JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 10, Number 1, 2003

Pp. 47–55

The Inverse Problems for Some Topological Indices in Combinatorial Chemistry

XUELIANG LI,[1] ZIMAO LI,[2] and LUSHENG WANG[2]

ABSTRACT

In the original paper, Goldman *et al.* (2000) launched the study of the inverse problems in combinatorial chemistry, which is closely related to the design of combinatorial libraries for drug discovery. Following their ideas, we investigate four other topological indices, i.e., the σ-index, the c-index, the Z-index, and the M_1-index, with a special emphasis on the σ-index. Like the Wiener index, these four indices are very popular in combinatorial chemistry and reflect many chemical and physical properties. We give algorithmic and analytical solutions for the inverse problems of the four indices. We also show that the SUBTREEVALUE reconstruction problem for the σ-index is NP-hard.

Key words: algorithms, drug discovery, σ-index, Z-index, and M_1-index.

1. INTRODUCTION

A topological index is a map from the set of chemical compounds represented by graphs to the set of real numbers. Experimental results show that many topological indices are closely correlated with some physicochemical characteristics. The *inverse* problem is defined as follows: given an index value, one wants to design chemical compounds (given as graphs or trees) having that index value. The inverse problem has applications in the design of combinatorial libraries for drug discovery.

In the original paper, Goldman *et al.* (2000) launched the study of the inverse problems in combinatorial chemistry, which is closely related to drug design and molecular recognitions (see Goldman *et al.*, 2000; Sheridan and Kearsley, 1995; and Venkatasubramanian *et al.*, 1995). They investigated the famous topological index, the Wiener index. Following their ideas, in this paper we investigate four other topological indices, the σ-index, the c-index, the Z-index, and the M_1-index, with a special emphasis on the σ-index. These four indices are very popular in molecular graph theory in chemistry. For details, see Merrifield and Simmons (1989), Trinajstic (1992), Rouvray (1973) and Sablić *et al.* (1992). We give algorithmic and analytical solutions for the inverse problems of the four indices. We also show that the *SUBTREEVALUE reconstruction* problem for the σ-index is NP-hard. Throughout this paper, we deal with simple graphs, and all lemmas and theorems are about connected graphs unless mentioned otherwise. For notations and terminology on graph theory, we refer to Harary (1969).

The rest of the paper is organized as follows: In Section 2, we investigate the σ-index, which composes the main body of this paper. Sections 3, 4, and 5 we consider the c-index, the Z-index, and the M_1-index, respectively.

[1]Center for Combinatorics, Nankai University, Tianjin 300071, China.
[2]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

2. THE σ-INDEX

Given a molecular graph G, the *σ-index* $\sigma(G)$ of G is defined to be the number of independent sets of all sizes, including the empty set. For example, the graph with a single vertex has σ-index value 2, and the star with n vertices has σ-index value $2^{n-1}+1$. The empty graph Φ (without any vertex) is considered as a connected graph. Obviously, $\sigma(\Phi)=1$. We can also treat Φ as a complete graph (a clique), an independent set, or a tree. The *σ-index* is also called the *independent set index*, or the *Merrifield and Simmons index* (Merrifield and Simmons, 1989).

The σ-index is one of the popular topological indices in chemistry. It is sensitive to some physical properties of molecules. In their paper, Merrifield and Simmons (1989) showed the correlation between σ-indices and boiling points.

Given a graph G and a vertex v of G, the independent sets of G can be partitioned into two parts, one is the set of all those independent sets containing v, and the other is the set of independent sets not containing v. So, we have the following.

Lemma 2.1. *The quantity $\sigma(G)=\sigma(G-v)+\sigma(G-v-v_1-v_2-\cdots-v_k)$, where $v_1, v_2, \ldots, v_k$ are the neighbors of v in G.*

Theorem 2.2. *For any integer $\sigma \geq 1$, there is a graph such that $\sigma(G)=\sigma$.*

Proof. Consider $G=K_{\sigma-1}$, the complete graph on $\sigma-1$ vertices. It is easy to see that $\sigma(K_{\sigma-1})=\sigma$. ■

Theorem 2.3. *Among all the graphs with n vertices, the complete graph K_n has the minimum value of the σ-index; whereas the star $K_{1,n-1}$ has the maximum value of the σ-index. Both of them are unique.*

Proof. It is easy to see that any graph with n vertices has a σ-index value of at least $n+1$. (Those $n+1$ independent sets are the empty set and the n singletons, one for each vertex in the graph.) Since $\sigma(K_n)=n+1$, K_n has the minimum value of the σ-index. For any graph with n vertices, if it has a pair of nonadjacent vertices, the value of its σ-index is at least $n+2$. So, K_n is the unique graph with the minimum value of the σ-index.

Now we prove that $K_{1,n-1}$ is the unique graph with the maximum value of the σ-index among connected graphs with n vertices. It is easy to see that the σ-index value of any graph G is at most the σ-index value of one of its spanning trees T; i.e., $\sigma(G)\leq\sigma(T)$, where the equality holds if and only if G is a tree.

We complete the proof by induction on n. Suppose that the star with $n-1$ vertices is the unique graph with the maximum σ-index value among all connected graphs with $n-1$ vertices. Consider a connected graph G of $n>1$ vertices with the maximum σ-index value. G must be a tree. We pick a leaf u in G and let v be its only neighbor. Then

$$\sigma(G)=\sigma(G-u)+\sigma(G-u-v).$$

Graph $G-u$ has $n-1$ vertices and graph $G-u-v$ has $n-2$ vertices. According to the induction hypothesis, $\sigma(G)$ reaches the maximum value if and only if $G-u$ is a star and $G-u-v$ is a graph with $n-2$ isolated vertices, which holds if and only if G is a star. The maximum value and uniqueness conditions are proved simultaneously. ■

Theorem 2.4. *Among all the trees with n vertices, the star $K_{1,n-1}$ has the maximum value of the σ-index; whereas the path P_n has the minimum value of the σ-index. Both of them are unique.*

Proof. Since $K_{1,n-1}$ is the unique graph with the maximum value of the σ-index among connected graphs with n vertices, it is also the unique tree with the same property.

Now we prove that P_n is the unique tree with the minimum value of the σ-index by induction on n. Suppose that the path P_{n-1} with $n-1$ vertices has the minimum value of the σ-index among all trees with $n-1$ vertices. Let T be any tree with n vertices. Pick a leaf u in T, and let v be its only neighbor. Then

$$\sigma(T) = \sigma(T-u) + \sigma(T-u-v).$$

According to the induction hypothesis, $\sigma(T)$ reaches the minimum if and only if $\sigma(T-u)$, and $\sigma(T-u-v)$ reaches the minimum if and only if $T-u$ and $T-u-v$ are paths. Hence, T is a path. The minimum and the uniqueness conditions are proved simultaneously. ∎

Computing the σ-index of a given tree T

Now, we design an algorithm to compute the σ-index for a given tree T. For computational purposes, we fix any vertex r of T as its root. Let T be a rooted tree, and $V(T)$ be the set of all vertices in T. For each $v \in V(T)$, T^v denotes the subtree of T rooted at v. Define $\sigma^v = \sigma(T^v)$. Let $v_1, v_2, \ldots, v_k$ be the children of v in T. Define $\sigma_v = \sigma^{v_1}\sigma^{v_2}\cdots\sigma^{v_k}$. Also, σ_v is the number of independent sets for T^v that do not contain vertex v. Similarly, we define $\bar{\sigma}_v = \sigma_{v_1}\sigma_{v_2}\ldots\sigma_{v_k}$. $\bar{\sigma}_v$ as the number of independent sets for T^v that contain vertex v. From Lemma 2.1, we can obtain the following.

Theorem 2.5. *Let $v_1, v_2, \ldots, v_k$ be the children of vertex v in the rooted tree T.*

(1) $\sigma^v = \sigma_v + \sigma_{v_1}\sigma_{v_2}\cdots\sigma_{v_k}$; and (2) $\sigma_v = \sigma^{v_1}\sigma^{v_2}\ldots\sigma^{v_k}$.

Based on Theorem 2.5, we design an algorithm to compute the σ-index $\sigma(T)$ for a given tree T. For each leaf v of the rooted tree T, set $\sigma^v = 2$ and $\sigma_v = 1$. (The empty set is also counted.) For each vertex $v \in V(T)$, we can use (1) and (2) to obtain the value σ^v in a bottom-up fashion. Then $\sigma(T)$ is obtained from σ^r, where r is the root of T. Figure 1 gives some examples. The number on each vertex is σ^v. The complete algorithm is given in Figure 2.

Theorem 2.6. *The running time of Algorithm 1 in Fig. 2 is $O(n)$, where n is the number of vertices in the tree T.*

Proof. Let d_v be the degree of vertex v in T. For each $v \in T$, computing σ_v requires $O(d_v)$ time. The total running time of the algorithm is then $\sum_{v\in V(T)} d_v$, which is upper bounded by $O(n)$. ∎

We implement Algorithm 1 in C language and compute all possible values of the σ-index for trees with at most 18 vertices. See www.personal.cs.cityu.edu.hk/~lizm/sigmalist.html. Our computation shows that for many integers less than 10,946, there is no corresponding tree with the σ-index value. On the other

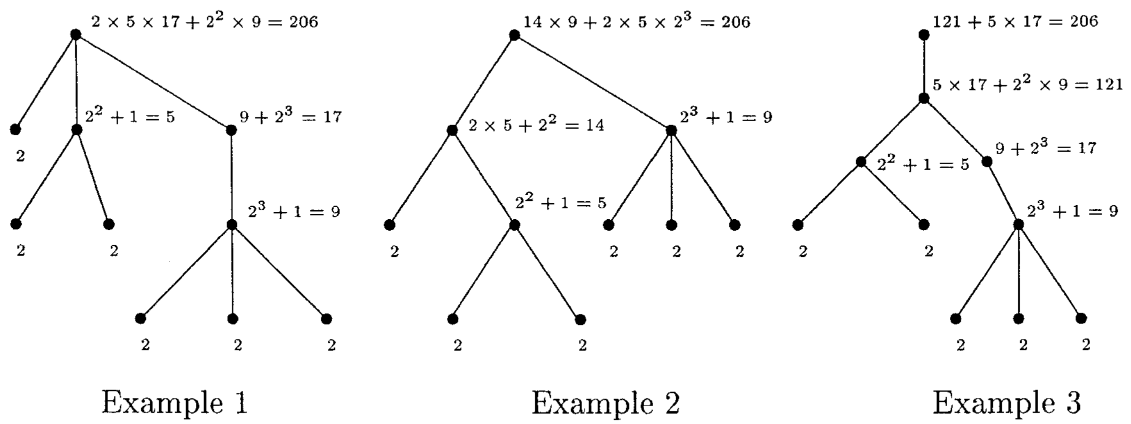

FIG. 1. Three examples.

Algorithm 1

Input A tree T with a root r.

Output The value σ^v for each $v \in V(T)$.

1. **for** each $v \in V(T)$ bottom up **do**

    compute σ^v using (1)

    compute σ_v using (2).

FIG. 2. Algorithm for computing the σ-index of a tree.

hand, as n increases, the number of trees with n vertices grows much faster than the maximum value of the σ-index for trees with n vertices. Thus, we have the following interesting problem.

Problem 2.7. *Is it true that for any large number N, there always exists a natural number $\sigma \geq N$ for which there is no tree T such that $\sigma(T) = \sigma$?*

Constructing a tree with a given σ-index

The *inverse* problem for trees is as follows: given a natural number σ, one wants to know if there is a tree T such that $\sigma(T) = \sigma$. If yes, construct the tree. Now, we give a dynamic programming algorithm to solve the problem.

From Algorithm 1, we know that for any $v \in V(T)$, we have three numbers n_1, (the number of vertices in T^v), σ_v, and σ^v. We denote them by an ordered triple $(n_1, \sigma_v, \sigma^v)$. In particular, for the tree T rooted at r we have a triple (n, σ_r, σ^r). In this way, we get a 3-D matrix M such that the entry $(n, \sigma_r, \sigma^r) = 1$, if there is a tree T rooted at r with n vertices and the other two parameters σ_r and σ^r; and 0, otherwise. The following theorem gives the recurrence equations for the dynamic programming algorithm.

Theorem 2.8. *Let T_1 and T_2 be two rooted trees rooted at u and v, respectively. T_1 and T_2 have the given triples $(n_1, \sigma_u, \sigma^u)$ and $(n_2, \sigma_v, \sigma^v)$, respectively. If T is constructed from T_1 and T_2 by connecting u and v with a new edge, then for T we have*

(1) $|V(T)| = n_1 + n_2$,
(2) $\sigma_v(T) = \sigma_v \sigma^u$ *and* $\sigma_u(T) = \sigma_u \sigma^v$,
(3) $\sigma^v(T) = \sigma_v \sigma^u + (\sigma^v - \sigma_v)\sigma_u$ *and* $\sigma^u(T) = \sigma_u \sigma^v + (\sigma^u - \sigma_u)\sigma_v$.

Note that any tree can be obtained from two small trees by connecting a vertex in one tree to a vertex in the other with an edge. By Theorem 2.8, we can get an algorithm for the inverse problem. The idea is to use the formulas in Theorem 2.8 to fill in a 3-D matrix M; each cell in M is (n, σ_u, σ^u), bottom-up. A standard backtracking method will give the corresponding tree. The running time is $O(n\sigma^2)$ since computing each cell requires $O(1)$ time. Note that the running time is pseudo-polynomial since it depends on the input value of the σ-index.

To set up a library in practice, all possible trees need to be recorded for a given σ-index value. This can be done by modifying the backtracking process. In the backtracking process, instead of finding one path from the last cell back to a cell in the first column/row in the matrix, we can find all possible paths leading to solutions. However, the total number of paths leading to solutions might be exponential. Another choice is to keep a graph that contains all possible paths leading to solutions and randomly choose some paths to generate trees. Jiang and Wang (2002) gave an example for multiple sequence alignment. The same idea can be used here.

The SUBTREEVALUE reconstruction problem

Now, we consider a new problem. Let v_1, v_2, ..., v_k be the children of v in a tree. Recall that $\bar{\sigma}_v = \sigma_{v_1}\sigma_{v_2}\ldots\sigma_{v_k}$ is the number of independent sets containing vertex v for the subtree rooted at v. Given a set of n numbers $\{\sigma_1, \sigma_2, \ldots \sigma_n\}$, we want to find a tree T of n vertices v_1, v_2, ..., v_n such that $\bar{\sigma}_{v_i} = \sigma_i$ for $i = 1, 2, \ldots, n$. The problem is called *SUBTREEVALUE reconstruction.*

Theorem 2.9. *The SUBTREEVALUE reconstruction problem is NP-complete.*

Proof. The reduction is from the 3-PARTITION problem, which is NP-complete (Garey and Johnson, 1979). In this problem, we are given a bound B and $3m$ elements, $s_1, s_2, \ldots, s_{3m}$, such that for each $i \in \{1, 2, \ldots, 3m\}$, $\frac{B}{4} < s_i < \frac{B}{2}$. The problem asks whether there exists a partition of the set $\{s_1, s_2, \ldots, s_{3m}\}$ into 3-element disjoint sets such that the sum of the 3 elements in each set is B.

We map the instance of 3-PARTITION to the following instance of SUBTREEVALUE reconstruction: the number 1 appears $\sum_{i=1}^{3m} s_i + 3m = mB + 3m$ times, the number 2^B appears m times, and the number $\Pi_{i=1}^{3m}(1 + 2^{s_i})$ appears once.

If we are given a partition $P = \{P_1, P_2, \ldots P_m\}$, where $P_i = \{s_{i,1}, s_{i,2}, s_{i,3}\}$ and $\sum_{j=1}^{3} s_{i,j} = B$, of a "Yes" instance of 3-PARTITION, we can build a tree in the following way: the root r has m children v_1, v_2, ..., v_m, each v_i has three children $v_{i,1}$, $v_{i,2}$, and $v_{i,3}$ corresponding to a set of 3-element P_i in the partition, and each $v_{i,j}$ has $s_{i,j}$ children at the bottom. We can see that this tree satisfies the given values of the SUBTREEVALUE reconstruction problem.

Conversely, suppose that we are given a tree T with the given values of the SUBTREEVALUE reconstruction problem. The m 2^m's and the fact that there is no other given number between 1 and 2^m ensure that there are m subtrees $T_1, T_2, \ldots, T_m$ in T such that each T_i has three levels of vertices and T_i contains m leaves at the bottom. The fact that there is no given number between 2^m and $\Pi_{i=1}^{3m}(1 + 2^{s_i})$ implies that T is a tree formed by connecting T_1, T_2, ..., T_m with a root r. The number $\Pi_{i=1}^{3m}(1 + 2^{s_i})$ implies that eliminating the roots of T_i's in the tree, we have $3m$ subtrees T_{s_i}'s; each T_{s_i} is a star containing s_i leaves. Since $\frac{B}{4} < s_i < \frac{B}{2}$, we know that each T_i contains three T_{s_i}'s, say, $T_{s_{i1}}$, $T_{s_{i2}}$, and $T_{s_{i3}}$, such that $s_{i1} + s_{i2} + s_{i3} = m$. Therefore, we can get a desired partition for the instance of 3-PARTITION. ■

3. THE *c*-INDEX

For a graph G, the *c-index* $c(G)$ is defined as the number of cliques of all sizes. The c-index is the complement of the σ-index. It is easy to see that $c(G) = \sigma(\bar{G})$ for any graph G. Clearly, $c(\Phi) = 1$, and $c(K_{1,n-1}) = 2n$, $c(P_n) = 2n$. Also, $c(C_3) = 8$ and $c(C_n) = 2n + 1$ for $n \geq 4$, where C_n stands for a polygon of n vertices.

Theorem 3.1. *For any natural number $c \neq 3, 5,$ or 7, there exists a graph G such that $c(G) = c$.*

Proof. It is easy to see that the graphs Φ, P_1, P_2, and P_3 have c-index values 1, 2, 4, and 6, respectively. For $c = 2k$ ($k \geq 4$), any tree with k vertices has a c-index value $2k = c$. (Those $2k$ cliques are k singletons, one for each vertex, $k-1$ sets containing two elements, one for each edge, and the empty set.) For $c = 2k+1$ ($k \geq 4$), we have $c(C_k) = 2k + 1 = c$.

Now we show that there is no graph with c-index 3, 5, or 7. In fact, it is easy to see that $c(G)$ increases when either $|V(G)|$ or $|E(G)|$ increases. Also, $c(C_3) = 8$ and $c(P_4) = 8$. Removing an edge from C_3, we get the nearest c-index smaller than 8, which is 6. Also, removing a vertex from P_4, we get the nearest c-index smaller than 8, which is again 6. So, there is no graph G with $c(G) = 7$. Since $c(P_3) = 6$, the possible graphs with c-index 3 or 5 are the graphs Φ, P_1, and P_2. However, as we pointed out at the beginning of the proof, it is not the case. Thus, the proof is completed. ■

Theorem 3.2. *Among all the graphs with n vertices, only trees have the minimum value of the c-index; whereas the complete graph K_n is the unique one with the maximum value of the c-index.*

Proof. Note that $c(G)$ increases when $|V(G)|$ or $|E(G)|$ increases. Then, the theorem follows from the fact that trees have the minimum number of edges among all connected graphs with n vertices; whereas the complete graph has the maximum number of edges among all connected graphs with n vertices. The details are omitted. ■

4. THE Z-INDEX

The *Z-index* of a graph G is defined to be the number of sets of independent edges of all sizes. Clearly, $Z(\Phi) = 1$, $Z(P_1) = 1$, $Z(P_n) = F_n$, where F_n is the Fibonacci number such that $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = F_1 = 1$, and $Z(K_{1,n-1}) = n$. The Z-index is also called the *matching index.*

The following result is similar to Lemma 2.1 for the σ-index.

Lemma 4.1. *Let $e = uv$ be any edge of G. Then $Z(G) = Z(G - e) + Z(G - u - v)$.*

Theorem 4.2. *For any natural number n, there exists a tree T such that $Z(T) = n$.*

Proof. Take $T = K_{1,n-1}$. Then $Z(T) = n$. ■

It is easy to verify the following:

Theorem 4.3. *Among all graphs with n vertices, $K_{1,n-1}$ is the unique one with the minimum value of the Z-index; whereas K_n is the unique one with the maximum value of the Z-index.*

Theorem 4.4. *Among all trees with n vertices, $K_{1,n-1}$ is the unique one with the minimum value of the Z-index; whereas P_n is the unique one with the maximum value of the Z-index.*

Proof. The first part comes from the first part of Theorem 4.3. Now, we prove the second part. Take any tree T with n vertices. Find a vertex u of T such that all the neighbors but one are leaves of T. Then by Lemma 4.1 we have

$$Z(T) = Z(T - v_1 - \cdots - v_{d_u - 1}) + (d_u - 1)Z(T - u - v_1 - \cdots - v_{d_u - 1}),$$

where $v_1, \ldots, v_{d_u - 1}$ are all the leaves of T that are adjacent to u. Note that $T - v_1 - \cdots - v_{d_u - 1}$ and $T - u - v_1 - \cdots - v_{d_u - 1}$ are trees with $n - d_u + 1$ and $n - d_u$ vertices, respectively. By induction on n, we have

$$Z(T) \leq Z(P_{n-d_u+1}) + (d_u - 1)Z(P_{n-d_u}) = F_{n-d_u+1} + (d_u - 1)F_{n-d_u}.$$

Since $d_u - 1 \leq F_{d_u} - 1$ $(d_u \geq 2)$, we have

$$\begin{aligned} Z(T) &\leq F_{n-d_u+1} + (F_{d_u} - 1)F_{n-d_u} \\ &= F_{n-d_u+1} + (F_{d_u-1} + F_{d_u-2} - 1)F_{n-d_u} \\ &= F_{n-d_u+1} + F_{d_u-2}F_{n-d_u} + (F_{d_u-1} - 1)F_{n-d_u} \\ &\leq F_{d_u-2}F_{n-d_u} + F_{n-d_u+1} + (F_{d_u-1} - 1)F_{n-d_u+1} \\ &= F_{d_u-1}F_{n-d_u+1} + F_{d_u-2}F_{n-d_u}. \end{aligned}$$

Note that

$$F_{k+l} = F_k F_{l+1} + F_{k-1}F_l.$$

So, we obtain that

$$Z(T) \leq F_n,$$

by taking $k = d_u - 1$ and $l = n - d_u$. That is, $Z(T) \leq Z(P_n)$, where equality holds if and only if $d_u = 2$. This implies that P_n is the unique tree with the minimum value of the Z-index. ■

5. THE M_1-INDEX

Let $d_1, d_2, \ldots, d_n$ be the degrees of the n vertices in a graph G. In the early work of the Zagreb group on the topological basis of π-electron energy (Trinajstic, 1992), two terms appeared in the approximate formula for the total π-energy of a conjugated system which may be used separately as topological indices

$$M_1(G) = \sum_{i=1}^{n} d_i^2,$$

and

$$M_2(G) = \sum_{ij \in E(G)} d_i d_j.$$

Here we only investigate the M_1-*index.*

Lemma 5.1. *The M_1-index $M_1(G)$ for any graph G is an even number.*

Proof. Since $\sum_{i=1}^{n} d_i = 2|E(G)|$ is an even number, there is an even number of odd numbers among $d_1, d_2, \ldots, d_n$. Therefore, $M_1(G) = \sum_{i=1}^{n} d_i^2$ is even. ■

Lemma 5.2. *There is no graph with* 4 *or* 8 *as its M_1-index value.*

Proof. By trying to write 4 or 8 into all kinds of sums of square numbers, one can see that there is no corresponding connected graph. ■

Theorem 5.3. *For any given even number $m_1 \neq 4$ or* 8, *there exists a tree T such that $M_1(T) = m_1$.*

Proof. We distinguish m_1 into two cases:
Case 1. $m_1 = 4k + 2$ $(k \geq 0)$
Take $n = k + 2$. Then $M_1(P_n) = 4k + 2$.
Case 2. $m_1 = 4k$ $(k \neq 1, 2)$
For $k = 0$, we know that $M_1(\Phi) = 0 = m_1$.
For $k = 1$ and 2, from Lemma 5.2, there is no corresponding graph and thus no tree with $m_1 = 4k$ as its M_1-index.
For $k \geq 3$, we first take the star $K_{1,3}$ (a claw) with $M_1(K_{1,3}) = 3^2 + 1 + 1 + 1 = 12 = 4 \times 3$. Then we grow $K_{1,3}$ by connecting a new vertex with a leaf step by step as in Fig. 3. Connecting a new vertex to a leaf increases the M_1-index value by 2^2. Thus, the graph shown in Fig. 3 has M_1-index value

$$3^2 + 1 + 1 + 1 + 2^2 + 2^2 + \ldots + 2^2 = 12 + 2^2 + 2^2 + \ldots + 2^2.$$

Therefore, for any $k \geq 3$, there is a tree as shown in Fig. 3 with M_1-index value $4k$.
In fact, taking any two trees T_1 and T_2 such that $M_1(T_1) = 4k' + 2$ and $M_1(T_2) = 4k'$ and growing them by connecting a new vertex to a leaf of them, step by step we can obtain many trees with M_1-index value $4k + 2$ or $4k$. ■

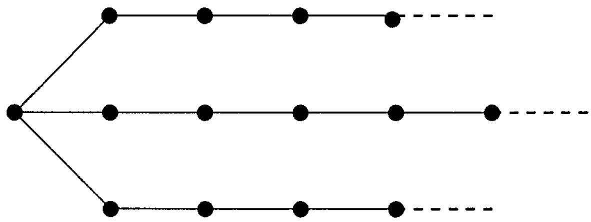

FIG. 3. Trees with the M_1-index equal to $4k$.

Lemma 5.4. *(1) If $e = ij$ is an edge in G, then $M_1(G - e) = M_1(G) - 2(d_i + d_j - 1)$; (2) If $f = kl$ is not an edge in G, then $M_1(G + f) = M_1(G) + 2(d_k + d_l + 1)$; (3) If $e = ij$ is an edge but $f = kl$ is not an edge in G, then $M_1(G - e + f) = M_1(G) + 2(d_k + d_l - d_i - d_j + 2)$ when $\{i, j\} \bigcap \{k, l\} = \Phi$; and $M_1(G - e + f) = M_1(G) + 2(d_k - d_i + 1)$ when $j = l$.*

Theorem 5.5. *Among all the graphs with n vertices, P_n is the unique one with the minimum value of the M_1-index; whereas K_n is the unique one with the maximum value of the M_1-index.*

Proof. The second part is obvious. Now, we show the first part by induction on the number n of vertices of a graph G. For $n = 1$ and 2, the conclusion is obviously true. Assume that for $n - 1$ the conclusion is true, i.e., P_{n-1} is the unique graph with the minimum value of the M_1-index. Now we add a new vertex v into a graph G' with $n - 1$ vertices to construct a graph G with n vertices. If we want $M_1(G)$ to be as small as possible, we have to connect v with a vertex of the smallest degree, say d, in G'. Then, we get $M_1(G) = 2(d + 1) + M_1(G')$. Clearly, $M_1(G)$ reaches the minimum if and only if both d and $M_1(G')$ reach their minimum values, which happens if and only if G' is a path, according to the induction hypothesis, and v is connected to an end vertex of the path. So, G must be a path with n vertices. The proof is complete. ■

Obviously, $M_1(P_n) = 4n - 6$ for $n \geq 2$; $M_1(P_n) = 0$ for $n = 1$; and $M_1(K_n) = n(n - 1)^2$.

Theorem 5.6. *Among all the trees with n vertices, P_n is the unique one with the minimum value of the M_1-index; whereas the star $K_{1,n-1}$ is the unique one with the maximum value of the M_1-index.*

Proof. The first part follows from the first part of Theorem 5.5. Now, we prove the second part. Let T be a tree and d_i the degree of the i-th vertex in T. Consider the vertex i in T such that $d_i \geq d_j$ for $j = 1, 2, \ldots, n$. If all leaves of T are adjacent to i, then $T = K_{1,n-1}$. Otherwise, at least one of the leaves in T, say j, is not adjacent to i. Let jk be an edge of T, then $d_k \leq d_i$. Constructing a new tree $T' = T - jk + ji$, we have, from Lemma 5.4 (3),

$$M_1(T') \geq M_1(T) - 2(d_i - d_k + 1) = M_1(T) + 2(d_i - d_k + 1) > M_1(T).$$

Repeating this process, we can obtain the star $K_{1,n-1}$ with the M_1-index value strictly increased. This also implies that $K_{1,n-1}$ is the unique tree with the maximum value of the M_1-index. □ ■

ACKNOWLEDGMENTS

The authors would like to thank the referees for their helpful suggestions and comments. This work is fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 1058/98E).

REFERENCES

Garey, M.R., and Johnson, D.S. 1979. *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York.

Goldman, D., Istrail, S., Lancia, G., Piccolboni, A., and Walenz, B. 2000. Algorithmic strategies in combinatorial chemistry. In *Proc. 11th ACM-SIAM Sym. on Discrete Algorithms*, 275–284.

Harary, F. 1969. *Graph Theory*, Addison-Wesley.

Jiang, T., and Wang, L. 2002. Algorithmic methods for multiple sequence alignment. In Jiang, T., Xu, Y., and Zhang, M.Q., eds. *Current Topics in Computational Molecular Biology*, MIT Press.

Merrifield, R.E., and Simmons, H.E. 1989. *Topological Methods in Chemistry*, John Wiley, New York.

Rouvray, D.H. 1973. The search for useful topological indices in chemistry. *American Scientist* 61(6), 729–735.

Sabljić, A., and Trinajstić, N. 1981. Quantitative structure-activity relationships: The role of topological indices. *Acta Pharm. Jugosl.* 31, 189–214.

Sheridan, R.P., and Kearsley, S.K. 1995. Using a genetic algorithm to suggest combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 35, 310–320.

Trinajstic, N. 1992. *Chemical Graph Theory*, CRC Press.

Venkatasubramanian, V., Chan, K., and Caruthers, J.M. 1995. Evolutionary design of molecules with desired properties using the genetic algorithm. *J. Chem. Inf. Comput. Sci.* 35, 188–195.

Address correspondence to:
Lusheng Wang
Department of Computer Science
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong

E-mail: lwang@cs.cityu.edu.hk