trees. Discrete Applied Mathematics, 158(14), 1536-1545.
- Gutman, I., Furtula, B., Ahmadi, M. B., Hosseini, S. A., Nowbandegani, P. S., & Zarrinderakht, M. (2013). The ABC index conundrum. *Filomat*, 27(6), 1075-1083.
- Furtula, B., Gutman, I., Ivanović, M., & Vukičević, D. (2012). Computer search for trees with minimal ABC index. Applied mathematics and computation, 219(2), 767-772.
- Lin, W., Chen, J., Wu, Z., Dimitrov, D., & Huang, L. (2018). Computer search for large trees with minimal ABC index. Applied Mathematics and Computation, 338, 221-230.
- Gutman, I., Furtula, B., & Ivanovic, M. (2012). Notes on trees with minimal atom-bond connectivity index. MATCH Commun. Math Comput. Chem, 67(2), 467-482.
- Lin, W., Lin, X., Gao, T., & Wu X. (2013). Proving a conjecture of Gutman concerning trees with minimal ABC index. MATCH Commun. Math. Comput. Chem., 69(20), 549-557.
- Gutman, I., & Furtula, B. (2012). Trees with smallest atom-bond connectivity index. MATCH Commun. Math. Comput. Chem., 68(1), 131-136.
- Ahmadi, M. B., Hosseini, S. A., & Nowbandegani, P. S. (2013). On trees with minimal atom bond connectivity index. MATCH Commun. Math. Comput. Chem, 69, 559-563.
- Ahmadi, M.B., Hosseini, S.A., & Zarrinderakht M. (2013). On large trees with minimal atom-bond connectivity index. MATCH Comm. Math. Comput. Chem., 69, 565-569.
- Hosseini, S.A., Ahmadi, M.B., & Gutman, I. (2014). Kragujevac trees with minimal atombond connectivity index. MATCH Commun. Math. Comput. Chem., 71, 5-20.
- Gan, L., Liu, B., & You, Z. (2012). The ABC index of trees with given degree sequence. MATCH Commun. Math Comput. Chem., 68, 137-145.
- Xing, R., & Zhou, B. (2012). Extremal trees with fixed degree sequence for atom-bond connectivity index. *Filomat*, 26, 683-688.
- Lin, W., Gao, T., Chen, Q., & Lin, X. (2013). On the atom-bond connectivity index of connected graphs with a given degree sequence. MATCH Commun. Math. Comput. Chem., 69, 571-578.

- Dimitrov, D. (2013). Efficient computation of trees with minimal atom-bond connectivity index. Appl. Math. Comput., 224, 663-670.
- 22. Lin, W., Chen, J., Chen, Q. A., Gao, T., Lin, X., & Cai, B. (2014). Fast computer search for trees with minimal ABC index based on tree degree sequences. *MATCH Commun. Math. Comput. Chem*, 72, 699-708.
- Lin, W., Ma, C., Chen, Q., Chen, J., Gao, T., & Cai, B. (2015). Parallel search trees with minimal ABC index with MPI+ OpenMP. MATCH Commun. Math. Comput. Chem., 73, 337-343.
- Ahmadi, M.B., Dimitrov, D., Gutman, I., & Hosseini, S.A. (2014). Disproving a conjecture on trees with minimal atom-bond connectivity index. *MATCH Commun. Math. Comput. Chem.*, 72, 685-698.
- Dimitrov, D. (2014). On structural properties of trees with minimal atom-bond connectivity index. Discrete Applied Mathematics, 172, 28-44.
- Dimitrov, D. (2016). On structural properties of trees with minimal atom-bond connectivity index II: Bounds on B1-and B2-branches. *Discrete Applied Mathematics*, 204, 90-116.
- Du, Z., & da Fonseca, C. M. (2016). On a family of trees with minimal atom-bond connectivity index. Discrete Applied Mathematics, 202, 37-49.
- Dimitrov, D., Du, Z., & da Fonseca, C. M. (2016). On structural properties of trees with minimal atom-bond connectivity index III: Trees with pendent paths of length three. *Applied Mathematics and Computation*, 282, 276-290.
- Dimitrov, D. (2017). On structural properties of trees with minimal atom-bond connectivity index IV: Solving a conjecture about the pendent paths of length three. Applied Mathematics and Computation, 313, 418-430.
- Dimitrov, D., Du, Z., & da Fonseca, C.M. (2018). Some forbidden combinations of branches in minimal-ABC trees. *Discrete Appl. Math.*, 236, 165-182.
- Dimitrov, D., & Milosavljević, N. (2018). Efficient computation of trees with minimal atom-bond connectivity index revisited. MATCH Commun. Math. Comput. Chem., 79, 431-450.
- Das, K.C., Elumalai, S., & Gutman, I. (2017). On ABC index of graphs. MATCH Commun. Math. Comput. Chem., 78, 459-468.
- Das, K.C. (2010). Atom-bond connectivity index of graphs. Discrete Appl. Math., 158, 1181-1188.
- Xing, R., Zhou, B., & Dong, F. (2011). On atombond connectivity index of connected graphs. Discrete Appl. Math., 159(15), 1617-1630.
- Gan, L., Hou, H., & Liu, B. (2011). Some results on atom-bond connectivity index of graphs. MATCH Commun. Math Comput. Chem., 66, 669-680.
- Chen, J., Liu, J., & Guo, X. (2012). Some upper bounds for the atom-bond connectivity index of graphs. Appl. Math. Lett., 25, 1077-1081.

#### Yiming Zheng

School of Information Science and Engineering, Xiamen University, Xiamen 361005, China. email: 451573076@qq.com

#### Wenshui Lin

Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen 361005, China and School of Information Science and Engineering, Xiamen University, Xiamen 361005, China.

email: wslin0xmu.edu.cn

#### Qi'an Chen

School of Information Science and Engineering, Xiamen University, Xiamen 361005, China. email: cheer@xmu.edu.cn

## Linshan Huang

School of Information Science and Engineering, Xiamen University, Xiamen 361005, China. email: 574986960@qq.com

# Zhixi Wu

School of Information Science and Engineering, Xiamen University, Xiamen 361005, China. email: 408763524@qq.com