



Computer search for trees with minimal *ABC* index

Boris Furtula^a, Ivan Gutman^{a,*}, Miloš Ivanović^a, Damir Vukičević^b

^a Faculty of Science, University of Kragujevac, P.O. Box 60, 34000 Kragujevac, Serbia

^b Department of Mathematics, University of Split, N. Tesle 12, 21000 Split, Croatia

ARTICLE INFO

Keywords:

Atom-bond connectivity index
ABC index
 Extremal graph
 Computer search

ABSTRACT

The *ABC* index is a degree-based molecular structure descriptor, that found chemical applications. Finding the connected graph(s) of a given order whose *ABC* index is minimal is a hitherto unsolved problem, but it is known that these must be trees. In this paper, by combining mathematical arguments and computer-based modeling we establish the basic structural features of the minimum-*ABC* trees.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

The atom-bond connectivity (*ABC*) index is one of the numerous recently introduced vertex-degree-based graph invariants, believed to be capable of serving as molecular structure descriptors [1,8]. It was put forward by Estrada et al. in 1998 [2], but remained unnoticed. Ten years later, Estrada published another article on the same topic [3], which attracted the attention of colleagues and triggered a long series of mathematical investigations [4–7,9,11–17].

The physico-chemical applicability of the *ABC* index is based on the fact that there exists an excellent (linear) correlation between *ABC* and the experimental heats of formation of alkanes [2,10]. In the paper [3], Estrada established the physical basis for this correlation. In fact, in [3] a new mathematical model was put forward, capable of rationalizing (both qualitatively and quantitatively) the experimentally established regularities for the stability of linear and branched alkanes, as well as for the strain energy of cycloalkanes. In Estrada's model [3], the term $(d_i + d_j - 2)/(d_i d_j)$ in Eq. (1) is interpreted as the count of the 1,2-, 1,3-, and 1,4-interactions in the carbon-atom skeleton. These interactions play a fundamental role in the energetics of alkane molecules.

If G is a graph of order n , and if d_i is the degree (=number of first neighbors) of its i -th vertex, $i = 1, 2, \dots, n$, then

$$ABC = ABC(G) = \sum_{ij} \sqrt{\frac{d_i + d_j - 2}{d_i d_j}}, \quad (1)$$

with the summation going over all pairs of adjacent vertices.

When a new graph invariant is studied, the first question is to determine its minimal and maximal value for graphs of a given order and to characterize the respective extremal species. Whereas the finding of the n -vertex graph with maximal *ABC* index is an easy task, the structure of the connected n -vertex graph(s) with minimal *ABC* appeared to be a significantly more difficult problem [16]. In this paper we perform a computer search and use computer-aided models, aimed at bringing us closer to the solution of the problem.

Of the numerous already established properties of the *ABC* index we point out the inequality $ABC(G) > ABC(G - e)$ which holds for all edges e of any graph G [13,15]. Its immediate consequences are that

* Corresponding author.

E-mail addresses: boris.furtula@gmail.com (B. Furtula), gutman@kg.ac.rs (I. Gutman), mivanovic@kg.ac.rs (M. Ivanović), vukicevi@pmfst.hr (D. Vukičević).

- (a) among all graphs of order n , the complete graph K_n has the greatest ABC index, and
 (b) among all connected graphs of order n , the smallest ABC index is attained by one or more trees.

Recall that, trivially, the n -vertex graph with minimal ABC index (equal to zero) is the edgeless $\overline{K_n}$. In addition, the n -vertex tree with maximal ABC index is the star [4], which also is an easy result.

2. First step: a brute-force computer search

In order to get some idea about the structure of the n -vertex tree(s) with minimal ABC index, we decided to check all trees of order n , up to n as large as possible, and to single out the tree(s) with smallest ABC . It was hoped that after some n the form of the minimum- ABC tree will emerge, enabling us to formulate a sound conjecture on its general structure. Whereas the evaluating of the ABC index for any particular tree is an elementary computational task, the true problem is that with increasing value of n the number of trees rapidly increases and becomes prohibitively large. Table 1 shows how many trees would be needed to examine in order to perform our naive direct approach.

2.1. Computational details

In view of the rapidly growing number of n -vertex trees, it turns out that the task of generating trees and identifying the minimum- ABC species using a single PC/workstation/server is feasible up to $n = 25$. For larger values of n a special strategy had to be adopted.

Consider, for example, the case of $n = 29$, when 5,469,566,585 distinct trees need to be generated and their ABC -values computed. Using Python script on a modern 2.4 GHz CPU to fulfill this task, it turns out that average speed on a single core is about 200,000 trees/min. Therefore, the entire process of calculating the ABC indices would take about 19 days on a single CPU core!

For larger n -values something different had to be developed. Our algorithm for identifying the minimum- ABC tree(s) consists of two successive steps:

- (1) Generating the trees using a recursive scheme.
- (2) Computing the ABC index for each generated tree in order to find its minimum value.

According to the theoretical discussion found in [18], phase (1) is not parallelizable at all, due to the recursive nature of the algorithm used. Even with all modern distributed computing technologies and frameworks, some algorithms have inherent sequentiality and cannot be efficiently decomposed into smaller semi-independent parts having the ability to be executed concurrently.

The situation is totally different regarding phase (2), since searching inside graph spaces is an easily parallelizable task, with single point of synchronization – reduction at the very end. In the simplest scenario, the work can be shared equally among the available processors statically, before the calculation takes place. After each CPU finds the minimum ABC index inside its own tree subspace, the result is reported to the master process which is then responsible to compute the grand total. In a more complicated scenario, the work could be distributed dynamically, but in this paper, the first (static load balancing) scenario was employed, since it is more suitable for the grid computing platform.

There is another issue with splitting the entire process into phases (1) and (2) as we did. That means that after phase (1) is finished, the trees have to be stored in some format in order to be distributed to the assigned CPUs. Taking the same example of $n = 29$, using standard ASCII format, more than 400 GB of disk space would be necessary to store all trees of the mentioned order. Having in mind that such large files have to be transferred over the network to reach the appropriate CPUs in the distributed computing scheme, it is obvious that it would be a big waste of time and network resources. The solution capable of reducing storage space up to 27 times has been found in the GZIP streaming; instead of storing trees in pure ASCII sequence,

Table 1
Number of trees with n vertices.

n	No. of trees	n	No. of trees
15	7741	25	104,636,890
16	19,320	26	279,793,450
17	48,629	27	751,065,460
18	123,867	28	2,023,443,032
19	317,955	29	5,469,566,585
20	823,065	30	14,830,871,802
21	2,144,505	31	40,330,829,030
22	5,623,756	32	109,972,410,221
23	14,828,074	33	300,628,862,480
24	39,299,897	34	823,779,631,721

Download English Version:

<https://daneshyari.com/en/article/4629974>

Download Persian Version:

<https://daneshyari.com/article/4629974>

[Daneshyari.com](https://daneshyari.com)