Enumeration of Polyhex Hydrocarbons to $h = 21$

Gilles Caporossi

Départment de Mathématiques et Génie Industriel, École Polytechnique de Montréal, Canada

Pierre Hansen

GERAD and Départment de Méthodes Quantitatives en Gestion, École des Hautes Études Commerciales, Montréal, Canada

Received December 18, 1997

An algorithm based on the boundary-edges code and the reverse search method is proposed for enumerating nonisomorphic planar simply connected polyhexes. These polyhexes are associated with vertices of a graph whose edges correspond to addition of a hexagon. A directed tree is defined on this graph. To this effect, a new father−son relationship is introduced: the father is the polyhex obtained when removing the hexagon associated with the first digit of the son's code. Then testing if a generated polyhex is a legitimate one in the enumeration can be done easily and efficiently. The resulting algorithm is used to enumerate polyhexes with $h \leq 21$ hexagons, a set of over one trillion molecules, which is >600 times larger than previously done.

INTRODUCTION

Counting and enumeration of molecules of various families have been important activities in mathematical chemistry for a long time. For surveys, including history of this endeavor, one may consult Balaban,[1] Bababan *et al.*,[2] Balasubramanian,[3] Bonchev and Rouvray,[4,5] Brunvoll *et al.*,[6] Dias,[7] Gutman and Cyvin,[8] Hosoya,[9] Knop *et al.*,[10] Mercier *et al.*,[11] and Trinajstić *et al.*[12,13] The two activities are distinct: counting elements of a set means finding how many there are, whereas enumerating them means providing in addition a concise description of each of them (i.e., a code). Information provided by enumeration is thus greater and more useful than that provided by mere counting. Indeed, when studying some property described by an invariant, the codes for elements of the set considered can be used to determine minimum, average, and maximum values for this invariant. Moreover, if this set is extremely large, one can use such codes to select a subset at random and then find a good estimate for the average value of the invariant.

Enumerating polyhex hydrocarbons is a much studied problem since the 1960s. The exponential growth of the number of polyhexes with the number of hexagons (denoted by h) makes this problem a good benchmark for enumeration methods. The first computer-oriented algorithm proposed for its resolution is by Balasubramanian *et al.*,[14] and is based on the *boundary code*, which appears to be the first computer code for polyhexes but is redundant and rather difficult to handle. The enumeration of planar simply connected polyhexes to $h = 10$,[15] $h = 11$,[16] and $h = 12$[17] used this algorithm. The next advance in polyhex enumeration came from the DAST *(Dualist Angle-restricted Spanning Tree)* code,[18] which is based on the dualist graph associated with every polyhex.[19] This tool is much more powerful than the boundary code and allowed enumeration of all polyhexes for $h = 13$,[20] $h = 14$,[21] $h = 15$,[18] and $h = 16$.[22] However, a certain weakness remains as every polyhex may have up to 12 different representations depending on the hexagon used as root of the dualist tree and the orientation of the polyhex. To avoid multiple enumeration, a special rule has to be used.[12] The last progress in the field was made by Tošić *et al.*[23] who proposed an original approach using a *"cage"* within which the polyhexes are placed. Their method appears to be based implicitly on a code using Cartesian positions like the *Wiswesser code* for polyhexes.[24] This method led to enumeration of all polyhexes with $h = 17$. Little more may be said about the efficiency of the algorithm of Tošić *et al.*[23] because computation times are not given.

The aim of the present paper is not only to give the number of polyhexes for $h = 18$, 19, 20, and 21 and related quantitative information, but also to introduce and illustrate new ideas in the enumeration of polyhexes. We propose an algorithm based on the reverse search method for enumeration proposed by Avis and Fukuda.[25] This method uses an enumeration tree and a specific rule to avoid duplication, as shall be explained later. The code used is the boundary-edges code (or BEC code; Hansen et al.[26] and Herndon and Bruce[27]). As stated by these last authors, this code has great potential for enumeration. This potential is particularly important if the code is combined with reverse search, as shown later.

DEFINITIONS

A **polyhex** (or benzenoid system) is a connected system of regular hexagons such that any two hexagons either share exactly one edge or are disjoint.[22,23] A **helicene** as opposed to a **planar polyhex,** is a polyhex that has at least two overlapping edges (Figure 1b). Note that even if a polyhex is planar, the corresponding molecule may have a twisted nonplanar geometry (see Herndon *et al.*[28] for a discussion). A **coronoid** is a polyhex that has at least one hole (Figure 1c), as opposed to a **simply connected** polyhex that has no hole at all (Figure 1a). Focusing on the incidences between

S0095-2338(97)00116-9 CCC: $15.00
Published on Web 05/30/1998

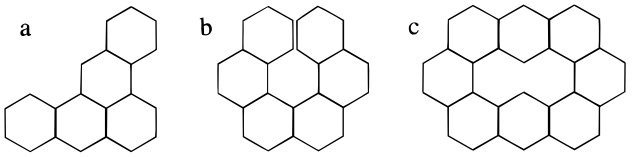


Figure 1. Examples of planar simply connected polyhex (a), helicene (b), and coronoid (c).

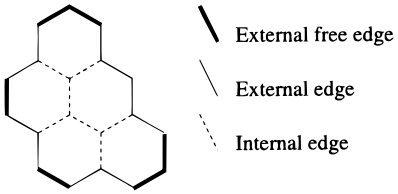
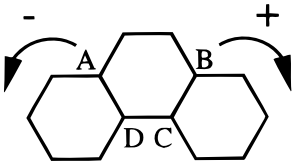


Figure 2. Edges of a polyhex.



	-	+
A	5153	3515
B	3515	5153
C	5351	1535
D	1535	5351

Figure 3. Construction of the BEC code.

edges and vertices, polyhexes can be viewed as graphs. The **degree of a vertex** in a graph is the number of edges adjacent to that vertex. In a polyhex, it is equal to 2 or 3. An **external edge** of a simply connected polyhex is an edge that belongs to a single hexagon (Figure 2). A **free edge** is an external edge with two vertices of degree 2 (Figure 2). The **boundary** of a polyhex is the cycle described by its external edges. An **external vertex** is a vertex belonging to the boundary of the polyhex. A **tree** is a connected graph without cycles. An **arborescence** is a directed tree such that each vertex is the initial one of exactly one directed edge, except for one vertex called the **root**. In a **reversed arborescence,** orientations of edges are reversed.

The **Boundary Edges Code** (BEC) is defined as follows for simply connected polyhexes (in which case it is equivalent to the PC-2 code[27]). Beginning at any external vertex of degree 3, which thus belongs to only two hexagons, follow the boundary of the polyhex noting by a digit the number of edges on the boundary for each successive hexagon encountered (the same hexagon may appear up to three times on the boundary, and hence may correspond to 1, 2, or 3 digits in the code). Then apply, if needed, a circular shift and/or reversal of the code to make it lexicographically maximum. Construction of the BEC code of a polyhex is illustrated in Figure 3. Observe that the code is unique but may be obtained in several ways in case of symmetry of the polyhex.

Proposition 1: *The BEC code of a polybex always begins with a digit greater than or equal to* 3.

Proof: Gutman and Cyvin[8] have shown that the number of free edges of a polyhex equals $6 + b$, where b is the number of external edges with two vertices of degree 3 in the polyhex (such edges are represented by a "1" in the BEC code). If the first digit of the code is ≤ 2, then there is no digit >2 in this code, as it is lexicographically maximum. Therefore no hexagon has free edges, which contradicts the property already mentioned.

For simplicity, the term "polyhex" will be used from now on instead of "planar simply connected polyhex". Furthermore, we will use "first hexagon" instead of "hexagon represented by the first digit of the code".

PRINCIPLES AND ALGORITHM

As in most polyhex enumeration methods, we proceed by successive additions of hexagons. This can be done in a depth-first way, adding hexagons until the desired number h_{max} is attained, or in a breadth-first way, building all polyhexes with a given number h of hexagons from those with $h - 1$ hexagons and so on. In both cases, there may be duplications because the same polyhex with h hexagons may be obtained from several ones with $h - 1$ hexagons.

To avoid repetitions, each polyhex with h hexagons is legitimately generated from one and only one polyhex with $h - 1$ hexagons using the reverse search method of Avis and Fukuda.[25] Validation (*i.e.*, checking) if a polyhex has been generated from its father, a specific polyhex with $h - 1$ hexagons is easy and quick due to a very efficient choice of a father−son relationship. The method is simple and avoids the use of long lists of already encountered polyhexes.

Avis and Eukuda's Reverse Search Method. Avis and Fukuda[25] introduced the *reverse search* method to solve efficiently a classical problem of operations research and computational geometry: enumeration of all vertices of a polytope (the method easily extends to the case of enumeration of all extreme points and extreme rays of a polyhedron). Algorithms for this problem explore in various ways the *adjacency graph* of the polytope. This graph has a set of vertices and a set of edges in one-to-one correspondence with vertices and edges of the polytope. In other words, it is obtained from the sets of vertices and edges by retaining topological (*i.e.*, incidence and adjacency) properties and neglecting metric ones. Usually, the adjacency graph is explored by *depth-first search* (see e.g., Aho et al.[29] for a description of this technique). The question is then, when arriving at a vertex to know whether it has already been explored or not. This problem was solved, for over 25 years, by keeping a long list of explored vertices and checking if the new vertex is present or not in that list (e.g., Dyer[30]). Such a procedure is both space and time consuming, as the number of vertices in a polytope may be very large even if its dimension is small (typically over one million for 10 variables). An alternate algorithmic scheme was proposed by Chen et al.[31] This scheme relies on adjacency lists between vertices, obtained first by enclosing the polytope in a simplex then adding to this simplex facets of the polytope one at a time and updating the lists of vertices and adjacencies. It is, however, still necessary to keep a fairly long list of vertices in the last added facet to determine their adjacencies.

In contrast, Avis and Fukuda[25] use a simple but powerful observation to avoid all intermediary lists: assume that a (reversed) arborescence can be defined on the adjacency

graph (or, in other words, that exactly one follower may be assigned to each vertex, except for a single one, the root). This can be done for the vertex enumeration problem by using Bland's[32] rule for the choice of entering and leaving variables, in a version of the simplex algorithm which is guaranteed to converge even in case of degeneracy. Then, exploration of the adjacency graph is done by depth-first search from the root. When arriving at a vertex, it is checked by "reversing the search" whether the vertex is reached from its follower. If yes, the vertex is viewed as legitimate and recorded. Otherwise, backtracking occurs. It was quickly realized that the reverse search method could be applied to many enumeration problems from combinatorial geometry and other fields.

In the case of polyhex enumeration, a graph is obtained by associating a vertex to each polyhex and including an edge from a vertex to another one if and only if the polyhex associated to the latter can be constructed by adding an hexagon to the polyhex associated to the former. Building a directed (reversed) arborescence (called enumeration tree) on this graph is discussed in the next subsection.

As observed by Gunnar Brinkmann, this enumeration method can also be viewed as an application of "orderly generation", already and independently proposed in the seventies by Read[33] and Faradzev;[34,35] see also Brinkmann[36] and McKay[37] for a related method, recent applications and ways to exploit symmetry.

Definition of the Enumeration Tree. To each polyhex with h hexagons, or son, we must associate a single polyhex with $h - 1$ hexagons, or father. Using the BEC code, a straightforward way to define a unique father for each polyhex would be to choose that one with the lexicographically largest code. Then, once a polyhex is generated from a potential father, its code must be scanned to identify all other potential fathers obtained by removing one hexagon, and the code of each of them must be computed. The polyhex will be a legitimate one if and only if the code of the polyhex from which it has been generated is the lexicographically largest one.

The high efficiency of the proposed algorithm is due to an alternative way to check whether a polyhex must be considered or not as a legitimate one. It is described by the following rule, which induces the enumeration tree.

Rule 1: *A polyhex shall be considered as legitimate if and only if the first digit of its BEC code corresponds to the last added hexagon.*

In other words, *Rule 1* states that the father of a polyhex is the polyhex obtained by removing its first hexagon.

The use of this enumeration tree (Figure 4) reduces significantly the complexity of the program. Instead of scanning the code of the polyhex and finding all its potential fathers, as already described, we just consider one of them. The complexity of the validation is then reduced in worst case from $O(h^3)$, due to the computation and canonization of the code of each potential father, to $O(h^2)$, the validation being included in the canonization of the code; as this operation is the critical one, the global complexity of the algorithm is reduced by a factor of h.

An ideal enumeration scheme would allow identification of the positions where hexagons may be added to generate a legitimate polyhex instead of any later validation. Such a scheme appears to be difficult to obtain. However, *a priori* identification of illegitimate additions is often possible, even before any computation, due to the definition of the enumeration tree (for example: adding an hexagon appearing as a 3 in the code while another hexagon is and remains represented by a 4 or 5 in the same code). When generation has not been shown to be illegitimate, canonization is easy because the starting point of the code of a legitimate polyhex is known before such an operation. Moreover, identification of a better starting point for the code brings this last operation to an end. One then rejects the new polyhex even before its code is totally defined.

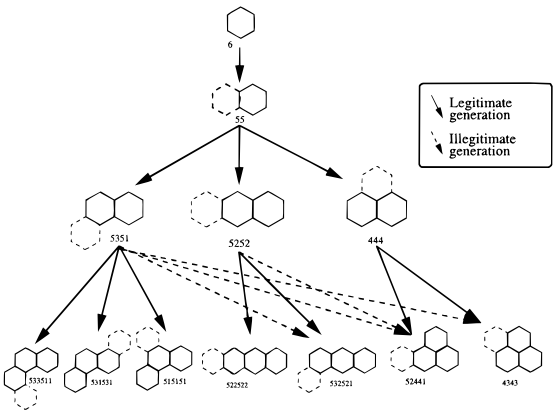


Figure 4. Representation of the enumeration tree.

Validity of the Enumeration Tree to $h = 21$. The enumeration tree is valid if and only if each polyhex has exactly one father. As the BEC code of a polyhex is unique, its first hexagon is also unique; therefore, every polyhex has at most one father. If the first hexagon appears only once in the code, removing it produces a polyhex. However, a problem occurs if this first hexagon appears twice in this code (it cannot appear three times because the code would then have to begin with a 1). Indeed, in that case, removing the first hexagon splits the polyhex in two. Such a polyhex has no father, because a disconnected structure is no longer a polyhex. Having no father, this polyhex cannot be generated by the algorithm and is called an *orphan.* The problem of orphans does not, however, occur in the enumeration of polyhexes with a sufficiently small number of hexagons as shown next.

Theorem 1: *No planar simply connected orphan has <29 hexagons.*

The proof of this theorem, being rather long, is given in the *Appendix*. If polyhexes with >28 hexagons should be enumerated, this could be done (in principle, as the computing time would be very large) with the algorithm proposed here, completed with a specialized algorithm for enumeration of orphans.

Addition of Hexagons. There are three legitimate ways to add hexagons to a polyhex.

1. A hexagon denoted by a digit $x \geq 3$ in the code with at least one free edge may support an addition. The digit "x" of the code is then replaced by "$a5b$" where $a + b + 1 = x$ and $a \geq 1$, $b \geq 1$ (Figure 5a).
2. Two adjacent hexagons represented by the sequence "xy" (where $x > 1$, $y > 1$) in the code may support the addition of an hexagon adjacent to both of them. The sequence "xy" of the code is then replaced by "$(x - 1)4(x - 1)$" (Figure 5b).

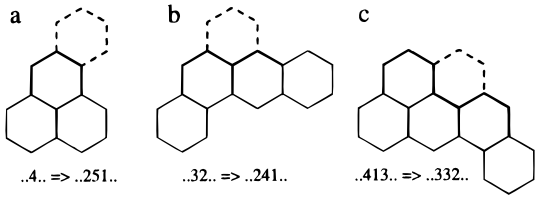


Figure 5. Addition of hexagons.

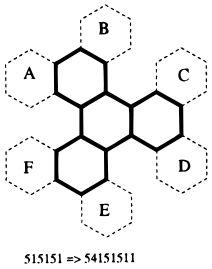


Figure 6. The polyhex 515151 may produce six times the same son.

3. Three consecutive hexagons on the boundary represented by "*x*1*y*" (where $x \geq 2$, $y \geq 2$) may support the addition of an hexagon adjacent to all three of them. The sequence *x*1*y* of the code is then replaced by "$(x-1)3(x-1)$" (Figure 5c).

Proposition 2: *No polyhex obtained by addition of an hexagon sharing more than three consecutive edges with the polyhex is legitimate*

Proof: The added hexagon would then have at most two external edges. According to *Rule 1*, the code would then begin with a digit ≤ 2. From *Proposition 1*, there is no corresponding polyhex.

Method to Avoid Multiple Generation of a Polyhex from the Same Father. In case the father has symmetry, it may produce many times the same polyhex (Figure 6). The number of polyhexes generated by the same polyhex being small (≈ 5 on average), we just keep a list of them and check for duplicates.

If the symmetry classes of polyhexes with $h-1$ hexagons are known when generating those with *h* hexagons, this step can be avoided. There are two cases: (*i*) if the father polyhex has no symmetry (which happens in most cases, see Tables 3−5), the test for multiple generation is skipped; and (*ii*) if the father has symmetry, one may consider one hexagon addition for each orbit. However, there does not seem to be much to be gained by such refinements (experiments with (*i*) led to a reduction in computation time of $\approx 1\%$).

Verification of Planarity. To avoid the generation of helicenes, we note the position of each external hexagon of the current polyhex in the plane and then determine the edges upon which it is possible to add hexagons without generating helicenes. Figure 7 shows the positions that must be controlled depending on the number of free edges of the added hexagon. If addition of an hexagon would form a helicene (or, which is equivalent when viewing the resulting polyhex as planar, which closes a hole of at least *one* hexagon), this addition is forbidden.

The Basic Algorithm. This algorithm enumerates all legitimate polyhexes with *h* hexagons that are generated by a given polyhex *P* with $h-1$ hexagons.

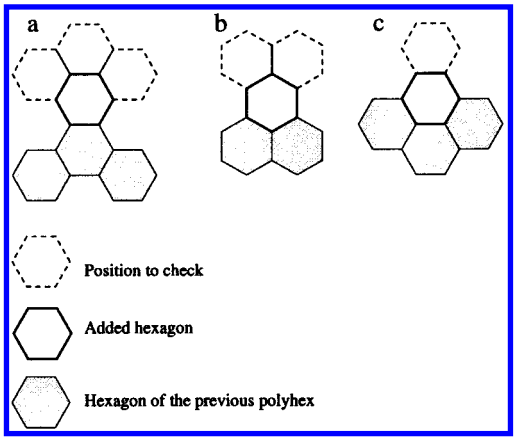


Figure 7. Verification of planarity in case the added hexagon has 3 (a), 2 (b), or 1 (c) free edges.

1. **Addition of hexagons:** Before adding any hexagon, first make sure the planarity condition will be respected. Then
   - Generate all possible polyhexes obtained by addition of a 5 to the code of *P*.
   - If the code of *P* does not begin with a 5, generate all possible polyhexes obtained by addition of a 4; otherwise, consider only the addition of a 4 adjacent to the initial 5.
   - If the code of *P* has no 5 and at most two 4, consider the addition of a 3.
2. **Validation:** Make sure that each generated polyhex can be described by a code beginning with the new hexagon. Discard those that do not.

Justification of the algorithm follows from *Rule 1* and definition of the BEC code, in particular its lexicographic property.

IMPLEMENTATION AND NUMERICAL RESULTS

Implementation. The enumeration is made in three steps.

1. **Initialization:** We first make a database of polyhexes for *h* rather small. The program can of course be initialized with the single polyhex with two hexagons, but this implies that the whole enumeration is done without interruption and any technical problem (such as an electric shutdown) forces the whole work to be done again. As the enumeration program is fast, reading a polyhex from a file takes more time than generating it from a smaller one and forces us to make a compromise between time and risk due to technical problems. For $h = 18$ to 20, we chose to start the enumeration with 67 files containing the 331 polyhexes with seven hexagons, whereas 6018 files containing the 30086 polyhexes with 10 hexagons were used for $h = 21$.
2. **Enumeration:** For each particular file, we generate all the descendants of each polyhex to the depth we need by recursive call of the algorithm previously described. The use of the enumeration tree allows simultaneous enumerations on different computers with no need for shared memory (this was used for $h = 21$). In the case of parallel treatment of the files, a system of lock files is used to avoid multiple computation of the same file as well as systematic backups of any partial result for the final validation.

Table 1. Number of Planar Simply Connected Polyhexes and Time Needed According to h

h	number of polyhexes	computation time
10	30 086	0.46 s
11	141229	2.32 s
12	669 584	11.42 s
13	3 198 256	58 s
14	15 367 577	4 min, 56 s
15	74 207 910	25 min, 33 s
16	359 863 778	2 h, 10 min, 20 s
17	1 751 594 643	11 h, 10 min, 02 s
18[a]	8 553 649 747	2 days, 9 h, 21 min
19[a]	41 892 642 772	12 days, 8 h, 14 min
20[a]	205 714 411 986	63 days, 14 h, 25 min
21[a]	1 012 565 172 403	≈330 days[b]

[a] New value. [b] The time for $h = 21$ is not given precisely because this enumeration has been done on a network of various computers (Sun Sparc 4,5,10,20 and Ultra) that are difficult to compare between themselves or with the Pentium 133 used for the others enumerations.

3. **Further results:** The number of carbon and hydrogen atoms of each polyhex is directly computed from the number of digits in its code, because it only depends on the number of internal vertices and of hexagons. The number of hexagons is equal to the depth of the enumeration tree and the number of internal vertices is then directly obtained from the number of digits of the BEC code.[26,27] The number of carcinogenic bay regions, which are represented by the sequence "515" in the code, is also easy to compute. Symmetry classes are also determined using simple rules described in ref 26. After the complete treatment of a data file, output files containing the number of each kind of polyhex generated are updated before proceeding to the next file.

This program was fast enough to complete successfully the enumeration of the aforementioned characteristics of all planar simply connected polyhexes with up to $h = 20$ hexagons with a PC-Pentium (133 Mhz) in 63 days, 14 h, and 25 min.

Numerical Results. The main new results obtained are the number of planar simply connected polyhexes with $h =$ 18, 19, 20 and 21 hexagons. These results are presented in Table 1. For $h \leq 20$, computations were done with a PC-Pentium. For $h = 21$, a network of various computers was used. Observe that for the case $h = 16$, the largest value for which computing time was reported previous to this paper, this time was reduced from 93 days on a PC 386 (20 Mhz)[22] to 2 h, 10 min, and 20 s on a PC-Pentium (133 Mhz). The number of polyhexes with $h = 21$ (*i.e.*, 1 012 565 172 403) is very large and highlights the efficiency of the enumeration method. However, because such a large set of polyhexes is difficult to use, a random sample has been selected, keeping polyhexes with a probability of 10^{-6} (the random number generator used for this purpose is due to L'Ecuyer[38]).

A formula estimating the number of polyhexes with h hexagons from the number of polyhexes with $h - 1$ hexagons was given by Aboav and Gutman.[39] The predictions are close to the real values for $h \leq 17$, but appear to underestimate the real values when $h > 17$, and the relative difference increases when h grows (Table 2).

Table 2. Exact and Estimated Numbers of Planar Simply Connected Polyhexes

h	exact number	estimated value	error	(error %)
10	30086	30087	1	(0.03)
11	141229	141183	−46	(−0.03)
12	669584	669782	198	(0.03)
13	3198256	3200916	266	(0.01)
14	15367577	15383524	15947	(0.10)
15	74207910	74277568	−69658	(0.09)
16	359863778	360078349	214571	(0.06)
17	1751594643	1751728873	134230	(0.0001)
18	8553649747	8548784328	−4865419	(−0.057)
19	41892642772	41838577888	−54064884	(−0.129)
20	205714411986	204910026644	−804385342	(−0.391)
21	1012565172403	1006213570016	−6351602387	(−0.627)

Table 3. Isomers for $h = 18$ According to Symmetry

isomer	number	C2h	D2h	C3h	D3h	C2v
$C_{52}H_{18}$	3	1	0	0	1	1
$C_{53}H_{19}$	53	0	0	0	0	5
$C_{54}H_{20}$	471	14	2	0	0	23
$C_{55}H_{21}$	2437	0	0	3	0	24
$C_{56}H_{22}$	10587	59	5	0	0	91
$C_{57}H_{23}$	39143	0	0	0	0	71
$C_{58}H_{24}$	133713	206	6	8	0	308
$C_{59}H_{25}$	424429	0	0	0	0	165
$C_{60}H_{26}$	1262442	601	6	0	0	871
$C_{61}H_{27}$	3515696	0	0	15	2	354
$C_{62}H_{28}$	9295801	1602	5	0	0	2252
$C_{63}H_{29}$	23000043	0	0	0	0	715
$C_{64}H_{30}$	53396021	3832	12	39	1	5191
$C_{65}H_{31}$	115992011	0	0	0	0	1139
$C_{66}H_{32}$	234832415	8050	16	0	0	10535
$C_{67}H_{33}$	435901284	0	0	99	2	1695
$C_{68}H_{34}$	738076219	14601	17	0	0	18721
$C_{69}H_{35}$	1123527420	0	0	0	0	1971
$C_{70}H_{36}$	1503478059	22472	47	134	0	28086
$C_{71}H_{37}$	1676318405	0	0	0	0	1474
$C_{72}H_{38}$	1460056378	24071	13	0	0	29539
$C_{73}H_{39}$	878110881	0	0	125	0	0
$C_{74}H_{40}$	296275836	14756	27	0	0	17445
total	8553649747	90265	156	423	6	120676

Tošić *et al.*[23] conjectured that the number of polyhexes with $h = 18$ hexagons would be in the range (8549 ± 4)-10^7, which is almost correct.

For each nonisomorphic isomer, the number of polyhexes for each class of symmetry was computed as well as the number of carcinogenic bay regions for $h \leq 20$. These numbers were known for $h \leq 14$ since 1992;[6] then they were obtained for $h = 15$, 16, and 17 by Tošić *et al.* in 1995.[23] All values have been confirmed by our program. The values for $h = 18$, 19, and 20 (except for carcinogenic bay regions, as the data is too voluminous; it may be obtained upon request) are now presented, for the first time, in Tables 3−5 (only columns corresponding to classes of symmetry for which there exists at least one polyhex with h hexagons are given). Observe that all isomers of $C_{81}H_{43}$ have no symmetry at all. This phenomenon can be explained as follows: From the relation $H = 4h + 4 - n_i$ from Gutman and Cyvin,[8] where H denotes the number of hydrogens, their number n_i of interior vertices is equal to 1. So, these polyhexes consist of three mutually adjacent hexagons to which are appended at most three disjoint catacondensed polyhexes. The common vertex of these three hexagons must belong to any axis of symmetry (as otherwise it would not be the unique interior vertex). But, having a symmetry axis and an even number of hexagons implies that at least one additional hexagon must

Table 4. Isomers for $h = 19$ According to Symmetry

isomer	number	C2H	D2H	C3H	D3H	C6H	D6H	C2v
$C_{54}H_{18}$	1	0	0	0	0	0	1	0
$C_{55}H_{19}$	18	0	0	0	0	0	0	3
$C_{56}H_{20}$	256	3	1	0	0	0	0	17
$C_{57}H_{21}$	1647	0	0	1	0	0	0	14
$C_{58}H_{22}$	7885	17	2	0	0	0	0	92
$C_{59}H_{23}$	32042	0	0	0	0	0	0	60
$C_{60}H_{24}$	116648	66	6	4	1	0	0	316
$C_{61}H_{25}$	388180	0	0	0	0	0	0	175
$C_{62}H_{26}$	1223950	189	7	0	0	0	0	814
$C_{63}H_{27}$	3622656	0	0	6	2	0	0	495
$C_{64}H_{28}$	10119046	589	10	0	0	0	0	2323
$C_{65}H_{29}$	26745152	0	0	0	0	0	0	1402
$C_{66}H_{30}$	67072867	1677	18	26	3	1	0	6037
$C_{67}H_{31}$	157998026	0	0	0	0	0	0	3113
$C_{68}H_{32}$	350234105	3829	17	0	0	0	0	12851
$C_{69}H_{33}$	727273192	0	0	41	0	0	0	6200
$C_{70}H_{34}$	1405607063	7948	24	0	0	0	0	24710
$C_{71}H_{35}$	2494602124	0	0	0	0	0	0	11851
$C_{72}H_{36}$	4038939706	15649	23	93	8	1	1	46152
$C_{73}H_{37}$	5870405899	0	0	0	0	0	0	18123
$C_{74}H_{38}$	7465388988	25543	47	0	0	0	0	72151
$C_{75}H_{39}$	7889345133	0	0	169	0	0	0	18007
$C_{76}H_{40}$	6500036300	26931	0	0	0	0	0	74547
$C_{77}H_{41}$	3704740320	0	0	0	0	0	0	8970
$C_{78}H_{42}$	1178741568	15472	34	203	5	0	0	40141
total	41892642772	97913	189	543	19	2	2	348564

Table 5. Isomers for $h = 20$ According to Symmetry

isomer	number	C2H	D2H	C2v
$C_{57}H_{19}$	4	0	0	1
$C_{58}H_{20}$	129	8	1	15
$C_{59}H_{21}$	1009	0	0	17
$C_{60}H_{22}$	5726	42	5	66
$C_{61}H_{23}$	25050	0	0	68
$C_{62}H_{24}$	97607	178	5	273
$C_{63}H_{25}$	345834	0	0	181
$C_{64}H_{26}$	1140529	576	5	845
$C_{65}H_{27}$	3540792	0	0	439
$C_{66}H_{28}$	10464798	1650	10	2360
$C_{67}H_{29}$	29228956	0	0	1013
$C_{68}H_{30}$	77591586	4498	8	6134
$C_{69}H_{31}$	195628098	0	0	2016
$C_{70}H_{32}$	468024447	10935	22	14544
$C_{71}H_{33}$	1055012528	0	0	3709
$C_{72}H_{34}$	2241109396	23872	31	30661
$C_{73}H_{35}$	4460933006	0	0	5663
$C_{74}H_{36}$	8257827535	46560	42	57964
$C_{75}H_{37}$	14051002527	0	0	7914
$C_{76}H_{38}$	21801082567	78191	40	95530
$C_{77}H_{39}$	30306079909	0	0	8333
$C_{78}H_{40}$	36713205973	109470	90	130675
$C_{79}H_{41}$	36879305655	0	0	5442
$C_{80}H_{42}$	28838451119	105790	14	123735
$C_{81}H_{43}$	15621799283	0	0	0
$C_{82}H_{44}$	4702507923	58460	37	65905
total	205714411986	440230	310	563503

be split in two by this axis. It cannot be adjacent to two of the initial hexagons, as this would imply $n_i \geq 2$, nor elsewhere on this axis, as then either $n_i \geq 2$ or the polyhex would be a coronoid. As the remaining number of hexagons (*i.e.*, 17) is not divisible by 2 or 3, there cannot be a rotational symmetry.

DISCUSSION

This paper reports an efficient enumeration of polyhexes with up to $h = 21$ hexagons. For this last number, there are over one trillion molecules, that is, >600 times more than in the largest sets previously enumerated. There are two reasons for which such a large enumeration could be achieved. First, recourse to the reverse search method of Avis and Fukuda[25] avoids listing all polyhexes with $h - 1$ hexagons before generating those with h hexagons and checking in that list for duplicates. Such an operation, both time and space consuming, was required by some former algorithms. Second, defining the son−father relationship between polyhexes by postulating that the father must be obtained by removal of the first hexagon of the current polyhex (or, which is equivalent, that the first digit of the BEC code of this polyhex correspond to the last added hexagon) makes checking that a polyhex is legitimate (or a new one) much faster than other options; as, for example, taking as father that one with the lexicographically largest code. Clearly, both of these innovations could be used *mutatis mutandis* in many other chemical enumeration problems (an example now under study is the enumeration of helicenes). This may well be more important than the additional information on numbers of polyhexes, and their breaking down by symmetry classes already given.

As usual, one may speculate on the possibility of enumerating still larger polyhex sets. The ever increasing power of computers makes this likely to happen in the near future (in fact, running in background mode our network of computers for ≈10 months instead of 6 weeks would solve the case $h = 22$). In view of the current rapid increase in computing power per dollar, further progress is to be expected. But, what is more interesting is to try to find out where progress can be made in the algorithms. The best theoretical complexity for enumeration is $O(h)$ per polyhex, if the code for each one must be written (this complexity might be lower if updating takes place and polyhexes are only counted, not enumerated). The present algorithm has a worst case complexity in $O(h^3)$ per polyhex, which is much higher than $O(h)$. However, the empirical average complexity is much lower. Comparing computation times for $h =$

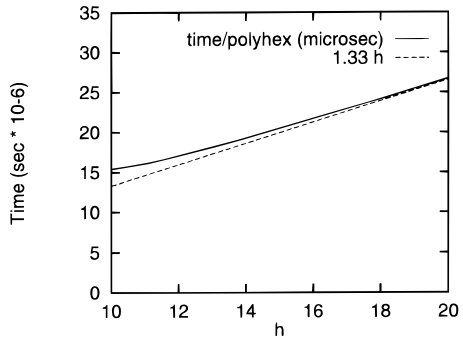


Figure 8. CPU time as a function of *h*.

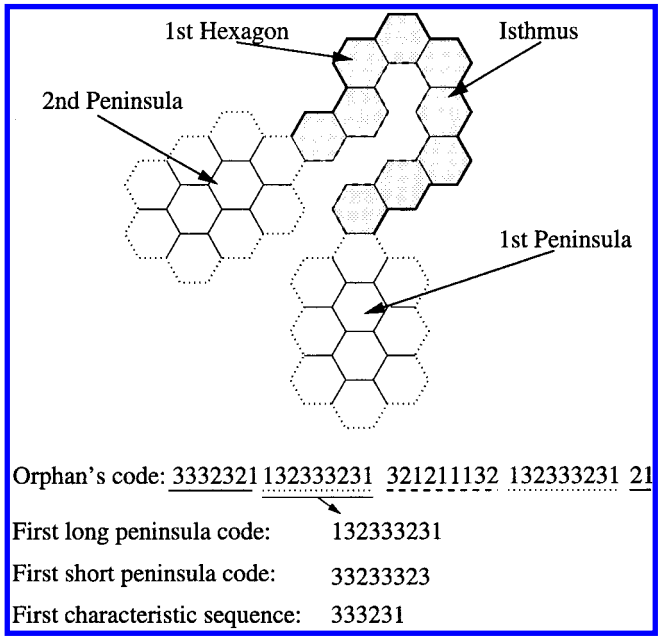


Figure 9. Definitions relating to orphans.

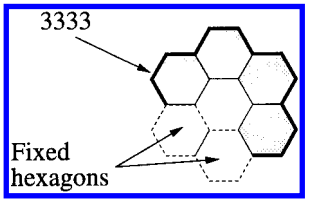


Figure 10. No orphan has more than three successive 3s.

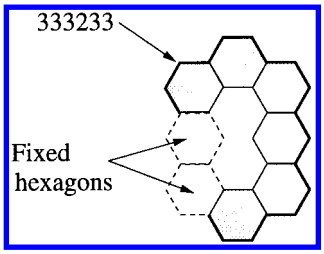


Figure 11. An orphan's code cannot contain "333233".

10 to $h = 20$ shows they are a bit better than $O(h)$ per polyhex. This is not a contradiction with the theoretical $O(h)$ because an asymptotic value of $\approx 1.33\ 10^{-6}\ h$ seconds per polyhex, which *is* $O(h)$, appears to hold (Figure 8). So, lowering the worst case complexity of the present algorithm might not help much, because most polyhexes seem to be enumerated rapidly. This also indicates that the average computing time for enumeration appears to be close to optimal, up to a constant factor. Possibly, much better results could be obtained for counting only. It thus appears that the conclusion we arrive at is similar to that of Tošić *et al.*[23]; that is, although some progress has been made, new ideas are needed and may be found sooner or later, which will further enhance polyhex counting and enumeration and justify its continued interest more than breaking records with faster computers.

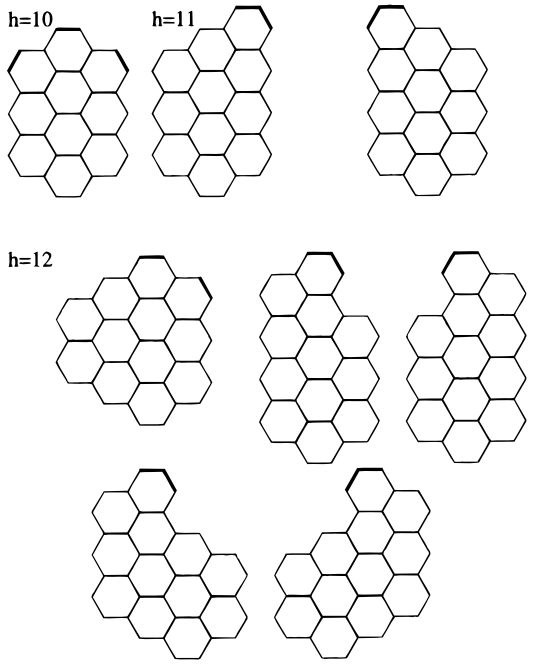


Figure 12. Peninsulas with 10 to 12 hexagon.

Table 6. Peninsulas with 10 to 12 Hexagons and Their Characteristic Sequences

h	peninsula code	characteristic sequence	Lemma 4 used
10	33323332	3332331	X
	33233323	333231	
	32333233	3332331	X
11	423323331	3332332	X
	423233322	333232	
	422333232	333232	
	413332332	3332332	X
12	4323233321	333232	
	4322333231	333231	
	4313332331	333233	X
	4133233313	333233	X
	4132333223	333231	
	4123332323	333232	
	332332332	332332	
	323323323	332332	

APPENDIX

In this appendix we determine the size of the smallest planar orphan, and, as a consequence show that no special procedure for orphans must be added to the basic algorithm for $h \leq 28$.

We first give a few more definitions about polyhexes. The n^{th} **hexagon** is the hexagon represented by the n^{th} digit in the code of the polyhex. The **isthmus** of an orphan is the maximal set of consecutive hexagons, including the first hexagon, which each appear twice in the code and are adjacent to exactly two hexagons which also appear twice in the code. A **peninsula** is one of the two polyhexes obtained by removing the isthmus of an orphan. The **first peninsula** appears first in the orphan's code and the **second**

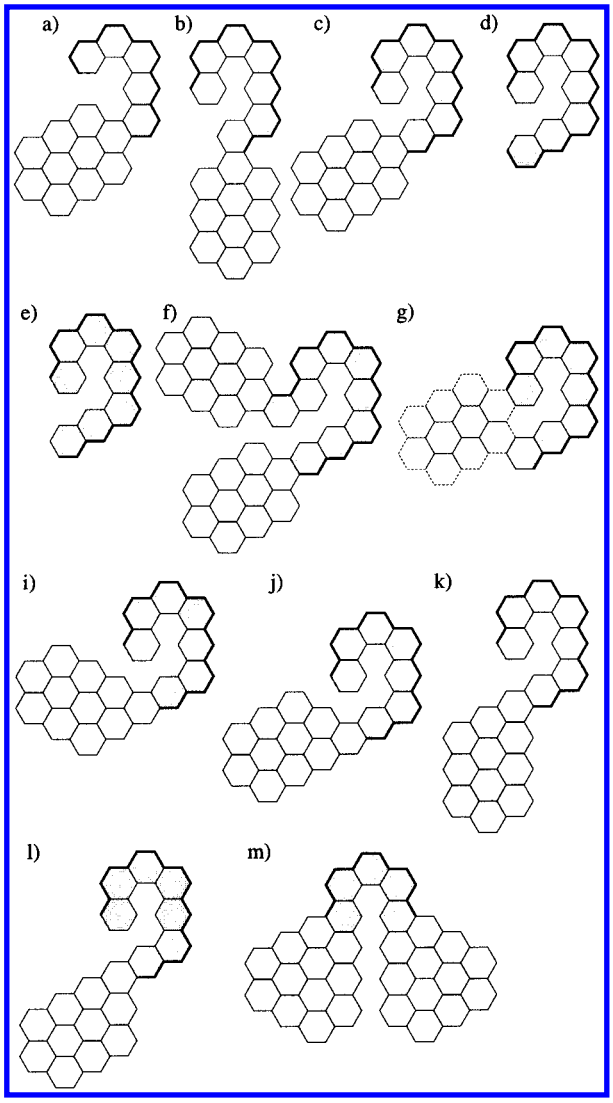


Figure 13. Configurations considered in the proof.

peninsula appears second. As we will use partial codes, the way they are defined is important; by convention, all codes used here are computed clockwise. The **isthmus code** is the string of digits representing the number of boundary edges of consecutive hexagons in the isthmus from the second peninsula to the first (the beginning of the code of the orphan belongs to this string). The isthmus has two fixed neighbors (one from each peninsula). The **long peninsula code** of a given peninsula is the string of digits representing hexagons belonging to that peninsula in the orphan's code, in the same order. The **short peninsula code** is the string of digits describing the peninsula assuming the isthmus is removed, beginning with the hexagon adjacent to the isthmus (Note that neither the short nor the long peninsula code is lexicographically ordered). The **characteristic sequence of a peninsula** is the lexicographically largest sequence that belongs to the long peninsula code. These concepts are illustrated on Figure 9.

The proof uses the following four Lemmas.

Lemma 1: *An orphan's code begins by a* 3.

Proof: The first hexagon of an orphan must appear twice in the code; thus it cannot be a 4 or a 5; from *Proposition 1,* it must be a 3.

Lemma 2: *An orphan's code contains no 4 or 5.*

Proof: This follows directly from *Lemma 1* and the lexicographical rule.

Lemma 3: *An orphan's code cannot have more than 3 successive 3's.*

Proof As, from *Lemma 2,* an orphan's code does not have any digit >3 and it must be lexicographically maximum, the largest sequence of successive 3s must be at the beginning of the code. Four successive 3s then imply that the polyhex has a hole (*i.e.*, that it is a coronoid) or that it is not planar (Figure 10).

Lemma 4: *An orphan's code cannot contain the sequence 333233.*

Proof Assume by contradiction the orphan's code contains the sequence 333233. Then, from *Lemmas* 1 to 3, the beginning of this code must be 333233, but this implies the polyhex has a hole or is not planar (Figure 11).

We next proove **Theorem 1,** that is, *No planar simply connected orphan has <29 hexagons*

Proof: The proof is by enumeration of small peninsulas and of their positions that respect the properties of the orphans given in Lemmas 1−4.

The first step is to enumerate all small peninsulas (considering those with $h \leq 12$ will suffice). These are all polyhexes that have at most three successive 3s, one 4, and no 5 in their code. Indeed, a 5 cannot be the first hexagon of a peninsula because it shares an edge with a hexagon of that peninsula that appears twice in the code; the 5 should then belong to the isthmus, which is impossible. The peninsulas up to 12 hexagons obtained by a simple but tedious enumeration are shown in Figure 12. The edges that can be shared with the isthmus are represented in bold. Note that mirror symmetric peninsulas are considered as different. This is necessary as replacing a peninsula by its mirror image might lead to a nonplanar polyhex.

Table 6 lists the short peninsula code of each of those peninsulas for each edge shared with the isthmus, as well as the corresponding characteristic sequence. An *X* in the last column indicates that the peninsula must not be considered according to Lemma 4.

Let h_i' denote the number of hexagons in the peninsula i ($i = 1, 2$). We first note that if the characteristic sequence of the first peninsula begins with 33323, a hexagon must be added before the second peninsula for planarity reasons (Figure 13a).

1. $h_1' = 10$. Any orphan containing the peninsula with 10 hexagons must begin with the sequence 33323. The isthmus code must contain 1333231 or 1333232, but this first configuration alone is not valid for lexicographical reasons (Figure 13b) and the other one alone does not respect planarity (Figure 13c).

   The next step is to add an hexagon before the first peninsula. There are two cases: 1333231 and 1333232.

   In the first case, there are three ways to place the new hexagon:

(a) 13332313: for planarity reasons, a hexagon must be added before the second peninsula making the isthmus code 113332313 or 213332313 but both do not make a planar polyhex unless another hexagon is added. There would then be at least 30 hexagons (10 + 10 + 10). We do not explore this case because a smaller orphan will be found later.

(b) 13332312 or 13332311: the lexicographical rule is not respected.

In the second case, the three ways to add the new hexagon are:

(a) 13332323: produces a nonplanar polyhex (Figure 13d)

(b) 13332322: forces us to add an hexagon before the second peninsula to respect planarity (Figure 13e). Then the isthmus becomes 113332322, and we obtain an orphan with 29 hexagons (Figure 13f).

(c) 13332321: requires the addition of an hexagon before the second isthmus to respect planarity (Figure 13g). Hence, the isthmus has nine hexagons and cannot produce an orphan with less than 29 hexagons.

2. $h_1' = 11$. The characteristic sequence of each of the two peninsulas for $h_1' = 11$ is 333232 and the isthmus must have at least seven hexagons. There are four ways to connect the first peninsula to the isthmus (represented in Figures 13i−l). The first two produce a nonplanar polyhex and the last two are not lexicographically largest. In both cases, a hexagon must be added to the isthmus. The smallest orphan thus obtained has 10 + 8 + 11 = 29 hexagons or more. We shall not explore this configuration further because it cannot lead to smaller orphans than previously obtained.

3. $h_1' = 12$. For the same reasons as for $h_1' = 11$, the peninsulas whose characteristic sequence begins with 33323 are not considered because they cannot be used to generate smaller orphans than already obtained. The only peninsula that might produce a smaller orphan is 332332332, but both peninsulas must have at least 12 hexagons not to have a characteristic sequence beginning with 33323. Then the orphan's code must begin with 332332 or 333. In the first case, the orphan has at least 30 hexagons, and in the second, two hexagons must be added for planarity and the orphan has (at least) 29 hexagons (Figure 13m).

4. $h_1' = 13$. Peninsulas with 13 hexagons have characteristic sequences beginning with 332 or 333, which implies an isthmus with at least three hexagons. If the second peninsula has only 12 hexagons, a hexagon must be added to the isthmus before it; whether we use it or a 13 hexagons peninsula, the orphan made has at least 29 hexagons (12 + 4 + 13 or 13 + 3 + 13). Using a smaller second peninsula need not be considered, as explained before.

5. $h_1' \geq 14$. For any 14 or more hexagons peninsula, either the characteristic sequence is at least 332, then the isthmus must have at least three hexagons and the smallest orphan has 12 + 3 + 14 = 29 hexagons (the characteristic sequence 333 needs at least five hexagons for the isthmus, the orphan then has 10 + 5 + 14 = 29 hexagons or more), or both peninsulas must have 14 hexagons or more, in which case the orphan cannot have less than 14 + 1 + 14 = 29 hexagons. No orphans with <29 hexagons can be obtained in that way.

ACKNOWLEDGMENT

Work supported by NSERC Grant #*GP*0105574, FCAR Grant #95*ER*1048 and a grant from CETAJ. The authors are greatful to A. T. Balaban and G. Brinkmann for helpful comments.

REFERENCES AND NOTES

(1) Balaban, A. T. Enumeration of isomers. In *Chemical Graph Theory;* 177−234. Abacus Press, Gordon and Breach: New York, 1990; pp 177−234.

(2) Balaban, A. T.; Brunvoll, J.; Ciolowski, J.; Cyvin, B. N.; Cyvin, S. J.; Gutman, I.; He, W. J.; Knop, J. V.; Kovačević, M.; Müller, W. R.; Szymanski, K.; Tošić, R.; Trinajstić, N. Enumeration of benzenoid and coronoid hydrocarbons. Z. *Naturforsch.* **1987**, *42a*, 863−870.

(3) Balasubramanian, K. Recent chemical applications of computational combinatorics and graph theory. In *Computational Chemical Graph Theory;* Rouvray, D. H., Ed.; Nova Science: New York, 1990.

(4) Bonchev, D.; Rouvray, D. H. *Chemical Graph Theory, Introduction and Fundamentals;* Abacus Press, Gordon and Breach: New York, 1991.

(5) Bonchev, D.; Rouvray, D. H. *Chemical Graph Theory, Reactivity and Kinetics;* Abacus Press, Gordon and Breach: New York, 1992.

(6) Brunvoll, J.; Cyvin, B. N.; Cyvin, S. J. Enumeration of benzenoid systems and other polyhexes. *Top. Curr. Chem.* **1992**, *162*, 65−180.

(7) Dias, J. R. Benzenoid hydrocarbons. In *Handbook of Polycyclic Hydrocarbons;* Elsevier: Amsterdam, 1987.

(8) Gutman, I.; Cyvin, S. J. *Introduction to the Theory of Benzenoid Hydrocarbons;* Springer-Verlag: 1989; p 23.

(9) Hosoya, H. Some recent advances in counting polynomials in chemical graph theory. In *Computational Chemical Graph Theory;* Rouvray, D. H., Ed.; Nova Science: New York, 1990.

(10) Knop, J. V.; Müller, W. R.; Szymanski, K. Computer-oriented molecular codes. In *Computational Chemical Graph Theory;* Rouvray, D. H. Ed.; Nova Science: New York, 1990.

(11) Mercier, C.; Sobel, Y.; Dubois, J. E. Darc/pelco method: A topological tool for qsar search and its reliable predictive capability. In *Chemical Graph Theory*, Abacus Press, Gordon and Breach: New York, 1990; pp 199−258.

(12) Trinajstić, N.; Nikolić, S.; Knop, J. V.; Müller, W. R.; Szymanski, K. *Computational Chemical Graph Theory;* Ellis Horwood: Chichester, U.K., 1991.

(13) Trinajstić, N. *Chemical Graph Theory;* CRC: Boca Raton, FL, 1992.

(14) Balasubramanian, K.; Kaufman, J. J.; Koski, W. S.; Balaban, A. T. Graph theoretical characterization computer generation of certain carcinogenic benzenoid hydrocarbons and identification of bay regions. *J. Comput. Chem.* **1980**, *1*, 149−157.

(15) Knop, J. V.; Szymanski, K.; Jericević, Z.; Trinajstić, N. Computer enumeration and generation of benzenoid hydrocarbons and identification of bay regions. *J. Comput. Chem.* **1983**, *4*, 23−32.

(16) Stojmenović, I.; Tošić, R.; Doroslovacki, R. Generating and Counting Hexagonal Systems. In *Graph Theory, Proceedings of the Sixth Yougoslav Seminar on Graph Theory*, 1985.

(17) He, W. J.; He, W. C.; Wang, Q. X.; Brunvoll, J.; Cyvin, S. J. Supplement to Enumeration of Benzenoid and Coronoid Hydrocarbons. Z. *Naturforsch.* **1988**, *43a*, 693−694.

(18) Nikolić, S., Trinajstić, N.; Knop, J. V.; Müller, W. R.; Szymanski, K. On the concept of the weighted spanning tree of dualist. *J. Math. Chem.* **1990**, *4*, 357−375.

(19) Balaban, A. T.; Harary, F. Enumeration and proposed nomenclature of benzenoid cata-condensed polycyclic aromatic hydrocarbons. *Tetrahedron* **1967**, *24*, 2505−2516.

(20) Müller, W. R.; Szymanski, K.; Knop, J. V.; On counting polyhex hydrocarbons. *Croat. Chem. Acta* **1989**, *62*, 481−483.
(21) Müller, W. R.; Szymanski, K.; Knop, J. V.; Nikolić, S.; Trinajstić, N. On the enumeration and generation of polyhex hydrocarbons. *J. Comput. Chem.* **1990**, *11*, 223−235.
(22) Knop, J. V.; Müller, W. R.; Szymanski, K.; Trinajstić, N. Use of small computers for large computations: Enumeration of polyhex hydrocarbons. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 159−160.
(23) Tošić, R.; Mašulović, D.; Stojmenović, I.; Brunvoll, J.; Cyvin, S. J.; Cyvin, B. J. Enumeration of Polyhex Hydrocarbons to $h = 17$. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 181−187.
(24) Henson, R. A.; Windlinx, K. J.; Wiswesser, W. J. Lowest order computer-oriented "ring-index" diagrams-verifying correct orientation of fused hexagonal ring systems. *Comput. Biomed. Res.* **1995**, *8*, 53−71.
(25) Avis, D.; Fukuda, K. Reverse search for enumeration. *Discrete Appl. Math.* **1996**, *65*, 21−46.
(26) Hansen, P.; Lebatteux, C.; Zheng, M. The boundary-edges code for polyhexes. *Theochem. J. Mol. Structures* **1996**, *363*, 237−247.
(27) Herndon, W.; Bruce, A. J. Perimeter code for benzenoid aromatic hydrocarbons. *Stud. Phys. Theor. Chem.* **1987**, *51*, 491−513.
(28) Herndon, W. C.; Nowak, P. C.; Dallas, A.; Connor, A.; Lin, P. Empirical model calculation for thermodynamic and structural properties of condensed polycyclic aromatic hydrocarbons. *J. Am. Chem. Soc.* **1992**, *114*, 41−47.
(29) Aho, A. V.; Hopcroft, J. E.; Ullman, J. D. *The Design and Analysis of Computer Algorithms;* Addison-Wesley: Reading, MA, 1974.
(30) Dyer, M. E. The complexity of vertex enumeration methods. *Math. Operations Res.* **1983**, *8*, 381−402.
(31) Chen, P.-C.; Hansen, P.; Jaumard, B. Online and offline vertex enumeration by adjacency lists. *Operations Res. Lett.* **1991**, *10*, 403−409.
(32) Bland, R. G. New finite pivoting rules for the simplex method. *Math. Operations Res.* **1977**, *2*, 103−107.
(33) Read, R. C. Every one a winner; *Ann. Discrete Math.* **1978**, *2*, 107−120.
(34) Faradzev, I. A. Generation of nonisomorphic graphs with given degree sequence (russian). In *Algorithmic Studies in Combinatorics;* Nauka: Moscow, 1978; pp 11−19.
(35) Faradzev, I. A. Constructive enumeration of combinatorial objects. In *Problemes Combinatoires et theorie des Graphes Colloque Internat. CNRS 260*; 1978; pp 131−135.
(36) Brinkmann, G. Fast generation of cubic graphs. *J. Graph Theory* **1996**, *23*, 139−149.
(37) McKay, B. D. Isomorph-free exhaustive generation, submitted to *J. Algorithms.*
(38) L'Ecuyer, P. Combined multiple recursive generators. *Operations Res.* **1996**, *44*, 816−822.
(39) Aboav, D.; Gutman, I. Estimation of the number of benzenoid hydrocarbons. *Chem. Phys. Lett.* **1988**, *148(1)*, 90−92.

CI970116N