MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **73** (2015) 337-343

ISSN 0340 - 6253

Parallel Search Trees with Minimal ABC Index with MPI + OpenMP

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

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

(Received August 19, 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. Lin et al [*MATCH Commun. Math. Comput. Chem.* **72** (2014)] presented an algorithm based on tree degree sequences, which can be parallelized easily. In this work we implement the algorithm with MPI + OpenMP, and identify all n-vertex tree(s) with minimal *ABC* index for $n\leq 400$ within 23 hours on a workstation group with 36 CPU cores.

1. Introduction and notations

We consider 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 $\mathcal{T}(\pi) = \{T \mid T$ 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 (*at* u_0), if $d(u_1) = d(u_2) = \cdots = d(u_{k-1}) = 2$ and $d(u_k) = 1$.

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, several search algorithms were developed. Gutman et al. [21] carried out a brute-force computer search and found n-vertex tree(s) with minimal *ABC* index for $n \leq 31$. Dimitrov [23] presented a search algorithm based on tree degree sequences, which was tested up to $n = 300$. Lin et al. [25] improved Dimitrov's algorithm and tested up to $n = 350$.

The current searching results all support the "modulo 7 conjecture", which was initially proposed by Gutman and Furtula [22], and modified by Dimitrov [23]. However, this plausible conjecture was recently shown to be completely false for sufficiently large n by Ahmadi et al. [26]. Hence, a much more efficient search algorithm or implementation is desired to provide trees with minimal *ABC* index of high order. In this work we implement the algorithm presented in [25] with MPI + OpenMP (see [27] and [28]), and identify all n-vertex tree(s) with minimal *ABC* index for $n \leq 400$ within 23 hours on a workstation group with 36 CPU cores.

2. Computer search algorithms based on tree degree sequences

Definition 2.1 [24]**.** 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 $v_1, v_2, \ldots$, 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 $v_{11}, v_{12}, \ldots$ 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 $\mathcal{T}(\pi)$.

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.

Theorem 2.3. Let $n \geq 10$, and $\pi = (\Delta = d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ ($d_t \geq 3$ and $d_{t+1} \leq 2$) is optimal. Let n_i denotes the number of i's in π. Then

(1) $n_1 = \lfloor \frac{n-t-1}{2} \rfloor$, and $n_2 = \lceil \frac{n-t-1}{2} \rceil$;

(2) $3 \leq \Delta \leq t \leq \lfloor \frac{n-7}{3} \rfloor$ and $\Delta \leq \lfloor \frac{n-1}{5} \rfloor$.

Proof. From Theorem 2.5 in [25], it remains to prove $\Delta \leq t$ and $\Delta \leq \lfloor \frac{n-1}{5} \rfloor$. Let $T^*(\pi)$ be the (unique) greedy tree in $\mathcal{T}(\pi)$ with $\Delta > t$. From Lemma 2.2, $T^*(\pi)$ minimizes the *ABC* index among all n-vertex trees. Since $d_\Delta \leq 2$ and $T^*(\pi)$ is greedy, $T^*(\pi)$ is a *Kragujevac tree* (defined in [20]). However, in [20] it was shown that, $\Delta = t$ should hold for a Kragujevac tree with minimal *ABC* index, contradicting $\Delta > t$.

Now $2n-2 = \sum_{i=0}^{n-1} d_i \geq \Delta + 3t + 2\lceil \frac{n-t-1}{2} \rceil + \lfloor \frac{n-t-1}{2} \rfloor \geq \Delta + 1.5t + 1.5n - 1.5 \geq 2.5\Delta + 1.5n - 1.5$, and $\Delta \leq \lfloor \frac{n-1}{5} \rfloor$ follows. ■

Dimitrov [23] presented the first search algorithm based on tree degree sequences. Lin et al. [25] improved Dimitrov's algorithm by using some features of optimal tree degree sequences (namely, Theorem 2.5 in [25]). For convenience, we call a tree degree sequence $\pi = (\Delta = d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ ($d_t \geq 3$ and $d_{t+1} \leq 2$) is *candidate*, if π satisfies the conditions of Theorem 2.3. Let $\bar{\pi} = (d_0, d_1, \cdots, d_t)$. The algorithm presented in [25] consists of the following three successive steps:

Step 1. Generate all $\bar{\pi}$'s such that $\pi = (\Delta = d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ is candidate;

Step 2. For each $\bar{\pi}$ 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.

Table 1. The performance of search algorithms.

Algorithm	Range of n	Time (approx.)	Test platform
Brute- force search [21]	$n \leq 31$	7 days for $n = 31$	Computer grid, 400 CPUs
Dimitrov's algorithm [23]	$30 \leq n \leq 300$	15 days	PC, 2 cores, 2.3 GHz
Algorithm in [25] serial implementation	$30 \leq n \leq 300$	75.5 hours	PC, 2 cores, 2.4 GHz
The present program MPI + OpenMP	$30 \leq n \leq 300$	0.21 hours	Workstation group, 36 cores
	$30 \leq n \leq 400$	23 hours	

With Theorem 2.3, we implemented the above algorithm with MPI + OpenMP, and employed a workstation group with 36 CPU cores as the test platform (2 Intel(R) Xeon(R) CPUs E5-2403 @1.80GHz (8 cores each) and 10 Intel(R) Core(TM) i3-3220 CPUs @3.30GHz (2 cores each)). Considering the easiness of implementation, the program was simply set to create total 14 processes with MPI, 2 on each of the two 8-cored CPUs, and 1 on each of the ten 2-cored CPUs. Each process deals with about $\left(\left\lfloor \frac{n-7}{3} \right\rfloor - 2\right)/14$ t's. Within a process, 8 threads are created with OpenMP, each of which deals with about $\max\left(t-2,\left\lfloor \frac{n-1}{5} \right\rfloor -2\right)/8$ Δ's. After each process (including the master process) finds the local minimum *ABC* index and corresponding tree degree sequence(s) for the t's assigned, the master process retrieves the data from other processes and output the global minimum *ABC* index and corresponding tree degree sequence(s). Though the platform is not so powerful, the performance is really surprising! Table 1 shows the superiority of our program.

3. Further discussions

The search results ($30 \le n \le 400$) of our program still support the modified modulo 7 conjecture [23]. However, to conduct a complete search for larger n (e.g., $n \ge 1000$) without high performance computing platform involved, our program is still too powerless. In sequel we intend to show the potentiality of the efficiency of our program to be further improved.

A graph G is said to contain a B_k*-branch* ($k \ge 2$), if there is an edge $uv \in E(G)$ such that the component of $G-uv$ containing v consists of k pendent paths of length 2 at v.

Lemma 3.1 [29]**.** A tree with minimal *ABC* index does not contain a B_k-branch, where $k \ge 5$.

Corollary 3.2. Let $n \ge 10$, and $\pi = (\Delta = d_0, d_1, \cdots, d_t, d_{t+1}, \cdots, d_{n-1})$ ($d_t \ge 3$ and $d_{t+1} \le 2$) is optimal. Then $d_t \le 5$.

Now, if we also consider the feature of optimal tree degree sequences in Corollary 3.2 in our implementation, the efficiency is dramatically increased by about 19%! It can be imaged that, if more features of optimal tree degree sequences are found, the efficiency of our program can be improved remarkably.

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

[25] 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.

[26] 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.

[27] http://www.mcs.anl.gov/project/mpich-high-performance-portable-implementation-mpi.

[28] http://openmp.org/wp/.

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