MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **72** (2014) 699-708

ISSN 0340 - 6253

Fast Computer Search for Trees with Minimal ABC Index Based on Tree Degree Sequences

Wenshui Lin, Junjie Chen, Qi'an Chen*, Tianyi Gao, Xin Lin, Bingyue Cai

School of Information Science and Technology, Xiamen University, Xiamen, P. R. China

(Received January 6, 2014)

Abstract

The atom-bond connectivity (*ABC*) index of a graph $G=(V,E)$ is defined as $ABC(G)=\sum_{v_iv_j\in E}\sqrt{[d(v_i)+d(v_j)-2]/[d(v_i)d(v_j)]}$, where $d(v_i)$ denotes the degree of vertex v_i of G. This recently introduced molecular structure descriptor found interesting applications in chemistry. However, the problem of characterizing trees with minimal *ABC* index remains open. In attempts to guess the general structure of such trees, some computer search algorithms were developed. Dimitrov [*Appl. Math. Comput. 224 (2013)*] presented an algorithm based on tree degree sequences. In this paper we improve this algorithm. Our algorithm generates only less than 2% tree degree sequences, and can find all the n-vertex tree(s) with minimal *ABC* index for $n\le 350$ within 8 days. Our search results support Dimitrov's "modulo 7 conjecture" concerning trees with minimal *ABC* index, and disprove a conjecture we proposed before.

1. Introduction and notations

We consider non-trivial connected simple graphs only. Such a graph will be denoted by $G=(V,E)$, where $V=\{v_0,v_1,\cdots,v_{n-1}\}$ and $E=E(G)$ are the vertex set and edge set of G, respectively. Let $d(v_i)$ denote the *degree* of vertex v_i, and $\Delta=\Delta(G)$ the *maximum degree* of G. $\pi(G)=(d(v_0),d(v_1),\cdots,d(v_{n-1}))$ is called the *degree sequence* of G.

* Corresponding author. E-mail address: cheer@xmu.edu.cn

Given a positive integer sequence $\pi = (d_0, d_1, \cdots, d_{n-1})$, if there exists a connected simple graph G with $\pi(G) = \pi$, then π is said to be a (graphic) degree sequence. In particular, if G is a tree, then π is called a tree degree sequence. Let $\mathcal{T}(\pi) = \{T \mid T \text{ is a tree and } \pi(T) = \pi\}$.

A path $P = u_0 u_1 \cdots u_k$ of length k ($k \geq 1$) in graph G with $d(u_0) \geq 3$ is said to be a *pendent path* of G, if $d(u_1) = d(u_2) = \cdots = d(u_{k-1}) = 2$ and $d(u_k) = 1$. The length of a longest path of G is called the *diameter* of G.

Molecular descriptors have found a wide application in QSPR/QSAR studies [1]. One of the best known is the *Randić index* introduced in 1975 by Randić [2], who has shown this index to reflect molecular branching. However, many physic-chemical properties are dependent on factors rather different than branching. In order to take this into account but at the same time to keep the spirit of the Randić index, in 1998 Estrada et al. [3] proposed the *atom-bond connectivity (ABC) index*. The *ABC* index of graph $G = (V, E)$ is defined as $ABC(G) = \sum_{v_i v_j \in E} \sqrt{[d(v_i) + d(v_j) - 2] / [d(v_i) d(v_j)]}$.

In the last few years there is an increased interest in the mathematical properties of the *ABC* index (See [4-20]). However, the problem of characterizing n-vertex tree(s) with minimal *ABC* index remains open.

In attempts to guess the general structure of n-vertex tree(s) with minimal *ABC* index, Gutman et al. [21] carried out a brute-force computer search, which consists of two successive steps: (1) Generate all the n-vertex trees using a recursive scheme, and (2) compute the *ABC* index of each generated tree in order to find its minimum value. Due to the rapidly increases of the number of n-vertex trees with n, the search, even with the second step parallelized, is not feasible for $n \geq 32$. Hence, with Conjecture 1.1, an uncompleted search was tested up to $n = 700$ [22].

Conjecture 1.1 [22]**.** (a) The n-vertex ($n \geq 10$) tree(s) with minimal *ABC* index has a single high-degree vertex, v_0. (b) To the vertex v_0 only branches with 2, 5, 7, 8, 9, or 11 vertices are attached.

However, Conjecture 1.1 is not always valid. For example, one can refer to [17] for a 312-vertex counter-example. Dimitrov [23] presented a novel computer search algorithm based on tree degree sequences, which can find all n-vertex tree (s) with minimal *ABC* index for $n \leq 300$ on a single processor platform in about 15 days.

In this paper, we improve Dimitrov's algorithm, and find all n-vertex tree(s) with minimal ABC index for $n \leq 350$. By considering some features of tree degree sequences of trees with minimal ABC index, our algorithm generates only less than 2% tree degree sequences.

Before present our algorithm, we firstly introduce some notations and results from integer partition theory, which will help us to explain the superiority of our algorithm.

An *integer partition (having k parts)* of a positive integer m is a representation of m as a sum of (k) positive integers, say $m = d_0 + d_1 + \cdots + d_{k-1}$ ($k \geq 1$). The summands $d_0, d_1, \cdots, d_{k-1}$ are called the *parts* of the partition. If $d_0 \geq d_1 \geq \cdots \geq d_{k-1}$, then the partition is said to be in *standard form*, and can be written as a sequence, i.e., $(d_0, d_1, \cdots, d_{k-1})$. $P(m,k)$ will denote the set of all integer partitions of m having k parts, $P(m) = \bigcup_{k=1}^{m} P(m,k)$, $P_d(m) = \{(d_0, d_1, \cdots, d_{k-1}) \in P(m) \mid d_0 = d \geq d_i, i = 1, \cdots, k-1, 1 \leq k \leq m\}$, and $P_{\geq d}(m,k) = \{(d_0, d_1, \cdots, d_{k-1}) \in P(m,k) \mid d_i \geq d, i = 0, 1, \cdots, k-1\}$. Let $p(m,k) = |P(m,k)|$, $p(m) = |P(m)|$, $p_d(m) = |P_d(m)|$, and $p_{\geq d}(m,k) = |P_{\geq d}(m,k)|$. Note that, it is defined that $p(0) = p(0,0) = 1$, and $p(m,0) = 0$ for all $m \geq 1$.

Lemma 1.2.

(1) $(d_0, \cdots, d_{k-1}) \in P_{\geq d}(m,k) \Leftrightarrow (d_0 - d + 1, \cdots, d_{k-1} - d + 1) \in P(m - k(d-1), k)$.

(2) For fixed $k \geq 1$, $p(m,k)$ strictly increases with $m \geq k$.

Proof. (1) Immediate from the definitions of $P_{\geq d}(m,k)$ and $P(m,k)$.

(2) Let $\phi : P(m,k) \to P(m+1,k)$ such that $\phi\big((d_0, d_1, \cdots, d_{k-1})\big) = (d_0 + 1, d_1, \cdots, d_{k-1})$ for each $(d_0, d_1, \cdots, d_{k-1}) \in P(m,k)$. Then it is easily seen that, ϕ is injective but not surjective. The conclusion follows. ∎

Lemma 1.3 [24]. $p(m,d) = p_d(m)$.

Lemma 1.4. If $m \leq 2d$, then $p_d(m) = p(m-d)$.

Proof. If $m < d$, $p_d(m) = p(m-d) = 0$, and if $m = d$, $p_d(m) = p(m-d) = 1$. Hence assume $m > d$, namely $0 < m - d \leq d$. Let $\phi : P_d(m) \to P(m-d)$ such that $\phi\big((d, d_1, \cdots, d_{k-1})\big) = (d_1, \cdots, d_{k-1})$ for each $(d, d_1, \cdots, d_{k-1}) \in P_d(m)$. It is easily seen that, ϕ is a bijection, and the conclusion holds. ∎

Lemma 1.5. $p(m) > \left(p(\lfloor \frac{m}{2} \rfloor)\right)^2$ if $m \geq 2$.

Proof. $m = 2\lfloor \frac{m}{2} \rfloor$ if m is even, and $m-1 = 2\lfloor \frac{m}{2} \rfloor$ if m is odd. From Lemma 1.2 (2), it is sufficient to show the case m is even. Then the conclusion follows from the facts that, $P(\lfloor \frac{m}{2} \rfloor) \times P(\lfloor \frac{m}{2} \rfloor) \subseteq P(m)$ and partition $(m) \in P(m) - P(\lfloor \frac{m}{2} \rfloor) \times P(\lfloor \frac{m}{2} \rfloor)$. ■

Lemma 1.6 [24]. $p(m) = \Theta(e^{\pi\sqrt{2m/3}} / m)$.

One may refer to [24] for more details of integer partition theory, as well as the algorithm (Algorithm 3.7: PARTITIONLEXSUCCESSOR) which will be used in our algorithm to generate the partitions of $P(m,k)$.

2. Computer search algorithms based on tree degree sequences

Definition 2.1 [25]**.** Suppose that the degrees of the non-leaf vertices are given, the greedy tree is achieved by the following 'greedy algorithm':

(i) Label the vertex with the largest degree as v (the root);

(ii) Label the neighbors of v as $v_1, v_2, \cdots$, assign the largest degree available to them such that $d(v_1) \geq d(v_2) \geq \cdots$;

(iii) Label the neighbors of v_1 (except v) as $v_{11}, v_{12}, \cdots$ such that they take all the largest degrees available and that $d(v_{11}) \geq d(v_{12}) \geq \cdots$;

(iv) Repeat (iii) for all the newly labeled vertices, always start with the neighbors of the labeled vertex with largest degree whose neighbors are not labeled yet.

Lemma 2.2 [13,14,16]**.** Given the degree sequence π, the greedy tree $T^*(\pi)$ minimizes the ABC index in $\mathcal{T}(\pi)$.

Dimitrov's algorithm [23] consists of the following three successive steps:

(1) Generate all tree degree sequences (recursively);

(2) Find corresponding greedy tree $T^*(\pi)$ for each generated degree sequence π;

(3) Calculate $ABC(T^*(\pi))$ and select the tree(s) with minimal value.

This approach was proven to be computationally superior to the brute-force search [21], mainly because it avoids generating and storing all trees. Moreover, by considering some known structural features (summarized in Lemma 2.3) of trees with minimal *ABC* index, the number of tree degree sequences have to be generated is significantly reduced.

Lemma 2.3 [15]**.** If $n \geq 10$ and T is an n-vertex tree with minimal *ABC* index, then each pendent path of T is of length 2 or 3, and T contains at most one pendent path of length 3.

Our algorithm is similar to Dimitrov's, but generates much less and shorter sequences. To accomplish this, the following results will be used. For convenience, we say a non-increasing positive integer sequence $\pi = (d_0, d_1, \cdots, d_{n-1})$ is optimal, if it is the degree sequence of an n-vertex tree with minimal *ABC* index.

Lemma 2.4 [19]**.** Let T be a tree with minimal *ABC* index, and d an integer with $0 \leq d \leq \Delta - 1$. Then the subgraph induced by the vertices of T whose degrees are greater than d is also a tree. In particular, the subgraph induced by the vertices of degree Δ is also a tree.

Theorem 2.5. Let $n \geq 10$, and $\pi = (d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ ($d_t \geq 3$ and $d_{t+1} \leq 2$) is optimal. Let n_i denotes the number of i's in π. Then $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$, $n_1 = \lfloor \frac{n-t-1}{2} \rfloor$, and $n_2 = \lceil \frac{n-t-1}{2} \rceil$.

Proof. Let $T \in \mathcal{T}(\pi)$ with minimal *ABC* index. From Lemmas 2.3 and 2.4, it is easily seen that, $n_2 = n_1$ or $n_1 + 1$. On the other hand, by counting the number of vertices of T, we have $1 + t + n_1 + n_2 = n$. Hence $n_1 = \lfloor \frac{n-t-1}{2} \rfloor$ and $n_2 = \lceil \frac{n-t-1}{2} \rceil$. It remains to show $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$.

By counting the sum of the degree of the vertices of T, we have $2n - 2 \geq 3(t+1) + n_1 + 2n_2$. Hence $2n - 2 \geq 3(t+1) + \lfloor \frac{n-t-1}{2} \rfloor + 2\lceil \frac{n-t-1}{2} \rceil \geq 3(t+1) + 3\frac{n-t-1}{2}$, and $t \leq \lfloor \frac{n-7}{3} \rfloor$. $t \geq 1$ holds from Lemma 4.2 of [15]. The proof is thus completed. ■

We call a non-increasing positive integer sequence $\pi = (d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ where $d_t \geq 3$ and $d_{t+1} \leq 2$ is candidate, if π satisfies the conditions of Theorem 2.5. Let $\bar{\pi} = (d_0, d_1, \cdots, d_t)$, $\tilde{\pi} = (d_0 - 2, d_1 - 2, \cdots, d_t - 2)$, and $D_t = \sum_{i=0}^{t} d_i = 2n - 2 - \lfloor \frac{n-t-1}{2} \rfloor - 2\lceil \frac{n-t-1}{2} \rceil$. From Theorem 2.5 and Lemma 1.2 (1), it is easily seen that, π is optimal $\Rightarrow \pi$ is candidate $\Leftrightarrow \bar{\pi} \in P_{\geq 3}(D_t, t+1) \Leftrightarrow \tilde{\pi} \in P(D_t - 2t - 2, t+1)$. Hence, in order to identify the n-vertex tree(s) ($n \geq 10$) with minimal *ABC* index, instead of generating all the tree degrees, it is sufficient to generate the candidate ones only, or equivalently, to generate all the $\tilde{\pi}$'s derived from

$\bar{\pi} \in P_{\geq 3}(D_t, t+1)$ or $\tilde{\pi} \in P(D_t - 2t - 2, t+1)$ for each t : $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$. Thus our algorithm consists of the following three successive steps:

Step 1. Generate all $\bar{\pi} \in P_{\geq 3}(D_t, t+1)$ (or $\tilde{\pi} \in P(D_t - 2t - 2, t+1)$) for each t: $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$;

Step 2. For each $\bar{\pi}$ (or $\tilde{\pi}$) compute $ABC(T^*(\pi))$ to find its minimum value;

Step 3. Output all the trees with degree sequence π satisfying $ABC(T^*(\pi))$ is minimum.

For fixed $n \geq 10$, the number of candidate tree degree sequences of length n, denoted by $S_c(n)$, is exactly $\sum_{t=1}^{\lfloor \frac{n-7}{3} \rfloor} p(D_t - 2t - 2, t+1)$, which is much less than $S(n) = p(2n-2, n)$, the number of tree degree sequence of length n. Table 1. shows the ratios $r(n) = S_c(n)/S(n)$ for some n's: $21 \leq n \leq 140$. In fact, we have the following result:

Theorem 2.6. $\lim\limits_{n \to +\infty} r(n) = 0$, and $r(n) \leq 2\%$ if $n \geq 23$.

Proof. Since $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$, it is easily to verify that,

$$D_t - 2t - 2 = 2n - 2 - \lfloor \tfrac{n-t-1}{2} \rfloor - 2\lceil \tfrac{n-t-1}{2} \rceil - 2t - 2 \leq 2n - 4 - 3\tfrac{n-t-1}{2} - 2t = \tfrac{1}{2}(n-t-5) < \lfloor \tfrac{n}{2} \rfloor - 1.$$

Hence $S_c(n) = \sum_{t=1}^{\lfloor \frac{n-7}{3} \rfloor} p(D_t - 2t - 2, t+1) < \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor - 1} p(\lfloor \frac{n}{2} \rfloor - 1, k) = p(\lfloor \frac{n}{2} \rfloor - 1)$ from Lemma 1.2 (2). On the other hand, from Lemmas 1.3-1.5, we have

$$S(n) = p(2n-2, n) = p_n(2n-2) = p(n-2) > \left(p(\lfloor \tfrac{n}{2} \rfloor - 1)\right)^2.$$

Thus, $r(n) < 1/p(\lfloor \frac{n}{2} \rfloor - 1)$, and $\lim\limits_{n \to +\infty} r(n) = 0$ from Lemma 1.6.

It is easy to verify that, $r(23) = 15/792 < 2\%$, and $r(n) < 1/p(11) = 1/55 < 2\%$ if $n \geq 24$.

The proof is thus completed. ■

Comparing with Dimitrov's algorithm, ours have two advantages, one is, we generate much less and shorter sequences, the other is, the generation of sequences can be easily parallelized. To test our algorithm, we implemented two C programs: Program 1. single-threaded, and Program 2. $\lfloor \frac{n-7}{3} \rfloor$-threaded, a thread for each t: $1 \leq t \leq \lfloor \frac{n-7}{3} \rfloor$. Program 1 run on Platform 1 (Intel Pentium Dual E2220 @ 2.4 GHz processor with 2.4 GHz, 2 GB RAM), and cost 75.5 hours for computation for $30 \leq n \leq 300$. Program 2 run on Intel Xeon CPU E5-2403 @1.80 GHz (2 processors, 8 cores) with 32.0 GB RAM, and cost 107.8 hours for computing $301 \le n \le 350$. Note that, our Platform 1 is almost equal to the test platform of Dimitrov, Intel i5 @2.3 GHz processor with 4 GB RAM. We shortened the running time from about 15 days to 75.5 hours for $n \le 300$. However, to conduct a complete search for larger n (e.g., $n \ge 500$) needs a better algorithm, implementation, or test platform. In fact, it is not wise to run a program with too many threads on a single workstation or server, but much better performance of Program 2 can be expected if we run it on a computer cluster. We leave it a task in the future.

Table 1. The ratios $r(n) = S_c(n)/S(n)$ for some n's: $21 \le n \le 140$.

n	$S_c(n)$	$S(n)$	$S_c(n)/S(n)$	n	$S_c(n)$	$S(n)$	$S_c(n)/S(n)$
21	10	490	2.040%	34	73	8349	0.982%
22	13	627	2.070%	35	83	10143	0.870%
23	15	792	1.890%	40	155	26015	0.820%
24	17	1002	1.700%	50	487	147273	0.600%
25	21	1255	1.670%	60	1389	715220	0.330%
26	24	1575	1.520%	70	3687	3087735	0.190%
27	27	1958	1.380%	80	9203	12132164	0.120%
28	32	2436	1.310%	90	21861	44108109	0.080%
29	37	3010	1.230%	100	49819	150198136	0.050%
30	42	3718	1.130%	110	109475	483502844	0.030%
31	49	4565	1.070%	120	233083	1482074143	0.016%
32	56	5604	1.000%	130	482626	4351078600	0.011%
33	63	6842	0.920%	140	974669	12292341831	0.010%

3. Our computer search results

In short, our search results support the following modulo 7 conjecture, which was initially proposed by Gutman and Furtula [22], and modified by Dimitrov [23].

Conjecture 3.1 [23]**.** Let T be an n-vertex tree with minimal *ABC* index.

(i) If $n \equiv 0 (\bmod 7)$ and $n \geq 175$, then T is $T_0(n)$.

(ii) If $n \equiv 1 (\bmod 7)$ and $n \geq 64$, then T is $T_1(n)$.

(iii) If $n \equiv 2 (\bmod 7)$ and $n \geq 1185$, then T is $T_2(n)$.

(iv) If $n \equiv 3 (\bmod 7)$ and $n \geq 80$, then T is $T_3(n)$.

(v) If $n \equiv 4 (\bmod 7)$ and $n \geq 312$, then T is $T_4(n)$.

(vi) If $n \equiv 5 (\bmod 7)$ and $n \geq 117$, then T is $T_5(n)$.

(vii) If $n \equiv 6 (\bmod 7)$ and $n \geq 62$, then T is $T_6(n)$.

Moreover, our search validates that, the tree $T_4(n)$ uniquely minimizes the *ABC* index of n -vertex trees for $n = 312, 319, 326, 333, 340, 347$. Hence the following conjecture we proposed before is disproved, to which $T_4(312)$ is the minimum counter-example.

Conjecture 3.2 [15]**.** If $n \geq 15$ and T is an n-vertex tree with minimal *ABC* index, then the diameter of the subgraph induced by the vertices of T whose degrees are greater than 2 is equal to 2.

Acknowledgments: We would like to thank the referees for their useful suggestions.

References

[1] R. Todeschini, V. Consonni, *Handbook of Molecular Descriptors*, Wiley-VCH, Weinheim, 2000.

[2] M. Randić, On characterization of molecular branching, *J. Am. Chem. Soc.* **97** (1975) 6609–6615.

[3] 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.

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

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

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

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

[8] R. Xing, B. Zhou, Z. Du, Further results on atom-bond connectivity index of trees, *Discr. Appl. Math.* **157** (2010) 1536–1545.

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

[10] L. Gan, H. Hou, B. Liu, Some results on atom-bond connectivity index of graphs, *MATCH Commun. Math Comput. Chem.* **66** (2011) 669–680.

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

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

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

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

[15] 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.

[16] W. Lin, T. Gao, Q. Chen, X. Lin, On the atom-bond connectivity index of connected graphs with a given degree sequence, *MATCH Commun. Math. Comput. Chem.* **69** (2013) 571–578.

[17] 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.

[18] 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.

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

[20] 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.

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

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

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

[24] D. L. Kreher, D. R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press, Boca Raton, 1998, Chapter 3.

[25] H. Wang, The extremal values of the Wiener index of a tree with given degree sequence, *Discr. Appl. Math.* **156** (2008) 2647–2654.