Oriented 2-Cell Embeddings of a Graph and Their Symmetry Classification: Generating Algorithms and Case Study of the Möbius-Kantor Graph

E. Lijnen* and A. Ceulemans

Departement Chemie, K.U. Leuven, Celestijnenlaan 200F, B-3001 Leuven, Belgium

Received April 21, 2004

We discuss a method to derive all symmetry-distinct oriented 2-cell embeddings of a given graph and classify them based on their symmetry. As an example, we apply the algorithm to the highly symmetrical trivalent Möbius-Kantor graph. Considering the derived 2-cell embeddings as carbon networks leads to some interesting negative curvature carbon allotropes.

1. INTRODUCTION

The study of highly symmetrical graphs offers interesting perspectives for the design of new molecular frameworks.[1,2] The molecular realization of a graph requires that its vertices are positioned in three-dimensional space. In this process topological aspects of graph theory become important.[3] We are especially interested in molecular realizations where the graph is mapped on a closed surface, such as a sphere or a torus. This process is called an embedding. In previous works we have described in detail molecular frameworks that result from particular highly symmetrical embeddings of the regular Klein and Dyck graphs, with 56 and 32 vertices, respectively.[4,5]

In the present paper we adopt a more general point of view and examine the set of all possible embeddings that a given graph may have. This set may be very large and show an interesting and rich structure. In the chemical literature it is often not realized that combinatorial techniques have been described which can retrieve all possible embeddings of a graph. Here we present an algorithm that makes use of group theory to generate all distinct embeddings of a graph. The strength of this algorithm is that it greatly reduces the number of embeddings by removing symmetry redundancies. The procedure is illustrated for the Möbius-Kantor graph with 16 vertices.[7] In principle this graph has 2^{16} embeddings, but symmetry factorization reduces this number to only 759 nonequivalent embeddings.

2. SURFACES AND EMBEDDINGS

Before describing an algorithm to find all oriented 2-cell embeddings of a graph, we first recall some terminology. It is well known that not every graph can be drawn in the plane in such a way that no edges intersect. Nonplanar graphs can avoid all edge intersections when they are *embedded* on an appropriate two-dimensional closed surface.[3] This family of surfaces is divided into two infinite classes,[8] namely the *orientable* and *nonorientable* surfaces. The former class is obtained by attaching n handles to a sphere to form the orientable surfaces S_n: S_0 being the sphere, S_1 the torus, S_2 the double torus, etc. The latter class is formed by taking the sphere and adding m crosscaps ($m \geq 1$) to form the nonorientable surfaces N_m: N_1 being the projective plane, N_2 the Klein-bottle, etc. When a graph is embedded, it divides the underlying surface into a set of 2-dimensional regions[9] called faces or 2-cells, just like the embedding of the cube graph on the sphere divides the surface into 6 quadrangular faces. On orientable surfaces, every face can be given an orientation (a clockwise or counterclockwise sense) in such a way that if one follows the face-orientations, every edge is traversed twice and in opposite directions. For the nonorientable surfaces however, such a consistent orientation of the faces does not exist. From a chemical point of view, the latter class of surfaces is less interesting as they cannot be geometrically realized in ordinary 3D-space. We therefore limit ourselves to the derivation of the embeddings of a graph on orientable surfaces. For these oriented embeddings the topology of the underlying surface and the number of faces are related by the famous Euler equation[10] which states

$$V - E + F = 2 - 2g \tag{1}$$

where V, E, and F are integers denoting the number of vertices, edges, and faces of the embedding. g is the genus of the embedding (= number of handles) and completely determines the topology of the underlying orientable surface.

3. CALCULATING ALL SYMMETRY-DISTINCT ORIENTED 2-CELL EMBEDDINGS

The oriented 2-cell embeddings of a graph are obtained in a combinatorial way. Redundancies of symmetry equivalent embeddings are removed subsequently.

3.1. Generating All Embeddings. The primary part of our algorithm will be devoted to the generation of all possible oriented 2-cell embeddings. For simplicity, we often talk about embeddings in the remainder of the text, but the reader should keep in mind its stricter meaning.

The generation of all embeddings is performed by describing the embeddings in a combinatorial way, known as the Heffter-Edmonds-Ringel rotation principle[11,12] which is thoroughly discussed in Mohar's book.[13] In this method, one starts by giving a consistent sense or handedness of rotation to every point on the orientable surface. An orientable surface

* Corresponding author phone: +32 16 32 73 80; fax: +32 16 32 79 92; e-mail: erwin.lijnen@chem.kuleuven.ac.be.

10.1021/ci049865c CCC: $27.50
Published on Web 08/18/2004

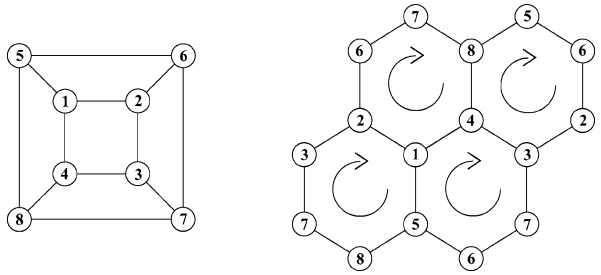

Figure 1. (Left) the cube graph (right) embedding of the cube on the torus.

Table 1. Rotation Scheme of the Toroidal Embedding of the Cube Depicted in Figure 1

1.	2	5	4		5.	1	8	6
2.	1	6	3		6.	2	7	5
3.	2	4	7		7.	3	6	8
4.	1	3	8		8.	4	5	7

provided with such a sense will be called an *oriented surface* and mappings on them are called *oriented embeddings*. Looking at the neighborhood of a vertex v on an oriented surface, the chosen sense determines an ordered sequence of edges around v, which is called a *local rotation* at v. This sequence is defined up to cyclic permutations. As a result, the number of possible local rotations at v, $n_r(v)$, is given by

$$n_r(v) = (deg(v) - 1)! \tag{2}$$

where $deg(v)$ denotes the degree or valency of vertex v. As an example in Figure 1 we show the cube graph and its embedding on a torus. This toroidal embedding corresponds to the set of local rotations given in Table 1 (we have chosen an anticlockwise rotation). Notice that instead of listing the consecutive edges, it is sufficient to list their end-vertices. The set of local rotations, one for each vertex, is denoted as a *rotation scheme*. It is obvious that every oriented embedding can be described by a rotation scheme. However, the reverse is also true, namely, every rotation scheme corresponds to an oriented embedding. The problem of finding all embeddings can thus be reformulated as the problem of finding all rotation schemes. As we shall see, this is straightforward. As a rotation scheme is completely defined by the local rotations at all of its vertices, the number of different rotation schemes (= the number of oriented embeddings) equals:

$$N_r = \prod_{v \in V} n_r(v) \tag{3}$$

Notice that a rotation scheme and its inverse (a rotation scheme where all local rotations are reversed) lead to oriented embeddings which are only distinguished by a reversal of the orientation of their faces. The rotation schemes are therefore in 1−1 correspondence with embeddings of the graph on oriented surfaces and in 2−1 correspondence with embeddings onto orientable surfaces. To retrieve all rotation schemes, we start from a randomly chosen rotation scheme and form all other schemes by changing the local rotation at one or more vertices. This generating process will be the driving force in the retrieval of all embeddings. Every time a new rotation scheme is constructed, our task will be to identify the corresponding oriented embedding. As this embedding is completely characterized by its set of oriented faces, we will describe how to extract these faces starting from a rotation scheme. For this purpose, every edge is seen as a set of two arcs (= directed edges) with opposite directions, hence the edge linking vertices u and v is split into the arcs (uv) and (vu). To trace the first face, one starts by walking along an arbitrary arc, say (uv). When the end-vertex v of that arc is reached, one continues the walk along a new arc starting at this vertex. This arc cannot be chosen freely but is uniquely determined as one always has to choose the arc following the current arc in the local rotation at v. Walking along this newly found arc (vw), one ends up in a new end-vertex w. Again one applies the previous strategy and finds the next arc. This procedure is repeated until one encounters the starting vertex u again, and the next arc to be chosen is the arc (uv) one originally started from. This last condition is necessary because one can end up at the starting vertex u, although the next arc to be chosen is different from the initial one. If this is the case, one obtains a face in which vertex u appears multiple times, which cannot be ruled out. Once a face has been found, the next face can be traced starting from an arc which does not yet belong to one of the previously found faces. The procedure is repeated until all arcs have been traversed once and therefore all faces have been found. The orientation of these faces is determined by the direction in which they were traversed during their generation. We now illustrate this face-tracing procedure starting from the rotation scheme of Table 1. Let us take as a starting point the arc (1−2). When we reach vertex 2, we continue with the arc following arc (1−2) in the local rotation at vertex 2. In Table 1 we see that in the local rotation at vertex 2, vertex 1 which corresponds to the arc (1−2), is followed by vertex 6, corresponding to arc (2−6), so the next arc in our walk will be (2−6). Traveling along this arc we end up at vertex 6. The next vertex of the face can now easily be found by looking for the vertex following 2 in the local rotation at 6, which is vertex 7. Continuing this procedure until the starting arc is encountered, leads to the face (1−2−6−7−8−4). The remaining three faces are found in a similar way, every time starting from arcs which were not used previously. Once all faces have been found, our embedding is completely characterized. The topology or genus of the underlying surface can easily be deduced using the Euler eq 1. With $V = 8$, $E = 12$, and $F = 4$ one finds the genus, $g = 1$, of a torus. A geometrical realization of the embedding can be obtained by appropriately gluing all faces together. In our case, fitting the four hexagonal faces together leads to the toroidal realization shown in Figure 1.

3.2. Isomorphism Checking. Although viewed as labeled structures all derived embeddings are distinct, some of them can be isomorphic when viewed as unlabeled structures. The number of such equivalent embeddings is related to their symmetry. We now describe an algorithm that calculates the symmetries of the embeddings and at the same time can be used to remove unwanted duplicates by checking for isomorphisms.

An automorphism of a graph is defined as a permutation of its vertices preserving all incidence relations. In other words, the vertex permutation also has to map edges onto edges. The permutations that satisfy this condition form a mathematical group, called the automorphism group or

symmetry group of the graph. When a graph is embedded on a surface, a permutation needs to satisfy a supplementary condition in order to be called an automorphism, namely, it also has to map faces onto faces. The set of permutations satisfying these conditions again forms a group which we shall call the automorphism group or symmetry group of the embedding and is necessarily a subgroup of the symmetry group of the nonembedded graph.

A polynomial algorithm for finding all automorphisms of a graph is not known, although the problem has also not been shown to be NP-complete. There are however a number of programs which are quite efficient in performing this task.[14] Finding the automorphism group of an embedding, as we shall see, is a much easier task.

For an oriented embedding one can distinguish two types of automorphisms: those that preserve the face orientations and those that reverse them. The orientation preserving automorphisms form a group which we shall call the rotational subgroup. Orientation reversing automorphisms however do not necessarily exist, but if they do, their number matches the number of orientation preserving automorphisms. Notice that this second set of automorphisms never forms a group, as it does not contain the identity element, which is orientation preserving. Both types can be illustrated by making an analogy with the spherical embedding of the cube graph. The total symmetry group of this embedding is the point group O_h which consists of the rotational subgroup O containing 24 orientation preserving rotational elements and its coset of 24 improper elements which reverse the orientation.

Depending on the existence of orientation reversing automorphisms, embeddings are classified as *reflexible* or *chiral*.[15] In a reflexible embedding there exist orientation reversing symmetry elements and the rotational subgroup necessarily forms a subgroup of index 2. In the chiral case however, no orientation reversing symmetries are present, so a chiral embedding and its orientation reversed counterpart form an enantiomeric pair.

We shall now first describe how all orientation preserving automorphisms of an embedding can be derived. For this purpose, every edge is again represented by a set of two oppositely oriented arcs. Every such arc belongs to exactly one of the oriented faces of the embedding. We start by choosing an arbitrary starting arc and map it on one of the $2E$ possible arcs. Such a mapping of an arc onto another one, exactly determines how the face containing the first arc has to be mapped on the face containing the second. Let us for example take the embedding of Figure 1 and map arc (1−2) onto arc (4−3). By doing so, face (1−2−6−7−8−4) is mapped onto face (4−3−7−6−5−1). Together with this face mapping, the images of all six arcs and vertices of the face (1−2−6−7−8−4) are also fixed. For the vertices for instance, one has to map 1 → 4, 2 → 3, 6 → 7, 7 → 6, 8 → 5, and 4 → 1. Of course, the mapping of the six arcs of face (1−2−6−7−8−4) also determines the images of their opposite arcs. For instance, because arc (8−4) is mapped onto arc (5−1), the opposite arcs (4−8) and (1−5) must also be identified. Therefore, the image of face (4−8−5−6−2−3) is (1−5−8−7−3−2), leading to some new arc and vertex mappings: 4 → 1, 8 → 5, 5 → 8, 6 → 7, 2 → 3, and 3 → 2. As long as in every step of this procedure the new vertex mappings are in correspondence with previous assignments, one just proceeds with the mapping procedure. When the images of all arcs have been determined and no discrepancies are found, one can conclude that the derived vertex permutation is indeed an automorphism and it can be stored. However, if there are contradictions in the vertex mappings, one directly concludes that the initial arc to arc mapping does not lead to an automorphism. In both cases one starts the same procedure over again until the starting arc has been mapped on all $2E$ possible arcs. As we have seen, the mapping of two arcs onto each other is sufficient to completely determine an automorphism, if it exists. This clearly puts an upper bound of $2E$ on the order of the rotational subgroup of orientation preserving automorphisms.[15]

The orientation reversing automorphisms, if they exist, can be found in a similar way, with the only difference that the face mapping is now performed onto the orientation reversed faces. For our case this means that if arc (1−2) is mapped onto arc (4−3), face (1−2−6−7−8−4) is mapped onto the orientation reversed face (4−3−2−6−5−8). The remainder of the procedure stays completely the same. Again, we find an upper bound of $2E$ on the set of orientation reversing automorphisms. The order of the total automorphism group of an embedding can therefore never exceed $4E$. Embeddings which realize this upper bound are called regular embeddings[16] and can be seen as generalizations of the Platonic solids to higher genus surfaces.[17]

We just discussed how the rotational subgroup respectively full symmetry group of an embedding can be calculated. Checking for isomorphisms can be done in a similar way. Two oriented embeddings are said to be isomorphic if there exists a mapping of the oriented faces of the first embedding onto the oriented faces of the second. Now that isomorphism is defined, we can discuss the relationship between the symmetry of an embedding and the number of duplicates found during the generation process. This number of duplicates corresponds to the number of ways the embedding can be labeled. Let us recall that our algorithm starts from a given labeled graph which is completely described by its adjacency list. The number of possible ways to label this graph (consistent with its adjacency list) is given by the order of the automorphism group of the graph, denoted as $|Aut(\text{graph})|$. If one wants to count the number of different labelings of an oriented embedding, one should realize that it is not only described by an adjacency list but also by a set of labeled oriented faces. In fact, a listing of the oriented faces will be sufficient as they also include all adjacency information. Let us start by considering an embedding with trivial C_1 symmetry. In this case all different consistent graph labelings lead to different sets of labeled oriented faces, so the number of labelings (= duplicates) for a nonsymmetrical oriented embedding equals $|Aut(\text{graph})|$. If the embedding however has a nontrivial symmetry, some of the $|Aut(\text{graph})|$ labelings lead to the same set of labeled oriented faces. The number of such equivalent labelings is obviously given by the order of the rotational subgroup of the embedding, denoted as $|Aut^+(\text{embedding})|$. The number of duplicates therefore equals

$$\#\,\text{duplicates} = \frac{|Aut(\text{graph})|}{|Aut^+(\text{embedding})|} \tag{4}$$

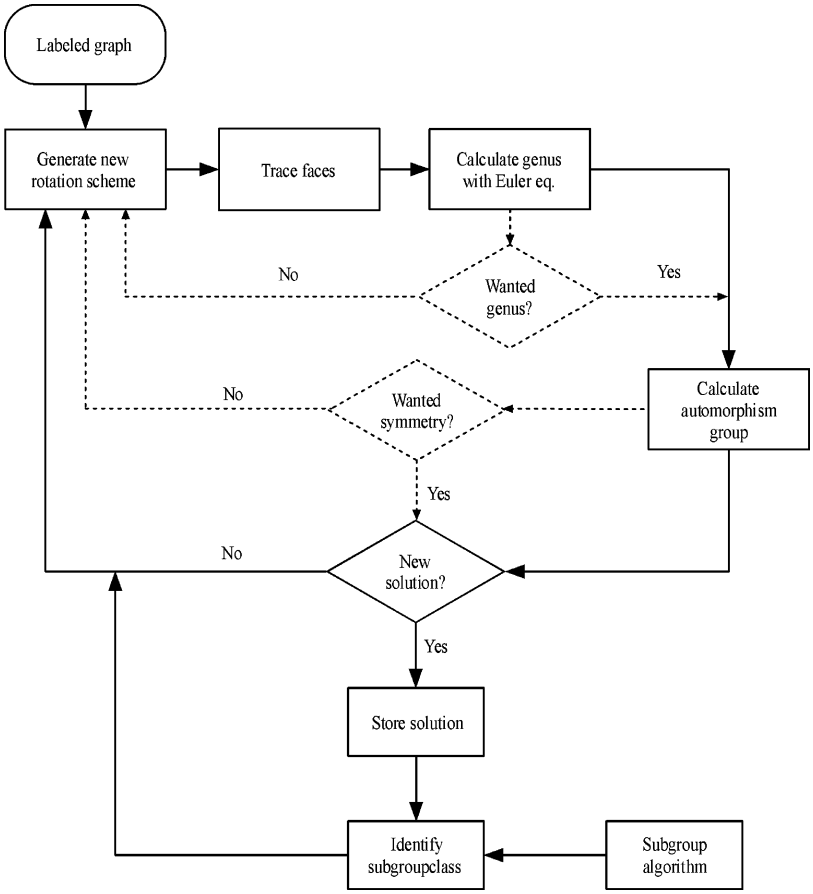

Figure 2. Flowchart of the program to find all symmetry-distinct oriented 2-cell embeddings.

which also includes the case with C_1 symmetry. In this way we can derive a simple dimensional formula, which relates the number of rotation schemes with the symmetries of the symmetry-distinct solutions:

$$\#\ \text{rotation schemes} = \sum_{\substack{\text{distinct} \\ \text{embeddings}}} \frac{|Aut(\text{graph})|}{|Aut^{+}(\text{embedding})|} \quad (5)$$

3.3. Implementation. In Figure 2 we show a flowchart of the resulting algorithm for deriving all symmetry-distinct oriented 2-cell embeddings. As a first step, a routine was written which produces all possible rotation schemes for a given graph and derives the faces for every solution. (See section 3.1.) As an oriented embedding is completely characterized by its oriented faces, every solution can be stored as a list of these faces. In most cases however, this leads to an astronomical number of solutions which have to be stored. To reduce this problem, once the faces of a new rotation scheme have been calculated, the symmetry of the embedding is derived, and it is instantaneously compared to the already stored solutions. (See section 3.2.) Only if the embedding is not isomorphic to a previously found solution, it will be added to the list. Of course we could check for isomorphism with every single solution in the list, but as this isomorphism testing is the most time-consuming step, it is advisable to use some simple heuristics to simplify the procedure. For instance, it is only necessary to check for isomorphism between solutions of the same genus or, even better, between solutions which have the same number of faces for each face size. If these heuristics are not yet sufficient to rule out isomorphism, the orders of their rotational subgroup and full symmetry group can also be compared. It is only after all these checks have been successfully passed, that one has to invoke the actual routine which checks for isomorphism. If one is only interested in embeddings on a surface with a given genus, one can check the genus of the embedding directly after its faces have been traced and decide, based on this genus, if the new embedding has to be investigated further or not. The same can be done if one is interested in embeddings with a certain symmetry. One just has to include a decision step after the symmetry group has been calculated. If these routines are all properly implemented and all rotation schemes have been investigated, one ends up with a complete list of all possible symmetry-distinct embeddings.

Table 2. Listing of All Symmetry-Distinct Embeddings of the Cube Graph

order →	48	24	16	12	8	6	4	3	2	1	total
$g = 0$	1	-	-	-	-	-	-	-	-	-	1
$g = 1$	1	-	2	-	-	1	1	-	-	-	5
$g = 2$	-	-	-	2	-	1	3	-	2	-	8
total	2	-	2	2	-	2	4	-	2	-	14

In Table 2 we list the numbers of all symmetry-distinct embeddings for the cube graph, sorted according to their genus and the order of their full symmetry group. As one can see, from the $2^8 = 256$ possible combinatorial rotation schemes only 14 remain after symmetry-reduction was applied. We can now use eq 5 to check these results. The automorphism group of the graph has order 48 and is isomorphic to O_h. Notice that all 14 embeddings are found

to be reflexible, so the orders of the corresponding rotational subgroups are half the orders listed in Table 2. This gives us

$$2*(48/24) + 2*(48/8) + 2*(48/6) + 2*(48/3) + 4*(48/2) + 2*(48/1) = 256 \quad (6)$$

which verifies eq 5. The two most symmetrical and moreover regular embeddings are the well-known embedding on the sphere and the toroidal embedding shown in Figure 1. This last embedding and an alternative all-hexagonal embedding on the torus with order 16 are also shown in ref 7.

We already know that the symmetry groups of all embeddings are necessary subgroups (with maximum order 4E) of the symmetry group of the graph. In the next section we discuss an algorithm that finds all subgroup conjugacy classes of the graph symmetry group and appoints every embedding to one of these subgroup classes.

4. SYMMETRY CLASSIFICATION OF THE EMBEDDINGS

4.1. Calculating All Subgroups of a Graph. There are a number of programs which are able to calculate the automorphism group of a graph.[14] Starting from this group, we will now calculate the class structure of the group and all its subgroups and subgroup classes.

Two elements A and B belong to the same class if and only if there exists a similarity transform $X^{-1}AX = B$, with X an element of the group, which transforms element A into element B. So, to obtain the group classes, we calculated all possible similarity transforms for each element and stored the results in matrix form. Using this matrix, the group elements can be subdivided into classes.

The subgroups and subgroup conjugacy classes are calculated using a generator based procedure. This procedure makes intensive use of the group multiplication table, so it is advisable to calculate and store it first. The strategy of the procedure is based on the following two lemmas:

1. A group with n generators has necessarily subgroups with $n - 1$ generators.
2. A subgroup can have more generators than its parent group. [As an example: D_{4h} has only two generators but its subgroup D_{2h} has three.]

Therefore our algorithm will be divided into several steps. In each of these steps, the subgroups and subgroup classes with n generators are calculated starting from those with $n - 1$ generators. As long as we find at least one new subgroup with n generators, we continue the procedure and increase the number of generators. If however no new n-generator subgroup is found, the procedure is stopped, because following lemma 1 there will be no subgroups with more generators. Subgroups are stored as binary lists with a length equal to the group order of the graph. In this list, every entry corresponds to a group element and equals 1 if the group element is part of the subgroup under consideration and 0 if it is not. This representation is well adapted for isomorphism and subgroup checking.

As a first step, we derive all "cyclic" or 1-generator subgroups. These subgroups are simply formed by taking the powers of the group elements. Let us remind that two subgroups belong to the same conjugacy class if there exists a group element X such that the similarity transforms $X^{-1}AX$ of all elements A of the first subgroup form the second subgroup or $H \sim H'$: $X^{-1}HX = H'$. As a result, two "cyclic" subgroups generated by elements belonging to the same group class will also belong to the same subgroup class. Having found all 1-generator subgroups, we can now turn to the next step in our algorithm and search for the 2-generator subgroups. For this purpose, it is not necessary to try all sets of two generators. In fact, all n-generator subgroups can be constructed starting from the subgroup classes with $n - 1$ generators. We just have to take a representative of each of these $(n - 1)$-generator subgroup classes and repeatedly add an extra generator, which does not yet belong to the representative. The procedure to find the corresponding n-generator subgroup, is as follows. We know that all elements belonging to the $(n - 1)$-generator subgroup and those belonging to the cyclic subgroup of the added generator are definitely part of the newly generated set. This set however does not necessarily lead to a new subgroup, as it may well generate the complete group at once. The generated subgroup can be found using the multiplication table. One starts by extracting from this multiplication table the submatrix formed by the rows and columns corresponding with the elements of our set. If this submatrix contains entries which correspond to new elements that are not yet part of our set, we add them to the set and construct a new submatrix using this extended set. This procedure is repeated until at a certain moment no new elements are found. The resulting set, if different from the complete group, forms a subgroup with n generators. Our next task is to check if this subgroup was already found. This can easily be done by comparing the binary list of the newly found subgroup with the binary lists of all previously discovered subgroups. If the binary list was not yet found, we have a new subgroup and add it to the subgroup list. This new subgroup H necessarily belongs to a new subgroup class and all conjugate subgroups $H' = X^{-1}HX$ belonging to the same subgroup class are immediately calculated by running over all X and stored in the subgroup list. We now return to our $(n - 1)$-generator representative and add a different generator. The procedure is repeated until all $(n - 1)$-generator representatives are combined with all possible generators. Now that all subgroups and subgroup classes with n generators are found, we can turn to the next step and search for all subgroups with $n + 1$ generators starting from representatives with n generators and again add to those all possible generators to see if they form some new $(n + 1)$-generator subgroups. Following lemma 1, the procedure can be stopped when during a step of the algorithm no new n-generator subgroups are found. In Figure 3 we give a schematic overview of this subgroup algorithm.

4.2. Symmetry Classification. In the previous section we described how to find all subgroups and subgroup classes of a given group. We can now use this information to classify the symmetry-distinct embeddings. As discussed earlier, the symmetry group of an embedding is always a subgroup of the full symmetry group of the graph, so every embedding can be appointed to one of the subgroup conjugacy classes of the graph. To find the corresponding subgroup class, the symmetry group of the embedding is first identified with one of the listed subgroups of the graph. In the subgroup algorithm, all subgroups were automatically sorted into subgroup conjugacy classes, so once the corresponding

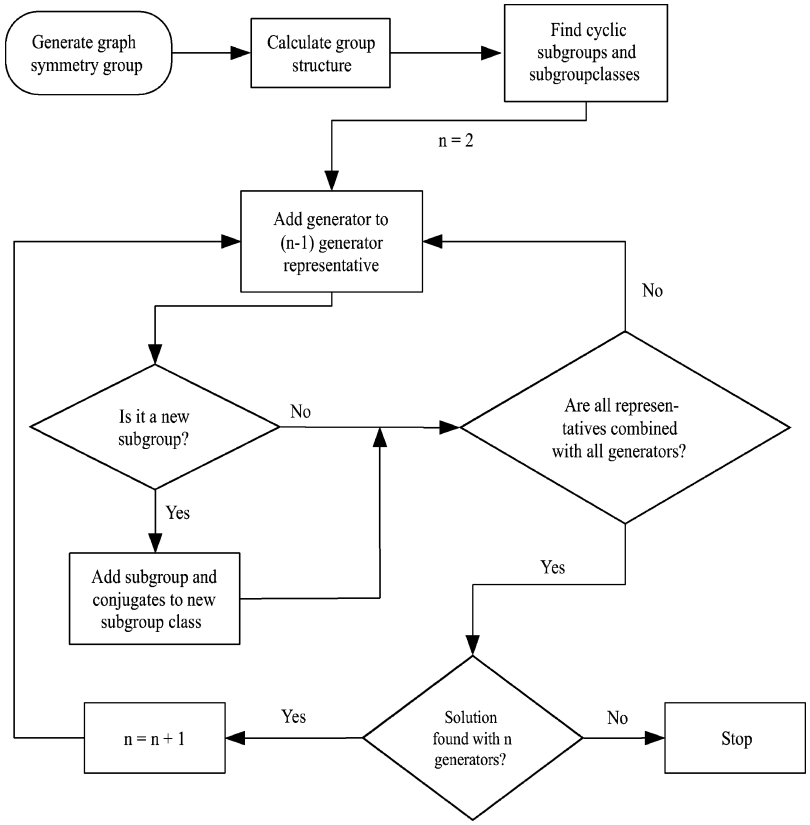

Figure 3. Flowchart of the routine used to find all subgroups and subgroup classes.

Table 3. Classification of All Symmetry-Distinct Solutions of the Cube Graph According to Their Genus and Subgroup Conjugacy Class

class →	O_h	D_{4h}	D_{3d}	D_3	D_2	$C_{2v}^{(a)}$	$C_{2v}^{(b)}$	σ_d	total
$g = 0$	1	-	-	-	-	-	-	-	1
$g = 1$	1	2	-	1	-	-	1	-	5
$g = 2$	-	-	2	1	1	1	1	2	8
total	2	2	2	2	1	1	2	2	14

subgroup is found, its conjugacy class is also known. In Figure 2 we see that this symmetry reduction can be applied directly after the identification of a new embedding.

In Table 3 we classified all symmetry-distinct embeddings of the cube graph according to their genus and subgroup conjugacy class. The group of the full graph has 48 elements and corresponds to O_h. Notice that although three embeddings have C_{2v} symmetry they do not all belong to the same conjugacy class. For the $C_{2v}^{(a)}$ class the 2-fold C_2 axis corresponds to the square of a C_4-axis of the O_h group and both planes are of type σ_d. For the $C_{2v}^{(b)}$ class however, the C_2 axis corresponds to the interstitial C_2' type element of the O_h group and has one plane of type σ_d and one of type σ_v. Also note that not all subgroups of O_h give rise to embeddings. As an example the index 2 subgroups O, T_d, and T_h are not realized.

In the next section we apply our algorithms to the highly symmetrical Möbius-Kantor graph and use some of its embeddings as building blocks for new negatively curved carbon allotropes.

5. CASE STUDY: THE MÖBIUS-KANTOR GRAPH

The Möbius-Kantor graph[6,7] (MK-graph) is characterized as the only cubic symmetric graph on 16 vertices. The term cubic means that every vertex is trivalent, and the term symmetric denotes that the symmetry group of the graph acts transitively on its arcs, in other words, for every set of two arcs, there exists a graph automorphisms which maps the first arc onto the second. The MK-graph is also known to be isomorphic with the generalized Petersen graph $GP_{8,3}$. In Figure 4 two alternative drawings of this MK-graph are shown.

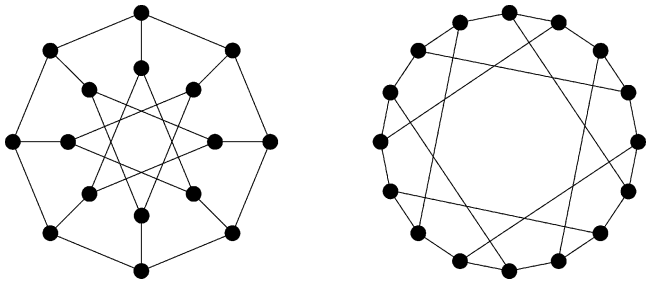

Figure 4. Two alternative drawings of the Möbius-Kantor graph.

5.1. The Möbius-Kantor Graph and Its Embeddings. To find all oriented 2-cell embeddings we numbered the vertices of our graph and calculated all $2^{16} = 65536$ possible rotation schemes. From these schemes we derived the corresponding embeddings and sorted them according to their genus and the order of their full symmetry group. The results are shown in Table 4. The order of the symmetry group of an embedding is bounded from above by $4E = 4 \cdot 24 = 96$. All orders are therefore necessarily divisors of 96. The derived solutions were instantaneously checked for isomorphism, and the resulting oriented symmetry-distinct solutions are listed in Table 5. Let us remind that the term *oriented* means that embeddings belonging to an enantiomeric pair are considered as distinct structures.

Table 4. All Possible Oriented 2-Cell Embeddings of the Möbius-Kantor Graph

order →	96	48	32	24	16	12	8	6	4	3	2	1	total
$g = 1$	-	-	6	-	-	-	-	-	-	-	-	-	6
$g = 2$	2	-	-	-	-	16	24	32	96	64	384	96	810
$g = 3$	2	-	6	-	-	16	24	224	864	192	4800	14784	20912
$g = 4$	-	-	-	16	-	32	48	288	480	128	9408	33408	43808
total	4	-	12	16	-	64	96	544	1440	384	14592	48384	65536

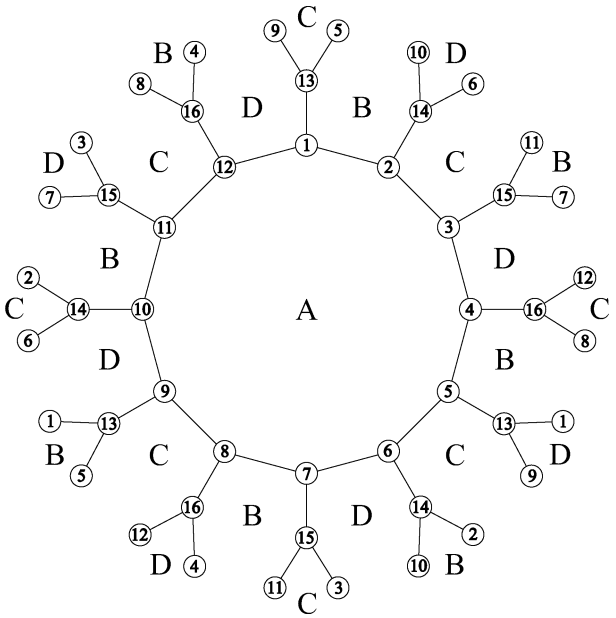

Figure 5. Regular embedding of the MK-graph on the triple torus.

Table 5. Symmetry-Distinct Oriented 2-Cell Embeddings of the Möbius-Kantor Graph

order →	96	48	32	24	16	12	8	6	4	3	2	1	total
$g = 1$	-	-	1	-	-	-	-	-	-	-	-	-	1
$g = 2$	1	-	-	-	-	1	1	1	2	2	5	2	15
$g = 3$	1	-	1	-	-	1	1	7	18	6	66	154	255
$g = 4$	-	-	-	2	-	2	2	11	10	4	109	348	488
total	2	-	2	2	-	4	4	19	30	12	180	504	759

Two of the embeddings in Table 5 especially draw our attention. They are the two embeddings which obtain the upper bound of $4E = 96$ symmetry elements and therefore embed regularly. The regular embedding on the triple torus consists of four faces ($A - D$) of size twelve and is depicted in Figure 5. The second regular embedding is formed by six octagonal faces ($E - J$) and maps on the double torus (Figure 6). Tables 6 and 7 give their corresponding rotation schemes.

The vertex numbering of the genus-2 embedding looks a little bit odd at first sight, but its meaning follows directly from a deeper connection between both regular embeddings. In fact, both structures are connected by a process called Petrie-dualization. This dualization process leaves the underlying graph of the embedding unchanged but replaces the faces of the embedding by a new set of faces, formed by taking Petrie-paths. A Petrie-path of an embedding is defined as a path where alternatively left and right turns are taken. As a consequence three consecutive edges of such a path never belong to the same parental face. In the case of a regular embedding, the Petrie-paths are closed, called Petrie-polygons, and form the faces of an alternative regular embedding. The connection between the numberings of our

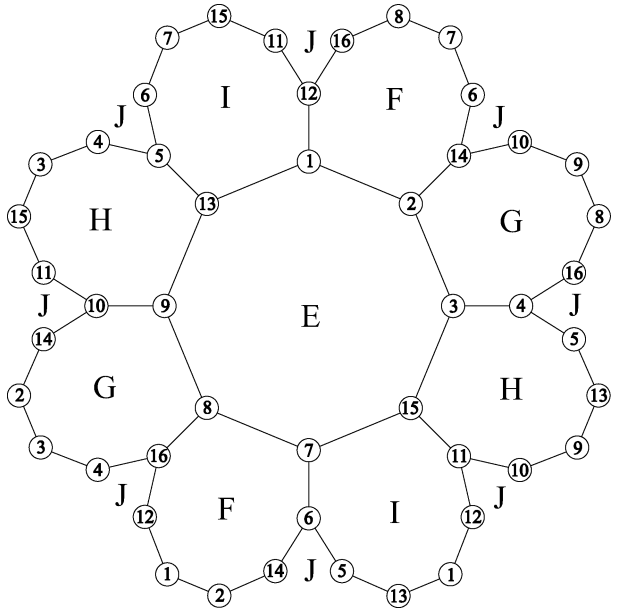

Figure 6. Regular embedding of the MK-graph on the double torus.

Table 6. Rotation Scheme of the Genus-3 Embedding of the MK-Graph

1. 2 13 12	5. 4 6 13	9. 8 10 13	13. 1 5 9
2. 1 3 14	6. 5 7 14	10. 9 11 14	14. 2 6 10
3. 2 4 15	7. 6 8 15	11. 10 12 15	15. 3 7 11
4. 3 5 16	8. 7 9 16	12. 12 1 16	16. 4 8 12

Table 7. Rotation Scheme of the Genus-2 Embedding of the MK-Graph

1. 2 12 13	5. 4 13 6	9. 8 13 10	13. 1 5 9
2. 1 3 14	6. 5 7 14	10. 9 11 14	14. 2 10 6
3. 2 15 4	7. 6 15 8	11. 10 15 12	15. 3 7 11
4. 3 5 16	8. 7 9 16	12. 1 16 11	16. 4 12 8

two regular surfaces can now easily be established. Let us start from the genus-3 embedding of Figure 5. Starting at arc (1−2) and taking a right turn we end up in arc (2−3). Our next turn needs to be to the left and leads to arc (3−15). Continuing this zigzag path one finds the Petrie-polygon (1−2−3−15−7−8−9−13). The other Petrie-polygons are traced similarly, every time starting from an arc which was not previously encountered. Because all arcs of a regular embedding are equivalent, all Petrie-polygons are equally sized and form an alternative regular embedding, in our case the regular genus-2 embedding of Figure 6. Notice that the Petrie-dual of an orientable embedding can be nonorientable and will therefore not necessarily be found by our algorithms.

A third interesting embedding is the all hexagonal embedding on the torus, shown in Figure 7. This embedding is *minimal* as it embeds the graph on the surface with lowest possible genus. Note that, although all faces are equally sized, this embedding is not regular. In fact, the embedding is reflexible, and its full symmetry group consists of only 32 elements.

5.2. Group Structure of the Möbius-Kantor Graph. The symmetry group of the MK-graph, which we shall call the MK-group, has order 96 and therefore coincides with the full symmetry groups of our regular embeddings. As a consequence, all group elements can be described by their action on these regular embeddings.

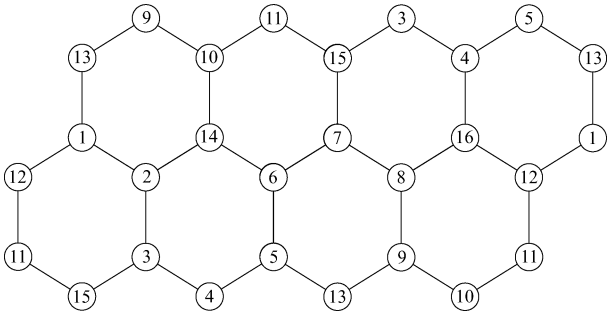

Figure 7. Minimal embedding of the MK-graph on the torus.

Using our algorithm to find the conjugacy classes of a group, the 96 automorphisms of the MK-group were divided into 13 classes:

$$E + 8C_{12} + 8C_6 + 8C_3 + 8C_{12}{}^5 + 2C_4 + C_2 + 6C_2' + 12\sigma_{||} + 12\sigma_{\perp} + 6C_4' + 12S_8{}^{(a)} + 12S_8{}^{(b)} \quad (7)$$

All these classes, except the last three, are easily described by their action on the genus-3 embedding. The C_{12} elements correspond to 12-fold rotations around axes going through the centers of the dodecagonal faces. Every axis leaves exactly one of these faces invariant and because the operations C_{12} and their inverses $(C_{12})^{-1}$ are part of the same conjugacy class, there are a total of $4 \cdot 2 = 8$ such elements. For instance, the C_{12} element through face A leads to the following automorphism:

$$C_{12}(A) \rightarrow (13,14,15,16)(1,2,3,4,5,6,7,8,9,10,11,12) \quad (8)$$

The eight C_3, C_6, and C_{12}^5 elements are in an analogous way related to respectively the fourth, second, and fifth powers of the C_{12} elements and their inverses. Notice that the 3-fold axes also leave four vertices invariant. Again we list the elements corresponding to rotations through face A:

$$C_3(A) \rightarrow (13)(14)(15)(16)(1,5,9)(2,6,10)(3,7,11)(4,8,12) \quad (9)$$

$$C_6(A) \rightarrow (13,15)(14,16)(1,3,5,7,9,11)(2,4,6,8,10,12) \quad (10)$$

$$C_{12}^5(A) \rightarrow (13,14,15,16)(1,6,11,4,9,2,7,12,5,10,3,8) \quad (11)$$

A C_4 element is formed by taking the third power of a C_{12} operation. It stabilizes all four dodecagonal faces so this class only contains two elements, the element C_4 and its inverse C_4^{-1}.

$$C_4 \rightarrow (1,4,7,10)(2,5,8,11)(3,6,9,12)(13,16,15,14) \quad (12)$$

The C_2 element stands on its own and is formed by taking the sixth powers of the C_{12} elements:

$$C_2 \rightarrow (1,7)(2,8)(3,9)(4,10)(5,11)(6,12)(13,15)(14,16) \quad (13)$$

The C_2' elements correspond to 2-fold rotations going through the edges on the genus-3 surface. Because every such operation leaves four edges invariant the total amount of operations equals $24/4 = 6$. As an example, the C_2' axis going through the edges (1−2), (4−5), (7−8), and (12−14) is described by

$$C_2' \rightarrow (1,2)(3,13)(4,5)(6,16)(7,8)(9,15)(10,11)(12,14) \quad (14)$$

The operations $\sigma_{||}$ are identified with symmetry planes going through vertices. As every such plane fixes 4 vertices and 2 edges, the class contains $24/2 = 12$ elements. For the plane going through the vertices 1, 2, 7, and 8 we have

$$\sigma_{||} \rightarrow (1)(2)(7)(8)(3,14)(4,10)(5,11)(6,15)(9,16)(12,13) \quad (15)$$

The other symmetry planes $\sigma_{\perp}$ leave no vertices invariant and run perpendicular through 2 edges. Consequently this class also contains twelve operations. The plane cutting edges (1−2) and (7−8) is characterized as

$$\sigma_{\perp} \rightarrow (1,2)(3,12)(4,11)(5,10)(6,9)(7,8)(13,16)(14,15) \quad (16)$$

To explain the last three classes, it is better to turn to the genus-2 regular embedding. The $S_8^{(a)}$ elements correspond to rotations around 8-fold axes going through the octagonal faces. Every such operation leaves two faces invariant, but because all odd powers of this operation belong to the same class there are a total of $6/2 \cdot 4 = 12$ such elements. For the $S_8^{(a)}$ operation through face E and J one has

$$S_8^{(a)}(E) \rightarrow (1,2,3,15,7,8,9,13)(4,11,6,16,10,5,12,14) \quad (17)$$

The six C_4' operations are easily seen to correspond to the second and sixth powers of $S_8^{(a)}$, leading to a total of $6/2 \cdot 2 = 6$ elements.

$$C_4'(E) \rightarrow (1,3,7,9)(2,15,8,13)(4,6,10,12)(5,14,11,16) \quad (18)$$

The fourth powers of the $S_8^{(a)}$ elements do not form new elements, but just the C_2 operation which was already found. The last class to be explained consists of the twelve $S_8^{(b)}$ type elements. Their action on the regular embeddings is not obvious at first sight, as the associated 8-fold axes do not intersect the surfaces of these embeddings. They can be found by following a path which takes alternately two right and two left turns. In this way we obtain a Hamiltonian cycle containing all 16 vertices. If we color the vertices of this cycle alternatively black and white, we introduce a kind of 8-fold operation through the center of this cycle which preserves the colors of the vertices. Let us start by considering the following element:

$$S_8^{(b)} \rightarrow (1,3,16,11,7,9,14,5)(2,4,16,11,15,8,10,6) \quad (19)$$

To retrieve this operation from the genus-3 surface, we start at vertex 1 and take two right turns to form path (1−2−3−4). The next two turns are to the left to form (1−2−3−4−16−12). In this way we eventually obtain the Hamiltonian cycle (1−2−3−4−16−12−11−15−7−8−9−10−

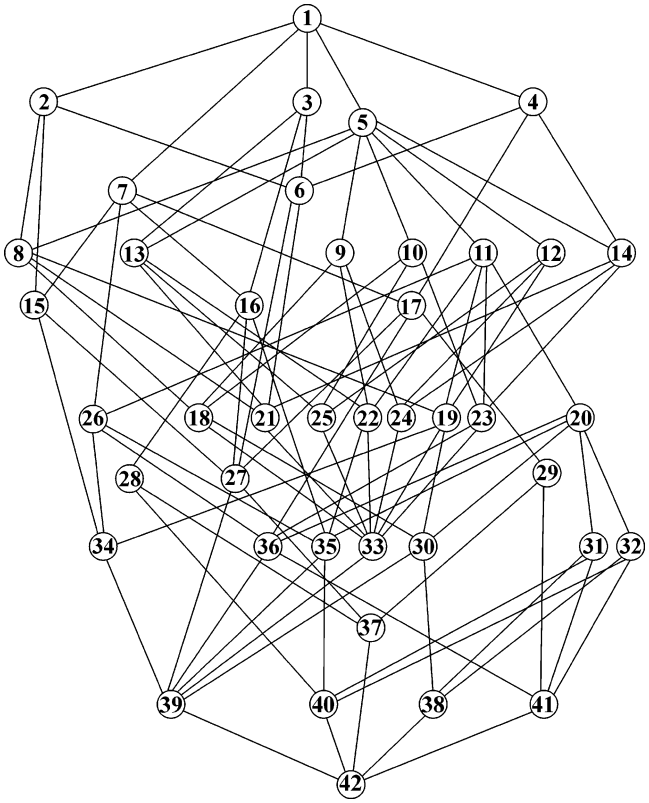

Figure 8. Genealogical tree for the full MK-group. The numbers refer to subgroups that are identified in Table 8. Subgroup orders are indicated in the left margin.

Table 8. Listing of the Subgroup Classes of the MK-Group[a]

1. (2) MK-group	15. (1) C_{12}	29. (2) D_3
2. (2) Aut^+(genus3)	16. (2) D_6	30. (2) D_2
3. (2) 48(bipartite) $\cong GL_2(3)$	17. (2) D_6	31. (2) D_2
4. (2) Aut^+(genus2) $\cong GL_2(3)$	18. (2) D_4	32. (2) D_2
5. (3) 32(torus) / 32.44	19. (2) $C_2 \times C_4$	33. (1) C_4'
6. (2) $SL_2(3)$	20. (2) D_{2h}	34. (1) C_4
7. (2) D_{12}	21. (2) Q_8	35. (2) D_2
8. (3) AES_{16} / 16.8	22. (2) D_4	36. (2) D_2
9. (2) D_8 / 16.12	23. (2) D_4	37. (1) C_3
10. (2) D_8 / 16.12	24. (1) $S_8^{(a)}$	38. (1) C_2'
11. (3) $C_2 \times D_4$ / 16.6	25. (1) $S_8^{(b)}$	39. (1) C_2
12. (2) M_8 / 16.11	26. (2) D_4	40. (1) $\sigma_{\|\|}$
13. (2) SD_{16} / 16.13	27. (1) C_6	41. (1) $\sigma_\perp$
14. (2) SD_{16} / 16.13	28. (2) D_3	42. (1) E

[a] Numbers in brackets denote the minimal number of generators necessary to generate the group.

14−6−5−13) which forms the two color preserving 8-vertex orbits of eq 19.

5.3. Subgroups of the Möbius-Kantor Graph. Starting from the MK-group, our subgroup algorithm found a total of 173 subgroups divided into 42 subgroup classes. The corresponding genealogical tree was constructed and is shown in Figure 8. In this tree a connection between two subgroups implies a direct group-subgroup relationship. In the genealogical tree the subgroup classes are presented as numbers. The subgroup classes corresponding to these numbers can easily be retrieved from Table 8. In this table the numbers between brackets denote the minimal number of generators necessary to generate each subgroup. Note that although the MK-group is generated by 2 generators there are three subgroups with 3 generators.

From the genealogical tree we see that the MK-group has five direct subgroup classes: three of order 48 (2, 3, and 4), one of order 32 (5), and also one of order 24 (7). The groups Aut^+(genus3) and Aut^+(genus2) correspond to the rotational subgroups of the regular genus-3 and regular genus-2 embeddings. The third subgroup class of order 48 does not appear as a rotational subgroup of an embedding but can be related to the bipartite character of the graph. This means that the vertices can be divided into two sets of nonadjacent vertices, correspondingly colored black or white. The graph automorphisms preserving the color of the vertices now naturally form this third and remaining halving subgroup. The groups Aut^+(genus2) and 48(bipartite) are also isomorphic with the general linear group $GL_2(3)$ of nonsingular two by two matrices over the field 3. The direct subgroup of order 32 can easily be identified as the full symmetry group of the unique toroidal embedding. The alternative numbering 32.44 is due to Hall and Senior,[18] who derived a numbering scheme for subgroups of order 2^n. The direct subgroup of order 24 is isomorphic with D_{12}, the twelve-fold dihedral group, and is the highest order subgroup belonging to the point-group family. The alternative subgroup class of order 24 can be identified with the special linear group $SL_2(3)$ of nonsingular two by two matrices over the field 3 and with determinant 1. There are a total of 7 classes of order 16: two of type D_8 (the 8-fold dihedral group), two of type SD_{16} (the semi-dihedral group on 16 elements), one of type AES_{16} (the almost-extra-special group on 16 elements), one of type M_8 (the metacyclic group generated by the direct product of an 8-fold and 2-fold cyclic group), and finally one of type $C_2 \times D_4$ (the direct product of the point groups C_2 and D_4). For the groups of lower order, two of them need some further explanation, namely the group $C_2 \times C_4$ (the direct product of the cyclic groups C_2 and C_4) and Q_8 (the quaternion group on 8 elements). All remaining classes correspond to some cyclic or dihedral point group.

5.4. Symmetry Classification of the Embeddings of the MK-Graph. Having found all symmetry-distinct embeddings and all subgroup classes of our MK-graph, we can now turn to the final step where all embeddings are classified according to the subgroup class of their symmetry group. The results are shown in Table 9 where the classification is based both on subgroup class and genus. We immediately notice that for only about half of the subgroup classes an embedding can be found exhibiting that specific symmetry. Some of the remaining classes still have a relationship to embeddings as they correspond to the rotational subgroup of some embedding. These classes are indicated in Table 9 by using an asterisk * instead of the usual − symbol on the line which lists the total amount of solutions for each symmetry type. The remaining subgroups have a non-topological origin and correspond to some other properties of the corresponding graph. For instance, like discussed before, the index-2 subgroup class with number 3 can be linked to the bipartite character of our graph.

6. MOLECULAR REALIZATIONS

During the last two decades interest has risen in the prediction of new carbon allotropes, stimulated by the discovery of Buckminsterfullerene C_{60} and carbon nanotubes. In recent years some highly symmetrical carbon analogues,

Table 9. Listing of All Symmetry-Distinct Embeddings According to Their Genus and Corresponding Subgroup-Conjugacy-Class

class →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$g=1$	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-
$g=2$	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$g=3$	1	-	-	-	1	-	-	-	-	-	-	-	-	-	-	1
$g=4$	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	1
total	2	*	-	*	2	-	2	-	-	*	-	*	-	-	*	2

class →	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$g=1$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$g=2$	1	-	-	1	-	-	-	-	-	-	-	1	-	-
$g=3$	-	-	-	1	-	-	-	-	-	-	-	6	1	2
$g=4$	1	-	-	-	-	1	1	-	-	-	1	5	5	-
total	2	-	-	2	-	1	1	-	-	-	1	12	6	2

class →	31	32	33	34	35	36	37	38	39	40	41	42	1 → 42
$g=1$	-	-	-	-	-	-	-	-	-	-	-	-	1
$g=2$	1	1	-	-	-	-	2	2	-	3	-	2	15
$g=3$	7	7	-	-	1	1	6	16	-	21	29	154	255
$g=4$	4	4	-	-	1	1	4	6	-	72	31	348	488
total	12	12	*	-	2	2	12	24	*	96	60	504	759

decorating negatively curved minimal surfaces have been proposed.[1,2,4] We shall see that the Möbius-Kantor graph has the ideal characteristics to serve as a template for some newly proposed negative curvature carbon allotropes. Because all vertices of the MK-graph are trivalent, they can be identified with sp^3 hybridized carbon atoms.

It is generally known that there exists a uniform and trivalent tessellation of the plane as it can be decorated by a set of regular hexagons. Its chemical carbon analogue is the graphite sheet. Cutting a parallelogram out of this graphite sheet, one can form an all hexagon nanotube by identifying one pair of parallel sides. In a similar way, an all hexagon torus can be constructed if the second pair of parallel sides is also identified. By introducing faces of size smaller than six into the graphite sheet, the surface gets a positive curvature and can eventually form completely closed allotropes such as for instance Buckminsterfullerene C_{60}. If however faces with a size greater than six are introduced, the sheet obtains a negative curvature and only embeds on the higher genus surfaces ($g \geq 2$).

In the case of the Möbius-Kantor graph, the girth (size of smallest cycle) equals six and the lowest genus surface on which the graph can be hypothetically embedded is the torus. From our calculations we know that such a unique mapping does indeed exist. It corresponds to the all hexagon embedding shown in Figure 7. Seen as a toroidal carbon structure, this realization is quite implausible as its small dimensions introduce a serious amount of strain. In principle, part of the strain can be removed by an opening of the torus to form a small cylinder. Connecting several of these cylinders together leads to a nanotube-like structure. The two nanotubes which can be formed starting from our toroidal embedding correspond to the (4,0) zigzag tube and the (2,1) chiral tube.[19] These tubes however have extremely small diameter and are therefore chemically unlikely.

From our previous calculations we see that all other embeddings have some faces of size greater than six and therefore map on higher genus surfaces ($g \geq 2$). The embedding of a such a single unit cell however, as in the toroidal case, gives rise to a serious amount of strain.

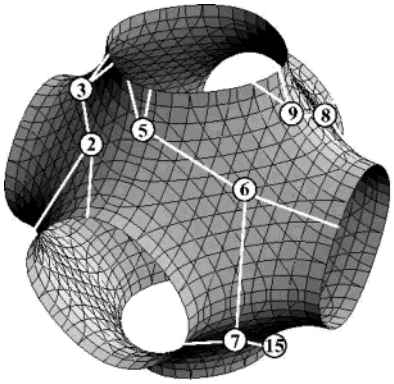

Figure 9. Regular genus-3 embedding of the MK-graph on the Plumber's Nightmare.

Therefore, instead of considering single unit embeddings, we will try to map the graph on an appropriate negatively curved surface, which can be used to form periodic lattices. In this case the symmetry of our unit cell must belong to the family of lattice groups. The first step of our method therefore consists of identifying the highest possible symmetry which a periodic embedding can obtain. For this purpose, we identify the rotational subgroup of our embedding and determine its highest order lattice subgroup. Once this subgroup is found, its symmetry elements can be used to determine the spatial locations of some or all of its vertices. Following this procedure we usually gather sufficient information to determine a spatial realization. In the next two sections we apply this method to find realizations for the genus-3 and genus-2 regular embeddings.

6.1. Realization of the Regular Genus-3 Embedding. The symmetry group of the genus-3 embedding (= the MK-group) is a general permutation group which is not isomorphic to any point group. It is therefore obvious that a spatial realization can never achieve this combinatorial symmetry. The highest order lattice subgroup class of the rotational subgroup Aut^+(genus3) can be retrieved using the genealogical tree of Figure 8. It is identified as class 18 which contains three subgroups of D_4 symmetry. We can freely choose one of the subgroups and list its symmetry elements to deduce the spatial positions of the vertices:

$$E \rightarrow (1)(2)(3)(4)(5)(6)(7)(8)(9)(10)(11)(12)(13)(14)(15)(16)$$

$$C_2' \rightarrow (1,6)(2,14)(3,10)(4,9)(5,13)(7,12)(8,16)(11,15)$$

$$C_2' \rightarrow (1,12)(2,16)(3,4)(5,15)(6,7)(8,14)(9,10)(11,13)$$

$$C_2' \rightarrow (1,2)(3,13)(4,5)(6,16)(7,8)(9,15)(10,11)(12,14)$$

$$C_2' \rightarrow (1,8)(2,7)(3,15)(4,11)(5,10)(6,14)(9,13)(12,16)$$

$$C_2 \rightarrow (1,7)(2,8)(3,9)(4,10)(5,11)(6,12)(13,15)(14,16)$$

$$C_4 \rightarrow (1,14,7,16)(2,6,8,12)(3,5,9,11)(4,13,10,15)$$

$$C_4^3 \rightarrow (1,16,7,14)(2,12,8,6)(3,11,9,5)(4,15,10,13)$$

Let us start by choosing a negatively curved surface on which our embedding can be realized. As our subgroup has D_4 symmetry it must be possible to find an embedding on the genus-3 *Plumber's Nightmare* (Figure 9) which in its undecorated form exhibits O_h symmetry. The genus of this

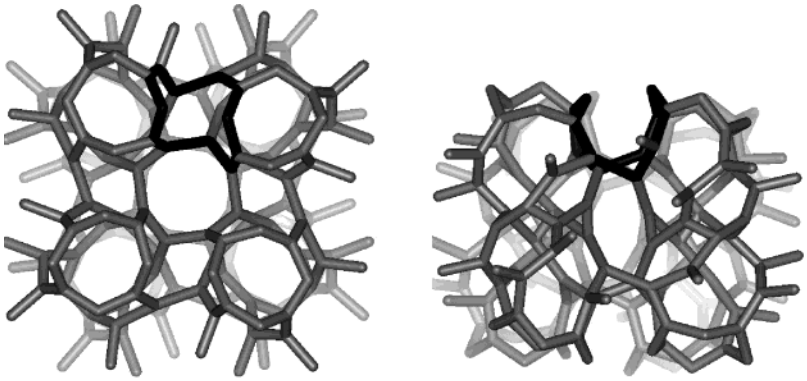

Figure 10. A 2 × 2 × 2-supercell of the genus-3 regular embedding viewed along the C_4 axis (left) and a C_2' axis (right). One dodecagonal face is indicated in black.

surface can easily be shown to be three as the identification of the three pairs of opposite holes leads to a triple torus. From the above listed vertex permutations we see that the 16 vertices of our MK-graph are divided into two orbits of eight vertices:

$$\text{Orbit 1} \rightarrow (2, 6, 8, 12, 1, 14, 7, 16)$$

$$\text{Orbit 2} \rightarrow (3, 5, 9, 11, 13, 10, 15, 4)$$

To determine all vertex locations it is therefore sufficient to fix the position of one vertex of each orbit. In Figure 9 we show such a realization where the vertices of the first orbit are fixed on the 3-fold axes of the original Plumber's Nightmare surface, and the vertices of the second orbit are positioned between an axial and equatorial hole. In traveling counterclockwise around the C_4 axis in the z-direction, one encounters successively the vertices (3−2−5−6−9−8−11−12) on the north side and the vertices (13−14−10−7−15−16−4−1) on the south side of the surface. Now that all vertex locations are determined, we turn to the step of locating the edges. Under the D_4 symmetry they are subdivided into 5 orbits:

$$\text{Orbit 1} \rightarrow (2-3), (6-5), (8-9), (12-11), (1-13), (14-10), (7-15), (16-4)$$

$$\text{Orbit 2} \rightarrow (2-14), (6-7), (8-16), (12-1)$$

$$\text{Orbit 3} \rightarrow (2-1), (6-14), (8-7), (12-16)$$

$$\text{Orbit 4} \rightarrow (3-4), (5-13), (9-10), (11-15)$$

$$\text{Orbit 5} \rightarrow (3-15), (5-4), (9-13), (11-10)$$

As there are only two vertex orbits it will be sufficient to localize only the edges starting from one of their representatives. We will choose vertices 5 and 6 respectively as they are positioned on the octant that is best visualized in Figure 9. The edges (6−5) and (6−7) corresponding to the edge orbits 1 and 2 are easily drawn on our surface without crossing any boundary. Following the rotation scheme of Table 6, the position of the third edge (6−14) leaving vertex 6 can be easily derived. It corresponds to the third edge orbit and leaves the unit cell at the right-hand side to appear again at the left-hand side. Notice that vertex 6 is located at the north side and vertex 14 at the south side, so there exists a mismatch between two equatorial neighboring subunits, which in this case equals 90°. The last two remaining edge-orbits again connect vertices from the north side and south side. These edges definitely have to cross the upper boundary as the already partially embedded graph blocks all other paths. Their relative positions can be deduced from Table 6. The mismatch between vertex 5 and vertices 13 and 4 are 90° and 180°, respectively. Having located all vertices and edges our spatial realization is completely determined. It should therefore be possible to locate our four dodecagonal faces. They however do not appear on the single unit embedding of Figure 9 as every face is distributed over more then one translational subunit. Let us for instance start with the arc (2−3) and follow the path of its corresponding face (1−2−3−4−5−6−7−8−9−10−11−12). To get to vertex 4 we already have to move to the above lying unit cell. It is however a short stay, as we have to return to our original unit cell when traveling to vertex 5. All together, when we have finished our complete facial path, we have visited a total of four unit-cells, so in order to visualize the dodecagonal face, we should depict more than one translational cell. Figure 10 shows a 2 × 2 × 2-supercell viewed along the C_4 and C_2' directions. Both structures were optimized using MM+ molecular modeling. Notice that the boundaries of Figure 10 slightly depass the supercell as all vertices and edges which are directly linked to vertices of the supercell are also included. One of the dodecagonal faces is emphasized in black and can be seen to be positioned between subunits.

6.2. Realization of the Regular Genus-2 Embedding. Investigation of the genealogical tree shows that the highest lattice subgroup class of the rotational subgroup Aut^+(genus2) of the genus-2 regular embedding corresponds to class 17 containing four D_6 subgroups. In Figure 11 we depict a 2D tubular system exhibiting this D_6 symmetry. It can be seen as a graphite honeycomb lattice where the bonds (dashed lines) are replaced by tubes which meet at the positions of the vertices. The parallelogram shows the actual unit cell of the lattice. The four tubes which cross the boundaries of this unit cell are pairwise connected and thus obviously lead to a genus-2 network. However, to fix the positions of our vertices we use the hexagonal prism (indicated by dashed lines) as unit cell. Under the operations of the D_6 subgroup, the vertices of the MK-graph are divided into 2 orbits, one of size 12 and one of size 4. The points belonging to the first orbit (vertices 1 to 12) can be placed around the central hole of the hexagonal unit cell. Those belonging to the second orbit (vertices 13 to 16) can be placed on the corners of the

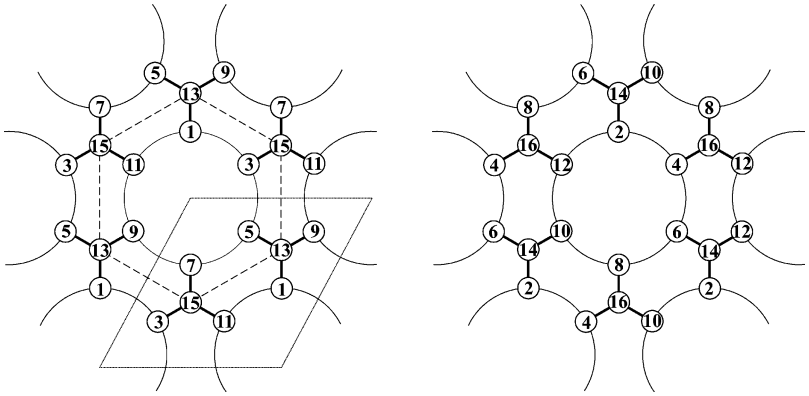

Figure 11. Embedding of the regular genus-2 embedding viewed along the C_6 axis. (Left) top view, (right) bottom view.

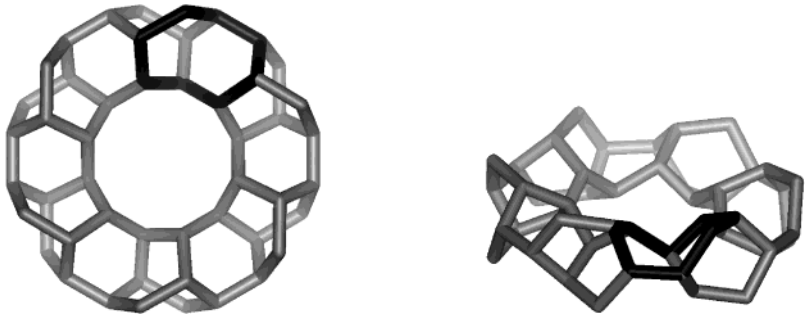

Figure 12. 3D realization of the regular genus-2 embedding viewed along two different directions. One octagonal face is indicated in black.

hexagonal unit cell, corresponding to the top and bottom of the junctions of our network. Notice, that there are twelve such corners, but only four of them are nonequivalent. In Figure 11 we show the placement of these vertices. The left side gives a top view of the tubular network viewed along the 6-fold direction. The right side gives the alternative view from the bottom. The placement of the edges is now an easy task. They are partitioned into three orbits, one of length 12 and two of length 6. The twelve edges of the first orbit correspond to the edges connecting the twelve vertices located on the internal hole to the vertex on the closest corner position. The other two edge orbits connect the upper six and lower six vertices of the first vertex orbit, thereby forming a kind of zigzag pattern going around the hole of the hexagonal unit-cell. This cycle of length twelve exactly corresponds to one of the dodecagonal faces of the genus-3 embedding. In this case however it is not realized as a face as it goes around the central hole and is therefore noncontractible. In Figure 12 we show a 3D realization of the hexagonal unit cell viewed along two directions. Notice that vertices of neighboring unit cells which are directly connected to atoms of the hexagonal unit cell are also included. The structure was optimized using MM+ molecular modeling, and one of the octagonal faces is indicated in black.

7. CONCLUSIONS

Our treatment has made effective use of group theory to find all symmetry-distinct embeddings of a given graph and classify them according to genus and subgroup class. Group theory has also been used to achieve molecular realizations, based on the highest point groups that the embedding and the orientable surface have in common. This procedure has been illustrated for the hitherto unsolved case of the Möbius-Kantor graph.

The investigated example further illustrates that the relationship between the group of the graph and the list of its embeddings raises many questions. Clearly not all subgroups correspond to embeddings, while it may be the case that a given subgroup is realized in many different embeddings. What is the most striking though is the contrast between the clear hierarchical group structure and the apparent lack of order in the set of embeddings. This remarkable asymmetry will be the subject of further study.

ACKNOWLEDGMENT

E. Lijnen holds a doctoral fellowship from the Belgian Science Foundation FWO.

REFERENCES AND NOTES

(1) King, R. B. Negative curvature surfaces in chemical structures. *J. Chem. Inf. Comput. Sci.* **1998**, *38*(2), 180−188.
(2) King, R. B. Novel highly symmetrical trivalent graphs which lead to negative curvature carbon and boron nitride chemical structures. *Discrete Math.* **2002**, *244*, 203-210.
(3) Gross, J. L.; Tucker, T. W. *Topological Graph Theory*; Dover Publications Inc.: New York, 2001.
(4) Ceulemans, A.; King, R. B.; Bovin, S. A.; Rogers, K. M.; Troisi, A.; Fowler, P. W. The heptakisoctahedral group and its relevance to carbon allotropes with negative curvature. *J. Math. Chem.* **1999**, *26*, 101−123.
(5) Ceulemans, A.; Lijnen E.; Ceulemans L. J.; Fowler P. W. The tetrakisoctahedral group of the Dyck graph and its molecular realization. *Mol. Phys.*, in press.
(6) Marusic, D.; Pisanski, T. The remarkable generalized Petersen graph G(8, 3). *Math. Slovaca* **2000**, *50*(2), 117−121.
(7) Marusic, D.; Pisanski, T. Symmetries of hexagonal molecular graphs on the torus. *Croat. Chem. Acta* **2000**, *73*, 969−981.
(8) Brahana, H. R. Systems of circuits of two-dimensional manifolds. *Ann. Math.* **1921**, *23*, 144−168.
(9) Ceulemans, A.; Lijnen E. The polyhedral state of molecular matter. *Eur. J. Inorg. Chem.* **2002**, *7*, 1571−1581.
(10) Federico, P. J. *Descartes on Polyhedra*; Springer: New York, 1982.
(11) Heffter, L. Über das Problem der Nachbargebiete. *Math. Ann.* **1891**, *38*, 477−508.

(12) Edmonds, J. A combinatorial representation for polyhedral surfaces. *Am. Math. Soc. Not.* **1960**, *7*, 646.
(13) Mohar, B.; Thomassen, C. *Graphs on Surfaces*; Johns Hopkins University Press: Baltimore, 2001.
(14) McKay, B. Nauty: A program for graph isomorphism testing. Available at http://cs.anu.edu.au/people/bdm/
(15) Conder, M.; Dobcsanyi, P. Determination of all regular maps of small genus. *J. Comb. Theory B* **2001**, *81*(2), 224−242.
(16) McMullen, P.; Schulte, E. *Abstract Regular Polytopes*; Cambridge University Press: Cambridge, UK, 2002.
(17) King, R. B. Platonic tesselations of Riemann surfaces as models in chemistry: nonzero genus analogues of regular polyhedra. *J. Mol. Struct.* **2003**, *656*, 119−133.
(18) Hall, M.; Senior, J. K. *The groups of order* 2^n; Macmillan: New York, 1964.
(19) Ceulemans, A.; Chibotaru, L. F.; Bovin, S. A. The electronic structure of polyhex carbon tori. *J. Chem. Phys.* **2000**, *112*(9), 4271−4278.

CI049865C