CROATICA CHEMICA ACTA CCACAA **75** (2) 577–601 (2002)

ISSN-0011-1643
CCA-2818

Original Scientific Paper

Building-block Computation of Wiener-type Indices for the Virtual Screening of Combinatorial Libraries*

*Ovidiu Ivanciuc*** *and Douglas J. Klein***

Department of Marine Sciences, Texas A & M University at Galveston,
Fort Crockett Campus, 5007 Avenue U, Galveston, TX 77551, USA

Received June 5, 2001; revised August 20, 2001; accepted August 21, 2001

Screening virtual and synthetic combinatorial libraries may facilitate rapid drug lead discovery by selecting subsets of molecules according to their similarity or dissimilarity toward specific compound collections. Topological indices computed from atomic connectivities or graph distances are increasingly used as structural descriptors in order to maximize the molecular diversity of libraries or to quantify the drug-like character of compounds. In this paper we present efficient equations for the computation of several distance-based topological indices of a molecular graph from the distance invariants of its subgraphs. These equations offer an effective way to compute for non-weighted molecular graphs the Wiener index, even/odd Wiener index, resistance distance index, Wiener polynomial, and even/odd Wiener polynomial. Using a simple and fast algorithm one can compute these topological indices for very large virtual combinatorial libraries without computing the indices from the atomic scale up for each individual compound – rather only distance-based indices of the building blocks are needed to generate the topological indices of the compound assembled from the building blocks.

Key words: virtual combinatorial libraries, drug discovery, topological indices, Wiener index.

* Dedicated to Professor Milan Randić on the occasion of the 70th birthday.

** E-mail: ivanciuc@netscape.net. and kleind@tamug.tamu.edu, respectively.

INTRODUCTION

The present revolution in the drug discovery process is driven by the recent progress in combinatorial chemistry and high throughput screening techniques. The random testing of chemicals is not the best method to obtain the maximum possible information from a combinatorial library, while the synthesis and screening of a very large number of compounds is extremely costly. The current trend is represented by an efficient utilization of computational techniques in the virtual screening of combinatorial libraries (VSCL) in order to increase the drug-like character and diversity of the compounds proposed for screening.[1,2] In order to be successful, this process of *in silico* compound selection must incorporate additional target-specific information.

Conventionally, the computational screening of chemical libraries is generally a four-step process:

– assemble the compounds from a group of building blocks;

– computation for each chemical compound of a set of structural descriptors;

– dimensionality reduction by selecting from the descriptors set a chemical space that is relevant for the investigated target;

– compound selection with a statistical algorithm that implements a similarity, diversity, or drug-like paradigm.

A large number of structural descriptors, many of them traditionally used in QSPR and QSAR, are currently used in VSCL in order to transform into a numerical form the structural features of molecules: physico-chemical or empirical (log *P*, molecular polarizability); constitutional (number of aromatic rings, number of rotatable bonds, number of hydrogen-bond donors, number of hydrogen-bond acceptors); structure keys and fingerprints; graph invariants (cyclomatic number, atom pairs, path counts); topological indices[3,4] (Wiener,[5,6] Randiæ,[7] Kier and Hall,[8] Balaban,[9,10] Harary[11,12]); geometric (polar surface area, molecular volume); quantum (HOMO energy, atomic charges); grid (various steric, electrostatic, and lipophilic fields).

Although SAR and QSAR studies offer a rich variety of structural descriptors, not all of them are fit for use in VSCL. While the typical number of compounds in SAR and QSAR is usually between 10 and 100, a virtual library can easily exceed 10^6 compounds. In order to be efficient, the *in silico* compound screening must use descriptors that require small computational resources, thus explaining the wide popularity of counts of atom types, counts of functional groups, fingerprints, constitutional descriptors, graph invariants and topological indices. This trend is apparent from the examination of several recent papers relevant to VSCL including: selection of drug-like compounds;[13,14] characterization of drug classes;[15] property distribution in drug databases;[16] classification of compounds according to the biological activity;[17] Focus-2D, which uses simulated annealing and genetic algorithms to generate targeted libraries;[18] design of immunosuppressive compounds;[19] characterization of building blocks chemical space;[20] design of libraries with CNS activity;[21] ChemGPS, a chemical global positioning system.[22]

Because the selection of candidates for synthesis from chemical libraries is time consuming whenever it involves hard to compute descriptors, such as 3D pharmacophores, quantum indices, grid descriptors or other relevant 3D descriptors, the first step in VSCL implements a fast and efficient filtering method intended to eliminate inappropriate reactants or products before going to more sophisticated structural descriptors and screening methods. For instance, there are promising ways of achieving an efficient sampling of the virtual product set by computing the structural descriptors of products without actually assembling the compounds from the building blocks.[23–27] This procedure can be applied for all additive or nearly additive descriptors, or for product descriptors that can be computed with a simple mathematical procedure from the corresponding descriptors of reactants or building blocks and a proper representation of the chemical process that takes place. Several such structural descriptors include: molecular weight, total number of atoms, number of atoms with a given atomic number Z, mass percent of atoms with a given atomic number Z, number of hydrogen bond donors, number of hydrogen bond acceptors, number of rotatable bonds, number of rings, number of aromatic rings, total number of bonds, number of single, double, triple, or aromatic bonds, atom types, number of various functional groups. Another approach for obtaining product descriptors based on the building blocks structure is to propose simplified algorithms and equations for the computation of those QSAR descriptors that traditionally require the examination of the whole 2D or 3D product structure: octanol-water partition coefficient obtained from atomic contributions;[28] calculation of molecular polar surface area as a sum of N, O, P and S atoms contributions;[29] approximate van der Waals molecular surface area computed from the atomic connectivity information;[30] molecular polar surface area computed from the number of hydrogen bond donors and acceptors;[31] fast log P computation from the sum of polar atoms, sum of nonpolar atoms, and sum of hydrogen bond donors and acceptors.[32]

Although graph invariants and topological indices represent highly used structural descriptors efficiently employed to quantify the drug-like character of compounds or to measure the similarity and diversity of chemical libraries, their computation from reactants or building blocks has not previously been emphasized as a major way to speed-up the VSCL process.

In the present paper we demonstrate several theorems for the calculation of some highly used distance-based topological indices of a molecular graph from the distance invariants of its subgraphs. These equations offer an efficient way to compute for non-weighted molecular graphs the Wiener index, even/odd Wiener index, resistance distance index, Szeged index, Wiener polynomial, and even/odd Wiener polynomial. As should be of especial use for combinatorial libraries, the computation of these descriptors is based on the molecular graph invariants of the corresponding building blocks.

WIENER INDEX

Let G be a simple (nonweighted) connected graph consisting of the vertex set $V(\mathrm{G})$ and the edge set $E(\mathrm{G})$, and having $|V(\mathrm{G})|$ vertices and $|E(\mathrm{G})|$ edges. Molecular graphs are non-directed connected graphs which represent chemical compounds. In molecular graphs vertices correspond to atoms and edges represent covalent bonds between atoms. Alkanes and cycloalkanes are usually represented as simple molecular graphs in which each vertex corresponds to a carbon atom and each edge corresponds to a carbon–carbon single bond. In this paper chemical compounds are represented as hydrogen-suppressed simple (non-weighted) graphs, *i.e.* all vertex-weights are zero and all bond-weights are equal to one. We must emphasize that even organic compounds containing heteroatoms and multiple bonds can be represented as simple molecular graphs whenever one is interested in computing topological indices and graph invariants that reflect only the molecular connectivity and are free from the influence of various heteroatoms and different types of bonding. Almost all commercial programs that compute distance-based topological indices use this convention, and due to this situation all VSCL applications of these indices use this simplification. However, molecules containing heteroatoms and multiple bonds can be represented as vertex- and edge-weighted graphs, with special parameters for vertices (atoms) and edges (bonds). The computation of the distance-based topological indices for weighted molecular graphs is presented in several recent papers.[33–36] With d_{ij} we denote the topological distance between vertices i and j of G, representing the length of the shortest path between vertices i and j, *i.e.* the number of edges on that path. A cut edge of G is an edge which if deleted breaks G into two disconnected subgraphs. A cut vertex of G is a vertex which if deleted breaks G into two disconnected subgraphs.

One of the most widely used graph descriptors, the Wiener index W, was initially defined only for alkanes.[5,6] Consider an acyclic graph $\mathrm{G} = \mathrm{G}(V, E)$ and denote with $n_{i(j)}$ and $n_{j(i)}$ the number of vertices situated on the different sides of the edge $\{i, j\}$; vertex i is counted in $n_{i(j)}$ while vertex j is counted in

$n_{j(i)}$. For acyclic graphs the Wiener index $W = W(\mathrm{G})$ of a graph G may be given by the formula:

$$W(\mathrm{G}) = \sum_{\{i,j\}\in E(\mathrm{G})} n_{i(j)} n_{j(i)} \tag{1}$$

where the summation goes over all edges from the edge set $E(\mathrm{G})$, $\{i, j\} \in E(\mathrm{G})$. For cyclic graphs the two invariants $n_{i(j)}$ and $n_{j(i)}$ are ambiguously so defined due to the existence of vertices k situated at the same distance from i and j, $d_{ik} = d_{kj}$. Hosoya extended the definition of the Wiener index to cyclic graphs with the aid of the distance matrix as »the half sum of the off-diagonal elements of a distance matrix $\boldsymbol{D}$ whose element d_{ij} is a number of bonds for the shortest path between atoms i and j«.[37] This extension of Eq. (1) to cyclic graphs is not unique, and alternative ways have been proposed. For example, the resistance distance index W' defined as the sum for all pairs of vertices of the corresponding elements in the resistance distance matrix $\boldsymbol{\Omega}$[38–41] has values coincident with W for acyclic graphs while for cycle-containing graphs they differ. Gutman proposed the Szeged index Sz computed with invariants $n_{i(j)}$ and $n_{j(i)}$ extended for cyclic graphs;[42,43] Sz shares with W' the same property of being identical with W for acyclic graphs and different for cyclic graphs. Diudea defined the Szeged matrix which is the source of a large number of topological indices.[44,45] More details regarding these distance-based topological indices and related graph invariants can be found in two recent reviews.[4,46] The actual definition of the Wiener index, which is valid also for vertex- and edge-weighted molecular graphs representing organic compounds with heteroatoms and multiple bonds, is:[4,34,35]

$$W(\mathrm{G}) = \sum_{i=1}^{N} \sum_{j=i}^{N} \boldsymbol{D}_{ij}(\mathrm{G}) \; . \tag{2}$$

The distance sum for vertex i from graph G, $W_i(\mathrm{G})$, is the sum of all distances between vertex i and all other vertices from G:

$$W_i(\mathrm{G}) = \sum_{j\in V(\mathrm{G})} d_{ij}(\mathrm{G}) \; . \tag{3}$$

For the computation of the Wiener index from subgraphs (building blocks) we will use a well-known graph decomposition formula for W:[47–49]

Theorem 1. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in \mathrm{A}$ and $b \in \mathrm{B}$. Denote various subgraphs of $\mathrm{G} = \mathrm{A–B}$ as in the Figure 1. Then the Wiener index of graph G is:

$$W(\mathrm{G}) = W(\mathrm{A}-\mathrm{B}) = W(\mathrm{A}) + |V(\mathrm{B})| W_a(\mathrm{A}) + |V(\mathrm{A})||V(\mathrm{B})| + |V(\mathrm{A})| W_b(\mathrm{B}) + W(\mathrm{B}) \; . \tag{4}$$

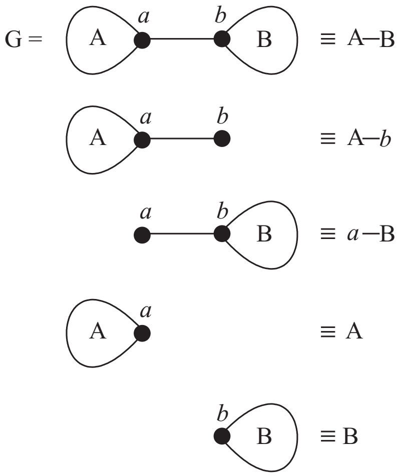

Figure 1. A graph G with a cut edge between vertices a and b, and its relevant subgraphs.

This formula is the core of an efficient algorithm for the fast computation of the Wiener index for compounds from combinatorial libraries. The description of the algorithm for computing the Wiener index from building blocks is first presented for the graphs from Figure 2:

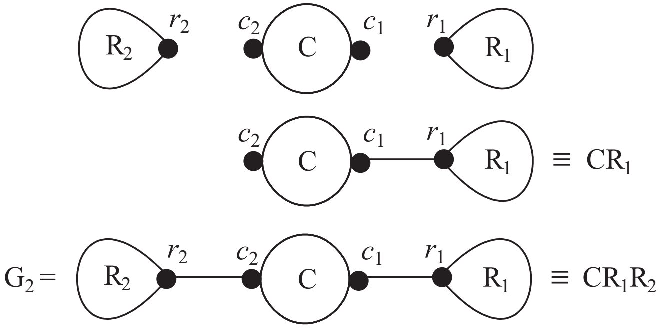

Figure 2. Chemical compound generation from a core structure C and two substituents R_1 and R_2.

Algorithm 1.

1. Consider a core structure C with two substitution atoms c_1 and c_2, and two substituents R_1 and R_2 each having one substitution atom, r_1 and r_2, respectively, all as in Figure 2.

2. Compute the Wiener indices for the core structure C and the two substituents R_1 and R_2, *i.e.* $W(C)$, $W(R_1)$, and $W(R_2)$.

3. Compute the distance sum for the substitution atoms c_1, c_2, r_1, and r_2, *i.e.* $W_{c_1}(C)$, $W_{c_2}(C)$, $W_{r_1}(R_1)$, and $W_{r_2}(R_2)$.

4. Determine the distance between the two substitution atoms from the core C, $d_{c_1c_2}$.

5. Link the substituent R_1 to C by inserting a bond between atoms r_1 and c_1, and obtain in this way CR_1.

6. Compute the Wiener index of CR_1, $W(CR_1)$, by applying Eq. (4) to the molecular graph CR_1 obtained by linking together C and R_1:

$$W(CR_1) = W(C) + |V(R_1)|W_{c_1}(C) + |V(C)||V(R_1)| + |V(C)|W_{r_1}(R_1) + W(R_1)\ . \qquad (5)$$

7. Determine the number of vertices in the subgraph CR_1:

$$|V(CR_1)| = |V(C)| + |V(R_1)|\ .$$

8. Compute the distance sum of atom c_2 in the intermediate structure CR_1 with the equation:

$$W_{c_2}(CR_1) = W_{c_2}(C) + (d_{c_1c_2} + 1)|V(R_1)| + W_{r_1}(R_1)\ . \qquad (6)$$

The distance sum $W_{c_2}(CR_1)$ is necessary for the computation of the Wiener index for the compound CR_1R_2.

9. Link the substituent R_2 to CR_1 by inserting a bond between atoms r_2 and c_2, and obtain in this way CR_1R_2.

10. Compute the Wiener index of CR_1R_2, $W(CR_1R_2)$, by applying Eq. (5) to the molecular graph CR_1R_2 obtained by linking together CR_1 and R_2:

$$W(CR_1R_2) = W(CR_1) + |V(R_2)|W_{c_2}(CR_1) + |V(CR_1)||V(R_2)| + \\ + |V(CR_1)|W_{r_2}(R_2) + W(R_2)\ . \qquad (7)$$

The scope of our investigation is to propose fast and efficient algorithms for computing distance-based topological indices for combinatorial libraries, and therefore it is important to compare the computational expense for computing the Wiener index W with the usual Eq. (2) and with Eq. (4), as proposed in the above algorithm. We will first examine the algorithm that involves Eq. (2). For a molecular graph consisting of N atoms the connection table is translated into the adjacency matrix $\boldsymbol{A}$ in $O(N)$ computer operations.

The computation of the distance matrix $\boldsymbol{D}$ from the adjacency matrix for each compound needs $O(N^3)$ operations, *i.e.* the computer time is proportional to N^3, indicating that when N increases this part of the algorithm will need a powerful computer. The computation of W from $\boldsymbol{D}$ with Eq. (2) is accomplished in $O(N^2)$ steps. From this analysis it is clear that the most computational demanding step is the generation of the distance matrix $\boldsymbol{D}$ from the adjacency matrix. Chemical compounds generated for combinatorial libraries are fairly large and the above analysis demonstrates that the standard algorithm for the Wiener index is not efficient for VSCL.

We will now examine the algorithm for computing the Wiener index W with Eq. (4) for the general case when a chemical compound $CR_1...R_m$ is generated from a core structure C and m substituents $R_1, R_2,..., R_m$, as presented in Figure 3.

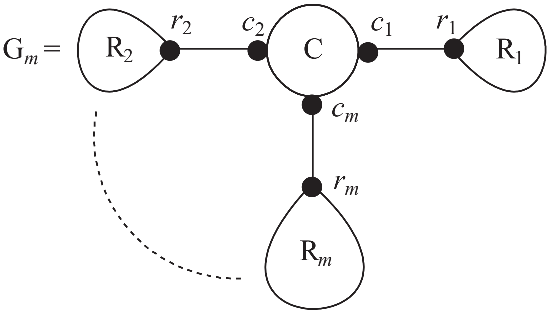

Figure 3. Chemical compound generation from a core structure C and m substituents $R_1, R_2,..., R_m$.

Algorithm 2.

1. Consider a core structure C with m substitution sites $c_1, c_2,..., c_m$ and m substituents $R_1, R_2,..., R_m$, each having one substitution atom $r_1, r_2,..., r_m$, respectively. The final molecular graph is presented in Figure 3.

2. Compute the Wiener indices for the core structure C and the m substituents $R_1, R_2,..., R_m$, *i.e.* $W(C)$, $W(R_1)$, $W(R_2)$,..., $W(R_m)$.

3. Compute the vertex sum for the substitution atoms $c_1, c_2,..., c_m, r_1, r_2,..., r_m$, *i.e.* $W_{c_1}(C)$, $W_{c_2}(C)$,..., $W_{c_m}(C)$, $W_{r_1}(R_1)$, $W_{r_2}(R_2)$,..., $W_{r_m}(R_m)$.

4. Determine the distance between all $m(m-1)/2$ pairs of substitution atoms from the core C, $d_{c_i c_j}$.

5. To an intermediate structure $CR_1...R_{i-1}$ add the substituent R_i by inserting a bond between atoms r_i and c_i, and obtain in this way $CR_1...R_i$.

6. Compute the Wiener index of $\mathrm{CR}_1\ldots\mathrm{R}_i$, $W(\mathrm{CR}_1\ldots\mathrm{R}_i)$:

$$W(\mathrm{CR}_1\ldots\mathrm{R}_i) = W(\mathrm{CR}_1\ldots\mathrm{R}_{i-1}) + |V(\mathrm{R}_i)|W_{c_i}(\mathrm{CR}_1\ldots\mathrm{R}_{i-1}) + \\ |V(\mathrm{CR}_1\ldots\mathrm{R}_{i-1})||V(\mathrm{R}_i)| + |V(\mathrm{CR}_1\ldots\mathrm{R}_{i-1})|W_{r_i}(\mathrm{R}_i) + W(\mathrm{R}_i)\ . \tag{8}$$

7. Determine the number of vertices in the subgraph $\mathrm{CR}_1\ldots\mathrm{R}_i$:

$$|V(\mathrm{CR}_1\ldots\mathrm{R}_i)| = |V(\mathrm{CR}_1\ldots\mathrm{R}_{i-1})| + |V(\mathrm{R}_i)|\ .$$

8. Update the distance sum of atoms c_j, $i < j \le m$, in the intermediate structure $\mathrm{CR}_1\ldots\mathrm{R}_i$ with the equation:

$$W_{c_j}(\mathrm{CR}_1\ldots\mathrm{R}_i) = W_{c_j}(\mathrm{CR}_1\ldots\mathrm{R}_{i-1}) + (d_{c_ic_j} + 1)|V(\mathrm{R}_i)| + W_{r_i}(\mathrm{R}_i)\ . \tag{9}$$

Repeat steps 5 through 8 until $W(\mathrm{CR}_1\ldots\mathrm{R}_m)$ is computed.

We give an equivalent formula for computing the updated distance sum for the atom c_k after the first i substituents have been added to the core structure C:

$$W_{c_k}(\mathrm{CR}_1\ldots\mathrm{R}_i) = W_{c_k}(\mathrm{C}) + \sum_{j=1}^{i}\Big[(d_{c_jc_k} + 1)|V(\mathrm{R}_j)| + W_{r_j}(\mathrm{R}_j)\Big]\ . \tag{10}$$

From a thorough examination of the above algorithm we can easily divide the computational expenses in two types: computation of several distance-based graph invariants for the building blocks and m times application of Eq. (4) for computing W of $\mathrm{CR}_1\ldots\mathrm{R}_m$. The computation of the Wiener index with Eq. (2) for C, R_1, R_2,..., R_m, distance sum from $\boldsymbol{D}$ for atoms c_1, c_2,..., c_m, r_1, r_2,..., r_m, and distances between all pairs of atoms c_1, c_2,..., c_m, is the most computational demanding step, but we have to consider that the size of the building blocks is much smaller than that of the final compound, and this computation is performed only once for the whole combinatorial library. The second step involves the m–fold application of Eq. (4) and has a time expense of $O(m)$, where m, in general, is smaller than 5. One important feature of the second step, which is repeated for each compound generated, is that the computational expenses do not depend on the number of atoms N from the final compound but on the number of building blocks, which is small and constant for a given virtual library. At the i^{th} stage, step 8 involves $m - i$ computations, so that as i ranges from 1 to m. Thus this step involves overall $m(m - 1)/2$ such computations for each compound, *i.e.*, a time $O(m^2)$ arises. It is now clear that the usual algorithm for computing the Wiener index with Eq. (2), which involves $O(N^3)$ operations per compound, is much less efficient than the algorithm that uses Eq. (4) and involves $O(m^2)$ operations per compound. Also, the larger the combinatorial library

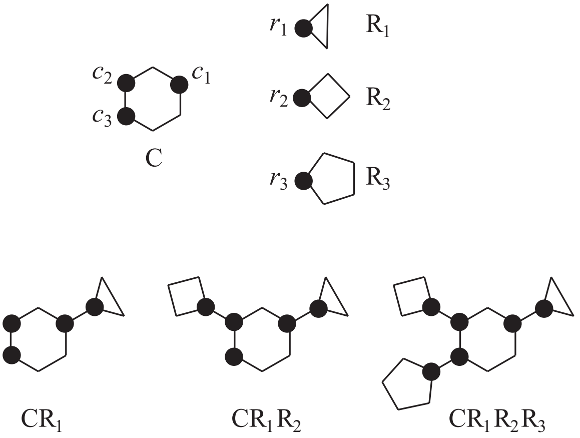

Figure 4. Core structure C and three substituents R_1, R_2 and R_3 used as example in the computation of distance-based topological indices from building blocks.

is, the greater the relative efficiency of our algorithm that uses Eq. (4), compared with the standard one.

We present now an application of Theorem 1 to the computation of the Wiener index W for a molecular graph generated from building blocks. Consider the core structure C and the three substituents R_1, R_2, and R_3 from Figure 4, all of them used to assemble the graph $CR_1R_2R_3$. The computation of the Wiener index for $CR_1R_2R_3$ is performed with a procedure similar to that one presented in Algorithm 1:

1. Compute the Wiener indices for the four building blocks: $W(\mathrm{C}) = 27$, $W(\mathrm{R}_1) = 3$, $W(\mathrm{R}_2) = 8$, $W(\mathrm{R}_3) = 15$.

2. Compute the distance sums for the six connection atoms: $W_{c_1}(\mathrm{C}) = 9$, $W_{c_2}(\mathrm{C}) = 9$, $W_{c_3}(\mathrm{C}) = 9$, $W_{r_1}(\mathrm{R}_1) = 2$, $W_{r_2}(\mathrm{R}_2) = 4$, $W_{r_3}(\mathrm{R}_3) = 6$.

3. Compute the shortest-path distance between all pairs of connection vertices from the core structure C: $d_{c_1c_2} = 2$, $d_{c_1c_3} = 3$, $d_{c_2c_3} = 1$.

4. Get the number of vertices in all four building blocks: $|V(\mathrm{R}_1)| = 3$, $|V(\mathrm{R}_2)| = 4$, $|V(\mathrm{R}_3)| = 5$, $|V(\mathrm{C})| = 6$.

5. Connect the substituent R_1 to C by joining vertices r_1 and c_1, obtaining in this way CR_1.

6. Compute the Wiener index of CR_1, $W(CR_1)$, with Eq. (5):

$$W(\mathrm{CR}_1) = 27 + 3 \times 9 + 6 \times 3 + 6 \times 2 + 3 = 87 \ .$$

7. Determine the number of vertices in the subgraph CR_1:

$$|V(\mathrm{CR}_1)| = 6 + 3 = 9 \ .$$

8. Compute the distance sum of atom c_2 in the intermediate structure CR_1 with Eq. (6):

$$W_{c_2}(\mathrm{CR}_1) = 9 + (2 + 1) \times 3 + 2 = 20 \ .$$

9. Compute the distance sum of atom c_3 in the intermediate structure CR_1 with the equation:

$$W_{c_3}(\mathrm{CR}_1) = W_{c_3}(\mathrm{C}) + (d_{c_1c_3} + 1)|V(\mathrm{R}_1)| + W_{r_1}(\mathrm{R}_1)$$

$$W_{c_3}(\mathrm{CR}_1) = 9 + (3 + 1) \times 3 + 2 = 23 \ .$$

10. Connect the substituent R_2 to CR_1 by joining vertices r_2 and c_2, obtaining in this way $\mathrm{CR}_1\mathrm{R}_2$.

11. Compute the Wiener index of $\mathrm{CR}_1\mathrm{R}_2$, $W(\mathrm{CR}_1\mathrm{R}_2)$, with Eq. (7):

$$W(\mathrm{CR}_1\mathrm{R}_2) = 87 + 4 \times 20 + 9 \times 4 + 9 \times 4 + 8 = 247 \ .$$

12. Determine the number of vertices in the subgraph $\mathrm{CR}_1\mathrm{R}_2$:

$$|V(\mathrm{CR}_1\mathrm{R}_2)| = 9 + 4 = 13 \ .$$

13. Compute the distance sum of atom c_3 in the intermediate structure $\mathrm{CR}_1\mathrm{R}_2$ with the equation:

$$W_{c_3}(\mathrm{CR}_1\mathrm{R}_2) = W_{c_3}(\mathrm{CR}_1) + (d_{c_2c_3} + 1)|V(\mathrm{R}_2)| + W_{r_2}(\mathrm{R}_2)$$

$$W_{c_3}(\mathrm{CR}_1\mathrm{R}_2) = 23 + (1 + 1) \times 4 + 4 = 35 \ .$$

14. Connect the substituent R_3 to $\mathrm{CR}_1\mathrm{R}_2$ by joining vertices r_3 and c_3, obtaining in this way $\mathrm{CR}_1\mathrm{R}_2\mathrm{R}_3$.

15. Compute the Wiener index of $\mathrm{CR}_1\mathrm{R}_2\mathrm{R}_3$, $W(\mathrm{CR}_1\mathrm{R}_2\mathrm{R}_3)$, with the equation:

$$W(\mathrm{CR}_1\mathrm{R}_2\mathrm{R}_3) = W(\mathrm{CR}_1\mathrm{R}_2) + |V(\mathrm{R}_3)|W_{c_3}(\mathrm{CR}_1\mathrm{R}_2) + |V(\mathrm{CR}_1\mathrm{R}_2)||V(\mathrm{R}_3)| +$$

$$|V(\mathrm{CR}_1\mathrm{R}_2|W_{r_3}(\mathrm{R}_3) + W(\mathrm{R}_3)$$

$$W(\mathrm{CR}_1\mathrm{R}_2\mathrm{R}_3) = 247 + 5 \times 35 + 13 \times 5 + 13 \times 6 + 15 = 580 \ .$$

EVEN/ODD WIENER INDEX

In a recent study we have partitioned all graph distances into two classes, *i.e.* even and odd distances, which are separately summed to give two distinct structural descriptors, $W^{\mathrm{e}}(\mathrm{G})$ and $W^{\mathrm{o}}(\mathrm{G})$.[50] The sum of even graph distances $W^{\mathrm{e}}(\mathrm{G})$ from the molecular graph G is:

$$W^{\mathrm{e}}(\mathrm{G}) = \sum_{i,j \in V(\mathrm{G})} d_{ij}^{\mathrm{e}}(\mathrm{G}) \tag{11}$$

where d_{ij}^{e} represents an even distance between vertices i and j, and the summation goes over all even distances from G. In a similar way we define $W^{\mathrm{o}}(\mathrm{G})$, the sum of odd graph distances from the molecular graph G, by collecting together all odd graph distances:

$$W^{\mathrm{o}}(\mathrm{G}) = \sum_{i,j \in V(\mathrm{G})} d_{ij}^{\mathrm{o}}(\mathrm{G}) \tag{12}$$

where d_{ij}^{o} represents an odd distance between vertices i and j. A simple relationship exists between the Wiener index W and the even/odd indices $W^{\mathrm{e}}(\mathrm{G})$ and $W^{\mathrm{o}}(\mathrm{G})$:

$$W(\mathrm{G}) = W^{\mathrm{e}}(\mathrm{G}) + W^{\mathrm{o}}(\mathrm{G}) \tag{13}$$

We introduce four notations related to even/odd graph distances: $d_i^{\mathrm{e}}(\mathrm{G})$, the set of even graph distances from G that have vertex i as an endpoint, $d_i^{\mathrm{o}}(\mathrm{G})$, the set of odd graph distances from G that have vertex i as an endpoint, $|d_i^{\mathrm{e}}(\mathrm{G})|$, the number of distances in $d_i^{\mathrm{e}}(\mathrm{G})$, $|d_i^{\mathrm{o}}(\mathrm{G})|$, the number of distances in $d_i^{\mathrm{o}}(\mathrm{G})$. In a graph G, the sum of even distances that have vertex i as an endpoint is the even distance sum for vertex i:

$$W_i^{\mathrm{e}}(\mathrm{G}) = \sum_{j \in V(\mathrm{G})} d_{ij}^{\mathrm{e}}(\mathrm{G})\ . \tag{14}$$

Similarly, the odd distance sum for vertex i is the sum of odd distances that have vertex i as an endpoint:

$$W_i^{\mathrm{o}}(\mathrm{G}) = \sum_{j \in V(\mathrm{G})} d_{ij}^{\mathrm{o}}(\mathrm{G})\ . \tag{15}$$

From Eqs. (11) – (15) we notice a significant similarity between the Wiener index W and the even/odd variants presented in this section. Therefore, it is no surprise that for W^{e} and W^{o} one can devise two theorems, equivalent to Theorem 1, for computing the respective topological indices from subgraphs obtained by deleting a cut edge.

Theorem 2. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G = A–B as in the Figure 1. Then the even Wiener index of graph G is:

$$W^{\mathrm{e}}(\mathrm{G}) = W^{\mathrm{e}}(\mathrm{A}-\mathrm{B}) = \sum_{i,j \in V(\mathrm{A})}^{i \le j} d_{ij}^{\mathrm{e}} + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} d_{ij}^{\mathrm{e}} + \sum_{i,j \in V(\mathrm{B})}^{i \le j} d_{ij}^{\mathrm{e}} =$$

$$W^{\mathrm{e}}(\mathrm{A}) + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} (d_{ia}^{\mathrm{e}} + 1 + d_{bj}^{\mathrm{o}}) + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} (d_{ia}^{\mathrm{o}} + 1 + d_{bj}^{\mathrm{e}}) + W^{\mathrm{e}}(\mathrm{B}) =$$

$$W^{\mathrm{e}}(\mathrm{A}) + \left|d_b^{\mathrm{o}}(\mathrm{B})\right| W_a^{\mathrm{e}}(\mathrm{A}) + \left|d_a^{\mathrm{e}}(\mathrm{A})\right| \left|d_b^{\mathrm{o}}(\mathrm{B})\right| + \left|d_a^{\mathrm{e}}(\mathrm{A})\right| W_b^{\mathrm{o}}(\mathrm{B}) +$$

$$\left|d_b^{\mathrm{e}}(\mathrm{B})\right| W_a^{\mathrm{o}}(\mathrm{A}) + \left|d_a^{\mathrm{o}}(\mathrm{A})\right| \left|d_b^{\mathrm{e}}(\mathrm{B})\right| + \left|d_a^{\mathrm{o}}(\mathrm{A})\right| W_b^{\mathrm{e}}(\mathrm{B}) + W^{\mathrm{e}}(\mathrm{B})\,. \qquad (16)$$

Theorem 3. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G = A–B as in the Figure 1. Then the odd Wiener index of graph G is:

$$W^{\mathrm{o}}(\mathrm{G}) = W^{\mathrm{o}}(\mathrm{A}-\mathrm{B}) = \sum_{i,j \in V(\mathrm{A})}^{i \le j} d_{ij}^{\mathrm{o}} + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} d_{ij}^{\mathrm{o}} + \sum_{i,j \in V(\mathrm{B})}^{i \le j} d_{ij}^{\mathrm{o}} =$$

$$W^{\mathrm{o}}(\mathrm{A}) + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} (d_{ia}^{\mathrm{e}} + 1 + d_{bj}^{\mathrm{e}}) + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} (d_{ia}^{\mathrm{o}} + 1 + d_{bj}^{\mathrm{o}}) + W^{\mathrm{o}}(\mathrm{B}) =$$

$$W^{\mathrm{o}}(\mathrm{A}) + \left|d_b^{\mathrm{e}}(\mathrm{B})\right| W_a^{e}(\mathrm{A}) + \left|d_a^{e}(\mathrm{A})\right| \left|d_b^{\mathrm{e}}(\mathrm{B})\right| + \left|d_a^{e}(\mathrm{A})\right| W_b^{\mathrm{e}}(\mathrm{B}) +$$

$$\left|d_b^{\mathrm{o}}(\mathrm{B})\right| W_a^{\mathrm{o}}(\mathrm{A}) + \left|d_a^{\mathrm{o}}(\mathrm{A})\right| \left|d_b^{\mathrm{o}}(\mathrm{B})\right| + \left|d_a^{\mathrm{o}}(\mathrm{A})\right| W_b^{\mathrm{o}}(\mathrm{B}) + W^{\mathrm{o}}(\mathrm{B})\,. \qquad (17)$$

Theorems 2 and 3 afford the computation of even and odd Wiener indices for combinatorial libraries, using algorithms similar to Algorithm 1 when the final compound is made up by three building blocks, or similar to Algorithm 2 when the final compound is assembled from four or more building blocks. For such applications one needs also equations, similar to those presented in Eqs. (6), (9), and (10) for the distance sum, used to update the value of the even or odd distance sum of a substitution atom from the core structure C after each addition of a substituent R_i.

Consider the case of a compound with the general formula $\mathrm{CR}_1\mathrm{R}_2$, as depicted in Figure 2. In computing the even distance sum of vertex c_2 from CR_1 one can distinguish two cases, namely when the shortest-path distance between c_1 and c_2 is even:

$$W_{c_2}^{\mathrm{e}}(\mathrm{CR}_1) = W_{c_2}^{\mathrm{e}}(\mathrm{C}) + (d_{c_1c_2}^{\mathrm{e}} + 1)\left|d_{r_1}^{\mathrm{o}}(\mathrm{R}_1)\right| + W_{r_1}^{\mathrm{o}}(\mathrm{R}_1) \tag{18}$$

and the case when the shortest-path distance between c_1 and c_2 is odd:

$$W_{c_2}^{\mathrm{e}}(\mathrm{CR}_1) = W_{c_2}^{\mathrm{e}}(\mathrm{C}) + (d_{c_1c_2}^{\mathrm{o}} + 1)\left|d_{r_1}^{\mathrm{e}}(\mathrm{R}_1)\right| + W_{r_1}^{\mathrm{e}}(\mathrm{R}_1)\ . \tag{19}$$

A similar situation is encountered in computing the odd distance sum of vertex c_2 from CR_1, when the shortest-path distance between c_1 and c_2 is even:

$$W_{c_2}^{\mathrm{o}}(\mathrm{CR}_1) = W_{c_2}^{\mathrm{o}}(\mathrm{C}) + (d_{c_1c_2}^{\mathrm{e}} + 1)\left|d_{r_1}^{\mathrm{e}}(\mathrm{R}_1)\right| + W_{r_1}^{\mathrm{e}}(\mathrm{R}_1) \tag{20}$$

or the shortest-path distance between c_1 and c_2 is odd:

$$W_{c_2}^{\mathrm{o}}(\mathrm{CR}_1) = W_{c_2}^{\mathrm{o}}(\mathrm{C}) + (d_{c_1c_2}^{\mathrm{o}} + 1)\left|d_{r_1}^{\mathrm{o}}(\mathrm{R}_1)\right| + W_{r_1}^{\mathrm{o}}(\mathrm{R}_1)\ . \tag{21}$$

Similar equations can be developed for the general case when m substituents are added to a common core, as in Figure 3, corresponding to Eqs. (9) or (10).

RESISTANCE DISTANCE INDEX

Klein and Randić introduced a new graph distance function named the resistance distance. Applying some results from the electrical network theory,[38–41] they identified a resistance-distance matrix $\boldsymbol{\Omega}$ as an alternative to the distance matrix $\boldsymbol{D}$. For the computation of the molecular matrix $\boldsymbol{\Omega}$, Klein and Randić superposed onto the molecular graph G an electrical network of resistors, in such a way that carbon atoms become nodes in the network and carbon-carbon single bonds are represented as 1 ohm resistors; the matrix element $\boldsymbol{\Omega}_{ij}$ is equal to the effective electrical resistance between the vertices i and j. Of course this choice of 1 ohm resistors is most appropriate for hydrocarbons without multiple bonds. A more a general definition and procedure for the computation of the resistance-distance matrix $\boldsymbol{\Omega}$ of weighted molecular graphs, corresponding to organic compounds with heteroatoms and multiple bonds may be made.[51] Consider an electrical network of resistors in which a node (vertex) i corresponds to a vertex (with the same label) in the molecular graph G, while each chemical bond $\{i,j\}$ from the molecular graph is represented as a resistor between nodes i and j. Each resistor has a value $Ew_{ij}(w)$ (in ohm) depending on the chemical nature of the atoms represented by vertices i and j, and on the type of the chemical bond between them; the $Ew_{ij}(w)$ parameter is computed using the weighting schemes w proposed in the literature.[33–36] The resistance distance matrix $\boldsymbol{\Omega}(w) = \boldsymbol{\Omega}(w,\mathrm{G})$ of a graph G with N vertices is the square $N\times N$ symmetric matrix whose non-diagonal element $\boldsymbol{\Omega}_{ij}$ is equal to the electrical resistance between vertices i and j:

$$\boldsymbol{\Omega}_{ij}(w,\mathrm{G}) = \begin{cases} \boldsymbol{\Gamma}_{ij}(w,\mathrm{G}) - 2\boldsymbol{\Gamma}_{ij}(w,\mathrm{G}) + \boldsymbol{\Gamma}_{ij}(w,\mathrm{G}) & \text{if} \quad i \neq j \\ 0 & \text{if} \quad i = j \end{cases}$$

where $\boldsymbol{\Gamma}(w)$ is the generalized inverse to the w-weighted Laplacian matrix. From the theory of electrical networks it is easy to determine that in the case of acyclic hydrocarbons (*e.g.* alkanes, alkenes, alkynes), the resistance distance matrix $\boldsymbol{\Omega}$ is identical with the distance matrix $\boldsymbol{D}$. The resistance distance matrix is the source of a topological index related to W, namely the resistance distance index W':

$$W'(\mathrm{G}) = \sum_{i,j \in V(\mathrm{G})}^{i<j} \boldsymbol{\Omega}_{ij}(\mathrm{G})\ . \tag{22}$$

For the computation of the resistance distance index for molecular graphs that can be decomposed in building blocks (subgraphs) by deleting one edge one can use an equation similar to that presented in Theorem 1 for the decomposition of W:

Theorem 4. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G as in the Figure 1. Then the resistance distance index of graph G = A–B is:

$$W'(\mathrm{G}) = W'(\mathrm{A-B}) = W'(\mathrm{A}) + |V(\mathrm{B})|\{W'(\mathrm{A}-b) - W'(\mathrm{A})\} - \\ |V(\mathrm{A})||V(\mathrm{B})| + |V(\mathrm{A})|\{W'(a-\mathrm{B}) - W'(\mathrm{B})\} + W'(\mathrm{B})\ . \tag{23}$$

If we denote with $W'_i(\mathrm{G})$ the resistance distance sum for vertex i from graph G

$$W'_i(\mathrm{G}) = \sum_{j \in V(\mathrm{G})} \boldsymbol{\Omega}_{ij}(\mathrm{G}) \tag{24}$$

the resistance distance index of graph G is

$$W'(\mathrm{G}) = W'(\mathrm{A-B}) = W'(\mathrm{A}) + |V(\mathrm{B})|W'_a(\mathrm{A}) + |V(\mathrm{A})||V(\mathrm{B})| + |V(\mathrm{A})|W'_b(\mathrm{B}) + W'(\mathrm{B})\ . \tag{25}$$

An application of Theorem 4 is presented for the computation of the resistance distance index W' for the graph $\mathrm{CR_1R_2R_3}$ assembled from the core structure C and three substituents $\mathrm{R_1}$, $\mathrm{R_2}$, and $\mathrm{R_3}$, as indicated in Figure 4. The computation of the resistance distance index for $\mathrm{CR_1R_2R_3}$ is performed with a procedure which is similar to Algorithms 1 and 2:

1. Compute the resistance distance indices for the four building blocks: $W'(\mathrm{C}) = 17.5$, $W'(\mathrm{R}_1) = 2$, $W'(\mathrm{R}_2) = 5$, $W'(\mathrm{R}_3) = 10$.

2. Compute the resistance distance sums for the six connection atoms: $W'_{c_1}(\mathrm{C}) = 5.833$, $W'_{c_2}(\mathrm{C}) = 5.833$, $W'_{c_3}(\mathrm{C}) = 5.833$, $W'_{r_1}(\mathrm{R}_1) = 1.333$, $W'_{r_2}(\mathrm{R}_2) = 2.5$, $W'_{r_3}(\mathrm{R}_3) = 4$.

3. Compute the resistance distance between all pairs of connection vertices from the core structure C: $\boldsymbol{\Omega}_{c_1c_2} = 1.333$, $\boldsymbol{\Omega}_{c_1c_3} = 1.5$, $\boldsymbol{\Omega}_{c_2c_3} = 0.833$.

4. Get the number of vertices in all four building blocks: $|V(\mathrm{R}_1)| = 3$, $|V(\mathrm{R}_2)| = 4$, $|V(\mathrm{R}_3)| = 5$, $|V(\mathrm{C})| = 6$.

5. Connect the substituent R_1 to C by joining vertices r_1 and c_1, obtaining in this way CR_1.

6. Compute the resistance distance index of CR_1, $W'(\mathrm{CR}_1)$, with the equation:

$$W'(\mathrm{CR}_1) = W'(\mathrm{C}) + |V(\mathrm{R}_1)| W'_{c_1}(\mathrm{C}) + |V(\mathrm{C})||V(\mathrm{R}_1)| + |V(\mathrm{C})| W'_{r_1}(\mathrm{R}_1) + W'(\mathrm{R}_1)$$

$$W'(\mathrm{CR}_1) = 17.5 + 3 \times 5.833 + 6 \times 3 + 6 \times 1.333 + 2 = 63 \ .$$

7. Determine the number of vertices in the subgraph CR_1:

$$|V(\mathrm{CR}_1)| = 6 + 3 = 9 \ .$$

8. Compute the resistance distance sum of atom c_2 in the intermediate structure CR_1:

$$W'_{c_2}(\mathrm{CR}_1) = W'_{c_2}(\mathrm{C}) + (\boldsymbol{\Omega}_{c_1c_2} + 1)|V(\mathrm{R}_1)| + W'_{r_1}(\mathrm{R}_1)$$

$$W'_{c_2}(\mathrm{CR}_1) = 5.833 + (1.333 + 1) \times 3 + 1.333 = 14.167 \ .$$

9. Compute the resistance distance sum of atom c_3 in the intermediate structure CR_1:

$$W'_{c_3}(\mathrm{CR}_1) = W'_{c_3}(\mathrm{C}) + (\boldsymbol{\Omega}_{c_1c_3} + 1)|V(\mathrm{R}_1)| + W'_{r_1}(\mathrm{R}_1)$$

$$W'_{c_3}(\mathrm{CR}_1) = 5.833 + (1.5 + 1) \times 3 + 1.333 = 14.667 \ .$$

10. Connect the substituent R_2 to CR_1 by joining vertices r_2 and c_2, obtaining in this way $\mathrm{CR}_1\mathrm{R}_2$.

11. Compute the resistance-distance index of $\mathrm{CR}_1\mathrm{R}_2$, $W'(\mathrm{CR}_1\mathrm{R}_2)$:

$$W'(\mathrm{CR}_1\mathrm{R}_2) = W'(\mathrm{CR}_1) + |V(\mathrm{R}_2)| W'_{c_2}(\mathrm{CR}_1) + |V(\mathrm{CR}_1)||V(\mathrm{R}_2)| +$$

$$|V(\mathrm{CR}_1)| W'_{r_2}(\mathrm{R}_2) + W'(\mathrm{R}_2)$$

$$W'(\mathrm{CR_1R_2}) = 63 + 4\times 14.167 + 9\times 4 + 9\times 2.5 + 5 = 183.167\ .$$

12. Determine the number of vertices in the subgraph $\mathrm{CR_1R_2}$:

$$|V(\mathrm{CR_1R_2})| = 9 + 4 = 13\ .$$

13. Compute the resistance distance sum of atom c_3 in the intermediate structure $\mathrm{CR_1R_2}$:

$$W'_{c_3}(\mathrm{CR_1R_2}) = W'_{c_3}(\mathrm{CR_1}) + (\Omega_{c_2c_3} + 1)|V(\mathrm{R_2})| + W'_{r_2}(\mathrm{R_2})$$

$$W'_{c_3}(\mathrm{CR_1R_2}) = 14.667 + (0.833 + 1)\times 4 + 2.5 = 24.5\ .$$

14. Connect the substituent $\mathrm{R_3}$ to $\mathrm{CR_1R_2}$ by joining vertices r_3 and c_3, obtaining in this way $\mathrm{CR_1R_2R_3}$.

15. Compute the resistance distance index of $\mathrm{CR_1R_2R_3}$, $W'(\mathrm{CR_1R_2R_3})$:

$$W'(\mathrm{CR_1R_2R_3}) = W'(\mathrm{CR_1R_2}) + |V(\mathrm{R_3})| + W'_{c_3}(\mathrm{CR_1R_2}) + |V(\mathrm{CR_1R_2})||V(R_3)| + |V(\mathrm{CR_1R_2})|W'_{r_3}(\mathrm{R_3}) + W'(\mathrm{R_3})$$

$$W'(\mathrm{CR_1R_2R_3}) = 183.167 + 5\times 24.5 + 13\times 5 + 13\times 4 + 10 = 432.667\ .$$

WIENER POLYNOMIAL

Hosoya used distances between pairs of graph vertices to define the Wiener polynomial, whose first derivative gives the Wiener index W;[52] several years later, an equivalent formula was proposed by a group of mathematicians.[53] The higher derivatives of the Wiener polynomial give a sequence of Wiener-type topological indices used with success in several QSPR models.[54–56] Let $d(\mathrm{G},k)$ be the number of pairs of vertices in G that are distance k apart, and denote the largest element of $\boldsymbol{D}$ by l, often called the diameter of G; obviously, $d(\mathrm{G},1)$ is equal to the number of edges. The Wiener polynomial of a graph G is defined by the following equation:

$$H(\mathrm{G},x) = \sum_{k=1}^{l} d(\mathrm{G},k)x^k = \sum_{i,j\in V(\mathrm{G})}^{i<j} x^{d_{ij}}\ . \tag{26}$$

The previously mentioned relationship between the Wiener polynomial and W is readily identified after computing the first derivative of H for $x = 1$:

$$\left[\frac{\mathrm{d}H(\mathrm{G},x)}{\mathrm{d}x}\right]_{x=1} = W(\mathrm{G})\ . \tag{27}$$

With H_i(G,x) we denote the vertex Wiener polynomial for the vertex $i \in V$(G). This graph polynomial collects all H(G,x) terms that have vertex i as an endpoint:

$$H_i(\mathrm{G},x) = \sum_{j \in V(\mathrm{G})} x^{d_{ij}} \ . \tag{28}$$

The Wiener polynomial of a graph can be computed from distance-based graph invariants of its subgraphs obtained after deleting a cut edge:

Theorem 5. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G as in the Figure 1. Then the Wiener polynomial H of graph G = A–B is

$$\begin{aligned} H(\mathrm{G},x) = H(\mathrm{A}-\mathrm{B},x) &= \sum_{i,j \in V(\mathrm{A})}^{i<j} x^{d_{ij}} + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} x^{d_{ij}} + \sum_{i,j \in V(\mathrm{B})}^{i<j} x^{d_{ij}} = \\ & H(\mathrm{A},x) + \sum_{i \in V(\mathrm{A})} \sum_{j \in V(\mathrm{B})} x^{d_{ia}+1+d_{bj}} + H(\mathrm{B},x) = \\ & H(\mathrm{A},x) + xH_a(\mathrm{A},x) H_b(\mathrm{B},x) + H(\mathrm{B},x) \ . \end{aligned} \tag{29}$$

Using the decomposition method presented in Theorem 5 one can for different chosen values of the variable x compute the Wiener polynomial $H(x)$ and various related topological indices very efficiently. This is especially true for generating topological indices for combinatorial libraries, when one can use an algorithm similar to Algorithm 1 when the final compound contains three building blocks, or similar to Algorithm 2 when the final compound is assembled from four or more building blocks. These algorithms must update the value of the vertex Wiener polynomial of a substitution atom from the core structure C after each addition of a substituent R_i. Consider the case of a compound with the general formula $\mathrm{CR_1R_2}$, as depicted in Figure 2. The vertex Wiener polynomial of vertex c_2 from $\mathrm{CR_1}$ is computed with the equation:

$$H_{c_2}(\mathrm{CR}_1,x) = H_{c_2}(\mathrm{C},x) + x^{d_{c_1c_2}+1} H_{r_1}(\mathrm{R}_1,x) \ . \tag{30}$$

A similar equation must be used in the general case when m substituents are added to a common core. From the recursion of (29) one can obtain recursions for the first (or higher) derivatives of H(G,x). *E. g.*, the derivative recursion at $x = 1$ gives the Wiener-number recursion, and the second-derivative recursion evaluated at $x = 1$ leads to a recursion for Randić's hyper-Wiener index[57] (as generalized beyond trees in Ref. 58).

EVEN/ODD WIENER POLYNOMIAL

A comparison of the formula of W from Eq. (2) and of H from Eq. (26) suggests that the procedure of separating the Wiener index W into even and odd terms can be applied also to the Wiener polynomial. The even Wiener polynomial H^{e} collects all even graph distances:[59]

$$H^{\mathrm{e}}(\mathrm{G},x) = \sum_{i,j \in V(\mathrm{G})}^{even} x^{d_{ij}^{e}} \tag{31}$$

where the summation goes over all even graph distances d_{ij}^{e}, and the variable x is an optimizable parameter. Similarly, the odd Wiener polynomial H^{o} collects all odd graph distances:[59]

$$H^{\mathrm{o}}(\mathrm{G},x) = \sum_{i,j \in V(\mathrm{G})}^{odd} x^{d_{ij}^{\mathrm{o}}} \tag{32}$$

where the summation goes over all odd graph distances d_{ij}^{o}, and the variable x is a parameter that can be optimized for each QSAR/QSPR property and data set of chemical compounds. The value of H^{e} and H^{o} for a given value of the variable x gives two novel optimizable topological indices that were used with success in QSPR studies.[59] The first derivative of the even Wiener polynomial H^{e} for $x = 1$ is equal to the even Wiener index W^{e}:

$$\left[\frac{\mathrm{d}H^{\mathrm{e}}(\mathrm{G},x)}{\mathrm{d}x}\right]_{x=1} = W^{\mathrm{e}}(\mathrm{G}) \tag{33}$$

while the first derivative of the odd Wiener polynomial H^{o} for $x = 1$ is equal to the odd Wiener index W^{o}:

$$\left[\frac{\mathrm{d}H^{\mathrm{o}}(\mathrm{G},x)}{\mathrm{d}x}\right]_{x=1} = W^{\mathrm{o}}(\mathrm{G})\ . \tag{34}$$

The even vertex Wiener polynomial for the vertex $i \in \mathrm{V(G)}$, $H_i^{\mathrm{e}}(\mathrm{G},x)$, collects all even-distance $H(\mathrm{G}, x)$ terms that have vertex i as an endpoint:

$$H_i^{\mathrm{e}}(\mathrm{G},x) = \sum_{j \in V(\mathrm{G})} x^{d_{ij}^{\mathrm{e}}} \tag{35}$$

while the odd vertex Wiener polynomial for the vertex $i \in V(G)$, $H_i^{\mathrm{o}}(\mathrm{G},x)$, collects all odd-distance $H(\mathrm{G}, x)$ terms that have vertex i as an endpoint:

$$H_i^{\mathrm{o}}(\mathrm{G},x) = \sum_{j \in V(\mathrm{G})} x^{d_{ij}^{\mathrm{o}}}\ . \tag{36}$$

A comparison of Eqs. (11) and (12) that define the even/odd Wiener indices with Eqs. (31) and (32) that define the even/odd Wiener polynomials, reveals an important similarity between them, indicating that we can develop theorems for computing the even and odd Wiener polynomials of graphs decomposed at a cut edge.

Theorem 6. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G as in the Figure 1. Then the even Wiener polynomial of graph G = A–B is

$$H^{\mathrm{e}}(\mathrm{G},x) = H^{\mathrm{e}}(\mathrm{A}-\mathrm{B},x) = \sum_{i,j\in V(\mathrm{A})}^{i<j} x^{d^{\mathrm{e}}_{ij}} + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{e}}_{ij}} + \sum_{i,j\in V(\mathrm{B})}^{i<j} x^{d^{\mathrm{e}}_{ij}} =$$

$$H^{\mathrm{e}}(\mathrm{A},x) + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{e}}_{ia}+1+d^{\mathrm{o}}_{bj}} + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{o}}_{ia}+1+d^{\mathrm{e}}_{bj}} + H^{\mathrm{e}}(\mathrm{B},x) =$$

$$H^{\mathrm{e}}(\mathrm{A},x) + xH^{\mathrm{e}}_{a}(\mathrm{A},x)H^{\mathrm{o}}_{b}(\mathrm{B},x) + xH^{\mathrm{o}}_{a}(\mathrm{A},x)H^{\mathrm{e}}_{b}(\mathrm{B},x) + H^{\mathrm{e}}(\mathrm{B},x)\ . \qquad (37)$$

Theorem 7. Let $\{a,b\}$ be a cut edge between two subgraphs A and B of G such that $a \in$ A and $b \in$ B. Denote various subgraphs of G as in the Figure 1. Then the odd Wiener polynomial of graph G = A–B is

$$H^{\mathrm{o}}(\mathrm{G},x) = H^{\mathrm{o}}(\mathrm{A}-\mathrm{B},x) = \sum_{i,j\in V(\mathrm{A})}^{i<j} x^{d^{\mathrm{o}}_{ij}} + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{o}}_{ij}} + \sum_{i,j\in V(\mathrm{B})}^{i<j} x^{d^{\mathrm{o}}_{ij}} =$$

$$H^{\mathrm{o}}(\mathrm{A},x) + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{e}}_{ia}+1+d^{\mathrm{e}}_{bj}} + \sum_{i\in V(\mathrm{A})}\sum_{j\in V(\mathrm{B})} x^{d^{\mathrm{o}}_{ia}+1+d^{\mathrm{o}}_{bj}} + H^{\mathrm{o}}(\mathrm{B},x) =$$

$$H^{\mathrm{o}}(\mathrm{A},x) + xH^{\mathrm{e}}_{a}(\mathrm{A},x)H^{\mathrm{e}}_{b}(\mathrm{B},x) + xH^{\mathrm{o}}_{a}(\mathrm{A},x)H^{\mathrm{o}}_{b}(\mathrm{B},x) + H^{\mathrm{o}}(\mathrm{B},x)\ . \qquad (38)$$

Similarly with the other formulas proposed in this paper, Theorems 6 and 7 are very efficient in computing even and odd Wiener polynomials and corresponding topological indices for molecules that have one or more cut edges. This is especially true for cases when a finite, small number of building blocks are combined to generate a huge number of chemical compounds. If the final compound consists of three building blocks the procedure is similar to Algorithm 1, while for compound assembled from four or more building blocks the procedure can be derived from Algorithm 2. In both cases the algorithms use a group of equations for updating the even or odd vertex Wiener polynomial of a substitution atom from the core structure C after each addition of a substituent R_i. We present the corresponding equations derived for a molecular graph with the general formula $\mathrm{CR}_1\mathrm{R}_2$, as presented in Figure 2. The computation of the even vertex Wiener polynomial for vertex c_2 from CR_1 has two possibilities, that is when the shortest-path distance between c_1 and c_2 is even:

$$H^{\mathrm{e}}_{c_2}(\mathrm{CR}_1, x) = H^{\mathrm{e}}_{c_2}(\mathrm{C}, x) + x^{d^{e}_{c_1c_2}+1} H^{\mathrm{o}}_{r_1}(\mathrm{R}_1, x) \tag{39}$$

and when the shortest-path distance between c_1 and c_2 is odd:

$$H^{\mathrm{e}}_{c_2}(\mathrm{CR}_1, x) = H^{\mathrm{e}}_{c_2}(\mathrm{C}, x) + x^{d^{\mathrm{o}}_{c_1c_2}+1} H^{\mathrm{e}}_{r_1}(\mathrm{R}_1, x)\ . \tag{40}$$

Analogously, in computing the odd vertex Wiener polynomial of vertex c_2 from CR_1, the shortest-path distance between c_1 and c_2 can be even:

$$H^{\mathrm{o}}_{c_2}(\mathrm{CR}_1, x) = H^{\mathrm{o}}_{c_2}(\mathrm{C}, x) + x^{d^{\mathrm{e}}_{c_1c_2}+1} H^{\mathrm{e}}_{r_1}(\mathrm{R}_1, x) \tag{41}$$

or odd:

$$H^{\mathrm{o}}_{c_2}(\mathrm{CR}_1, x) = H^{\mathrm{o}}_{c_2}(\mathrm{C}, x) + x^{d^{\mathrm{o}}_{c_1c_2}+1} H^{\mathrm{o}}_{r_1}(\mathrm{R}_1, x)\ . \tag{42}$$

Similar equations can be obtained for the general case when m substituents are added to a common core, making this procedure general enough to be applied in VSCL.

CONCLUSIONS

Graph invariants and topological indices are extensively used in QSPR, SAR and QSAR studies as effective numerical descriptors of the chemical structure. Their recent applications in the investigation of chemical libraries revealed their efficiency in quantifying the similarity, diversity, and the drug-like character of chemicals. In all QSPR and QSAR studies the topological indices are computed with efficient numerical methods.[60] These numerical methods have the advantage of being general, *i.e.* one can use them for any chemical structure, but often they involve significant computational resources. For example, the computation of the distance matrix from the adjacency matrix requires $O(N^3)$ operations, per N-atom molecule. As an alternative to the computation of topological indices with numerical methods, mathematical chemists developed recurrence relationships or subgraph decomposition equations, which calculate a graph invariant from the invariants of selected subgraphs, usually obtained after deleting a vertex, edge, or cycle. However, these subgraph decomposition equations have not previously found an application in computing topological indices for QSAR studies. The main reason evidently is the relatively small number of compounds, usually between 10 and 100, in the QSAR studies. In such cases the topological indices are obtained with numerical methods because this is the most straightforward and general procedure, and the computation time is not critical.

However, the process of screening virtual and synthetic combinatorial libraries has some special characteristics that make attractive and time-effective the computation of topological indices with subgraph decomposition equations. Using a relative small number of reactants that are combined following a general reaction scheme, a virtual library can easily exceed 10^6 compounds. Therefore, the time needed to compute the structural descriptors becomes $10^6 \times O(N^3)$, and thence is critical, so that recent investigations have been dedicated to the development of simplified algorithms for the calculation of descriptors, such as the molecular polar surface area approximated as the sum of N, O, P and S atoms contributions,[29] or the van der Waals molecular surface area computed from the atomic connectivity information.[30] In this paper we have demonstrated several subgraph decomposition equations for distance-based topological indices and graph invariants: the Wiener index, even/odd Wiener index, resistance distance index, Szeged index, Wiener polynomial, and even/odd Wiener polynomial. We have proposed a simple and fast algorithm for the computation of these topological indices for very large virtual combinatorial libraries without actually assembling the individual compounds from the building blocks. The procedure uses numerical methods to compute distance-based graph invariants only for the building blocks. The values are stored and used to compute topological indices for the reaction products with the subgraph decomposition equations proposed in this paper. In this way, the most expensive step in computing distance-based topological indices, namely the generation of the distance matrix, is skipped. The time investment proportional to the total number of compounds scales as $O(m^2)$ when there are m substituent positions around a core. We further note that such recursions as we have developed here may be used to compute averages, standard deviations, and perhaps even extremal values of suitable linear combinations representing selected properties, all without repetitions for each compound – see, *e.g.*, Refs. 24–27 for examples. Similar algorithms can be developed for connectivity indices and vertex- and edge-weighted molecular graphs.[61,62] Such directions are to be addressed in future investigations.

Acknowledgment. – The authors acknowledge the financial support of this research by the Welch Foundation of Houston, Texas.

REFERENCES

1. J.-L. Fauchère, J. A. Boutin, J.-M. Henlin, N. Kucharczyk, and J.-C. Ortuno, *Chemom. Intell. Lab. Syst.* **43** (1998) 43–68.
2. D. K. Agrafiotis, J. C. Myslik, and F. R. Salemme, *Mol. Diversity* **4** (1999) 1–22.
3. O. Ivanciuc and A. T. Balaban, *The Graph Description of Chemical Structures*, in: J. Devillers and A. T. Balaban (Eds.), *Topological Indices and Related Descriptors*

in QSAR and QSPR, Gordon and Breach Science Publishers, Amsterdam, 1999, pp. 59–167.

4. O. Ivanciuc and T. Ivanciuc, *Matrices and Structural Descriptors Computed from Molecular Graph Distances*, in: J. Devillers and A. T. Balaban (Eds.), *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach Science Publishers, Amsterdam, 1999, pp. 221–277.
5. H. Wiener, *J. Am. Chem. Soc*. **69** (1947) 17–20.
6. H. Wiener, *J. Am. Chem. Soc*. **69** (1947) 2636–2638.
7. M. Randić, *J. Am. Chem. Soc*. **97** (1975) 6609–6615.
8. L. B. Kier and L. H. Hall, *Molecular Connectivity in Structure-Activity Analysis*, Research Studies Press, Letchworth, 1986.
9. A. T. Balaban, *Chem. Phys. Lett*. **89** (1982) 399–404.
10. A. T. Balaban, *Pure Appl. Chem*. **55** (1983) 199–206.
11. D. Plavšić, S. Nikolić, N. Trinajstić, and Z. Mihalić, *J. Math. Chem*. **12** (1993) 235–250.
12. O. Ivanciuc, T.-S. Balaban, and A. T. Balaban, *J. Math. Chem*. **12** (1993) 309–318.
13. Ajay, W. P. Walters, and M. A. Murcko, *J. Med. Chem*. **41** (1998) 3314–3324.
14. J. Sadowski and H. Kubinyi, *J. Med. Chem*. **41** (1998) 3325–3329.
15. A. K. Ghose, V. N. Viswanadhan, and J. J. Wendoloski, *J. Comb. Chem*. **1** (1999) 55–68.
16. T. I. Oprea, *J. Comput.-Aided Mol. Design* **14** (2000) 251–264.
17. L. Xue and J. Bajorath, *J. Chem. Inf. Comput. Sci*. **40** (2000) 801–809.
18. W. Zheng, S. J. Cho, and A. Tropsha, *Int. J. Quantum Chem*. **69** (1998) 65–75.
19. G. Grassy, B. Calas, A. Yasri, R. Lahana, J. Woo, S. Iyer, M. Kaczorek, R. Floc'h, and R. Buelow, *Nature Biotechnol*. **16** (1998) 748–752.
20. A. Linusson, J. Gottfries, F. Lindgren, and S. Wold, *J. Chem. Inf. Comput. Sci*. **40** (2000) 1320–1328.
21. Ajay, G. W. Bemis, and M. A. Murcko, *J. Med. Chem*. **42** (1999) 4942–4951.
22. T. I. Oprea and J. Gottfries, *J. Comb. Chem*. **3** (2001) 157–166.
23. S. Shi, Z. Peng, J. Kostrowicki, G. Paderes, and A. Kuki, *J. Mol. Graphics Modell*. **18** (2000) 478–496.
24. L. Bytautas and D. J. Klein, *J. Chem. Inf. Comput. Sci.* **38** (1998) 1063–1078.
25. L. Bytautas and D. J. Klein, *J. Chem. Inf. Comput. Sci.* **39** (1999) 803–818.
26. L. Bytautas, D. J. Klein, and T. G. Schmalz, *New J. Chem*. **24** (2000) 329–336.
27. L. Bytautas and D. J. Klein, *J. Chem. Inf. Comput. Sci.* **40** (2000) 471–481.
28. S. A. Wildman and G. M. Crippen, *J. Chem. Inf. Comput. Sci*. **39** (1999) 868–873.
29. P. Ertl, B. Rohde, and P. Selzer, *J. Med. Chem*. **43** (2000) 3714–3717.
30. P. Labute, *J. Mol. Graph. Modell*. **18** (2000) 464–477.
31. T. Österberg and U. Norinder, *J. Chem. Inf. Comput. Sci*. **40** (2000) 1408–1411.
32. T. I. Oprea, *SAR QSAR Environ. Res*. **12** (2001) 129–141.
33. O. Ivanciuc, T. Ivanciuc, and A. T. Balaban, *J. Chem. Inf. Comput. Sci*. **38** (1998) 395–401.
34. O. Ivanciuc, *J. Chem. Inf. Comput. Sci*. **40** (2000) 1412–1422.
35. O. Ivanciuc, *Rev. Roum. Chim*. **45** (2000) 289–301.
36. O. Ivanciuc, T. Ivanciuc, and A. T. Balaban, *Vertex- and Edge-Weighted Molecular Graphs and Derived Structural Descriptors*, in: J. Devillers and A. T. Balaban (Eds.), *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach Science Publishers, Amsterdam, 1999, pp. 169–220.
37. H. Hosoya, *Bull. Chem. Soc. Jpn.* **44** (1971) 2332–2339.

38. D. J. Klein and M. Randić, *J. Math. Chem.* **12** (1993) 81–95.
39. H.-Y. Zhu and D. J. Klein, *J. Chem. Inf. Comput. Sci.* **36** (1996) 1067–1075.
40. D. J. Klein, *MATCH* (*Commun. Math. Comput. Chem.*) **35** (1997) 7–27.
41. D. J. Klein and H.-Y. Zhu, *J. Math. Chem.* **23** (1998) 179–195.
42. I. Gutman, *Graph Theory Notes New York* **27** (1994) 9–15.
43. I. Gutman and A. A. Dobrynin, *Graph Theory Notes New York* **34** (1998) 37–44.
44. M. V. Diudea, *J. Chem. Inf. Comput. Sci.* **37** (1997) 292–299.
45. M. V. Diudea, *J. Chem. Inf. Comput. Sci.* **37** (1997) 300–305.
46. M. V. Diudea and I. Gutman, *Croat. Chem. Acta* **71** (1998) 21–51.
47. O. E. Polansky and D. Bonchev, *MATCH* (*Commun. Math. Chem.*) **21** (1986) 133–186.
48. M. V. Diudea and I. E. Kacsó, *MATCH* (*Commun. Math. Chem.*) **26** (1991) 255–269.
49. D. Bonchev, L. Bytautas, and D. J. Klein, *On the Wiener Number of Proper (or Thorny) Graphs*, Second Indo-US Workshop on Mathematical Chemistry, May 30 – June 3, 2000, University of Minnesota, Duluth, Minnesota, USA.
50. O. Ivanciuc, T. Ivanciuc, D. J. Klein, W. A. Seitz, and A. T. Balaban, *J. Chem. Inf. Comput. Sci.* **41** (2001) 536–549.
51. O. Ivanciuc, *ACH – Models Chem.* **137** (2000) 607–631.
52. H. Hosoya, *Discrete Appl. Math.* **19** (1988) 239–257.
53. B. E. Sagan, Y.-N. Yeh, and P. Zhang, *Int. J. Quantum Chem.* **60** (1996) 959–969.
54. E. Estrada, O. Ivanciuc, I. Gutman, A. Gutierrez, and L. Rodríguez, *New J. Chem.* **22** (1998) 819–822.
55. I. Gutman, E. Estrada, and O. Ivanciuc, *Graph Theory Notes New York* **36** (1999) 7–13.
56. E. V. Konstantinova and M. V. Diudea, *Croat. Chem. Acta* **73** (2000) 383–403.
57. M. Randić, *Chem. Phys. Lett.* **211** (1993) 478–483.
58. D. J. Klein, I. Lukovits, and I. Gutman, *J. Chem. Inf. Comput. Sci.* **35** (1995) 50–52.
59. O. Ivanciuc, T. Ivanciuc, and D. J. Klein, *SAR QSAR Environ. Res.* **12** (2001) 1–16.
60. O. Ivanciuc and J. Devillers, *Algorithms and Software for the Computation of Topological Indices and Structure-Property Models*, in: J. Devillers and A. T. Balaban (Eds.), *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach Science Publishers, Amsterdam, 1999, pp. 779–804.
61. O. Ivanciuc and D. J. Klein, *J. Chem. Inf. Comput. Sci.* **42** (2002) 8–22.
62. O. Ivanciuc, *Internet Electron. J. Mol. Des.* **1** (2002) 1–9, http://www.biochempress.com.

SAŽETAK

Računanje indeksa Wienerova tipa izgradnjom blokova za virtualno pretraživanje kombinatoričkih knjižnica

Ovidiu Ivanciuc i Douglas J. Klein

Pretraživanje virtualnih i sintetskih kombinatoričkih knjižnica može omogućiti brže pronalaženje lijekova, osobito ako se odaberu podskupovi molekula po njihovoj sličnosti (ili različnosti) specifičnim kolekcijama spojeva. Topologijski indeksi, računani iz atomske povezanosti ili udaljenosti grafova sve se više rabe kao strukturni deskriptori kako bi se povećao skup različitih klasa molekula u knjižnicama spojeva ili da bi se kvantificirala biološka svojstva razmatranih spojeva. U ovome članku autori su izveli djelotvorne jednadžbe za računanje nekoliko topologijskih indeksa, koje se temelje na udaljenostima u molekulskom grafu, iz invarijanata udaljenosti njihovih podgrafova. S pomoću tih jednadžbi moguće je djelotvorno računati za nevagane molekulske grafove: Wienerov indeks, parni/neparni Wienerov indeks, indeks otporne udaljenosti, Wienerov polinom i parni/neparni Wienerov polinom. Rabeći jednostavan i brzi algoritam može se izračunati navedene topologijske indekse za vrlo velik broj virtualnih kombinatoričkih knjižnica bez računanja indeksa za svaki pojedinačni spoj – već samo indekse za strukturne blokove koji se onda upotrebljavaju za generiranje indeksa spojeva sastavljenih od tih blokova.