Journal for Geometry and Graphics
Volume 19 (2015), No. 2, 227–236.

Random Realization of Polyhedral Graphs as Deltahedra

Naoya Tsuruta[1], Jun Mitani[2], Yoshihiro Kanamori[2], Yukio Fukui[2]

[1]*School of Media Science, Tokyo University of Technology*
1404-1 Katakuramachi, Hachioji, Tokyo 192-0982, Japan
email: tsurutany@stf.teu.ac.jp

[2]*Graduate School of Systems and Information Engineering, University of Tsukuba*
email: {mitani,kanamori,fukui}@cs.tsukuba.ac.jp

Abstract. In this paper, we propose a method for realizing a polyhedral graph as a deltahedron, i.e., a polyhedron with congruent equilateral triangles as faces. Our experimental result shows that there are graphs that are not realizable as deltahedra. We provide an example of non-realizable graphs which are obtained by trying to construct deltahedra from each of the simple cubic polyhedral graphs with up to 10 vertices. We also show that the infinite families of non-realizable graphs can be obtained by solving the graph isomorphism problem.

Key Words: deltahedron, polyhedral graph, geometric realization

MSC 2010: 51M20, 05C10, 68R10, 52B05

1. Introduction

A *deltahedron* is a polyhedron whose faces are congruent equilateral triangles. Only eight of these are convex: those having 4, 6, 8, 10, 12, 14, 16, or 20 faces [5]. Coplanar faces sharing an edge are not allowed. The tetrahedron, octahedron, and icosahedron are the three deltahedra that are regular solids. If we permit non-convex shapes, then the number of deltahedra is infinite, because we can compose larger deltahedra by gluing two smaller deltahedra.

There are several subclasses of deltahedra. Cundy listed 17 *biform* deltahedra, which have only two forms of vertices [4]. Olshevsky added another 11 biform deltahedra to Cundy's list [9]. These lists did not permit intersecting faces, so these biform deltahedra are solids. Shephard presented 34 *isohedral* deltahedra, and McNeill added six further examples to Shephard's list [8, 13]. Isohedral deltahedra are face-transitive and may include intersecting faces. Trigg defined *spiral* deltahedra as those constructed from strips of equilateral triangles [14].

Each of these classes of deltahedra have their own particular properties. Therefore, the configurations of the vertices are very limited. We decided to loosen the conditions and see

ISSN 1433-8157/$ 2.50

what kinds of shapes are possible in the world of deltahedra. In this study, we tried to construct deltahedra from each of the simple polyhedral graphs and counted the graphs that can be constructed as deltahedra. We call these "deltahedral graphs". It is hard to determine whether a graph will form a deltahedron by examining only its combinatorial structure. Thus, we solve a geometric realization problem, which is the problem of determining whether a triangulation of an orientable surface can be realized geometrically in $\mathbb{R}^3$ as a polyhedron without self-intersections [6].

We propose a random realization method for constructing a deltahedron from a polyhedral graph and provide examples of the constructed deltahedra with up to 10 vertices. In our realization process, we generate an initial polyhedron with non-equilateral triangles and then deform the faces into equilateral triangles by a gradient method, because the graph does not provide the locations of the vertices. Note that the resulting deltahedron has a small geometric error and is not theoretically exact. Our realization process does not necessarily guarantee the non-realizability of a graph. We also propose an approach to determine the deltahedral non-realizability of a graph. The idea is to detect the operation used to construct the graph. Augmentation is an operation that joins each appendage polyhedron to its own single-core face. A polyhedron composed by adding a non-realizable structure is also non-realizable. We show that the augmentation can be detected by solving a graph isomorphism problem, and this detection is useful for finding an infinite family of graphs that are non-realizable as deltahedra.

Our deltahedral realization problem is a particular case of a geometric realization problem. In general, Bokowski and Guedes de Oliveira [1] showed that there is a non-realizable triangulation of the orientable surface of genus 6, and Schewe [12] showed that we can construct non-realizable triangulations for any number of vertices with genus 5 or 6. However, for surfaces of genus $1 \leq g \leq 4$, the problem remains open. The conditions for deltahedral realization are stricter than those. Each face must be realized as an equilateral triangle, and it is necessary to calculate the geometric coordinates to check whether there exist self-intersections or edges whose dihedral angle is equal to 180°. In this paper, we focus on surfaces of genus 0. Also the previous studies on deltahedra mentioned above focused on the genus-0 surface. Although a few deltahedra with $g > 0$ are known, we are not aware of published studies.

2. Deltahedral graphs

Polyhedral graphs are three-connected planar graphs. These graphs contain not only triangular faces, but also polygonal faces with more than three edges. A *cubic* polyhedral graph is a three-connected cubic planar graph which has only triangular faces. This graph is realized as a polyhedron whose faces are triangles, that is, a simple polyhedron. Deltahedra are a subclass of simple polyhedra because they are composed of equilateral triangles. Therefore, the graphs of deltahedra are a subclass of the graphs of simple polyhedra. The relation between them is shown in Figure 1.

Here we define a *deltahedral* graph as a graph which can be realized as a deltahedron. Although there are various kinds of deltahedra, we include only deltahedra without self-intersections and without edges for which the dihedral angle is 180°. For example, all the polyhedra in Figure 2 are composed of congruent equilateral triangles. However, the lower left one $(9_{24}, N)$ has intersecting faces, and the lower right one $(8_{12}, N)$ has coplanar and connecting faces. In this paper, we will not consider deltahedra like these lower ones but only

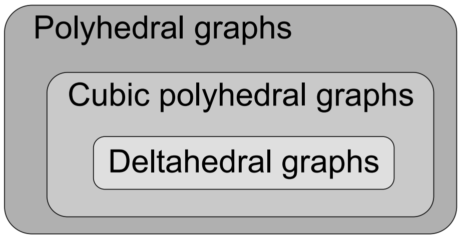


Figure 1: Relation between polyhedral graphs and deltahedral graphs

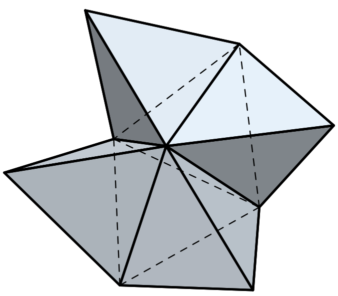
those like the topmost example $(9_{17}, D)$ in Figure 2. The code below each figure is composed of two numbers and a character. The numbers represent the number of vertices and the index of the graph. The index follows the order of an existing graph generation algorithm [3], and they are classified into one of two categories: 'D' for deltahedral graphs, and 'N' for non-deltahedral graphs. For example, $(6_1, D)$ is the first six-vertex deltahedral graph that is generated by that algorithm. The important thing is that more than one deltahedron may be obtained from a single graph. Figure 3 shows a deltahedral graph that has one convex form and two non-convex forms. If a graph has at least one deltahedron, we say it is *deltahedral*.

3. Approach

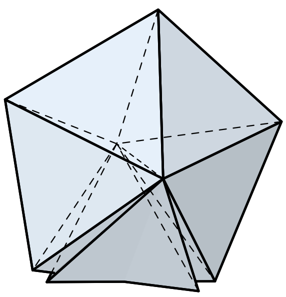
We have to prepare the graphs which are combinatorially different before the realization process. The class of deltahedra is a subset of the class of simple polyhedra whose faces are triangles. Therefore, the number of three-connected cubic polyhedral graphs is an upper bound on the number of deltahedral graphs. Graph enumeration has been widely discussed, and there are many approaches to it. We used the planar graph generation program plantri [3] to obtain the three-connected cubic planar graphs.

We used two steps to realize the graphs. First, each graph was embedded without intersections in the 2D plane with straight line edges. Then, we used graph lifting [11] and an iterative deformation process for the attempt to construct a deltahedron from the graph. The constructed polyhedron may have coplanar neighboring faces or may not even be a solid. The graph was considered to be a deltahedral graph only if the constructed geometry satisfied the conditions for a deltahedron.

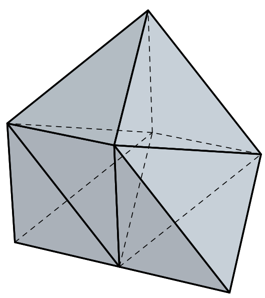
As mentioned above, non-convex deltahedra form an infinite class. So it is necessary to



(a) $(9_{17}, D)$



(b) $(9_{24}, N)$ has intersecting faces



(b) $(8_{12}, N)$ has coplanar and connecting faces

Figure 2: Polyhedra with equilateral triangles. The first polyhedron is a deltahedron and the others are not.

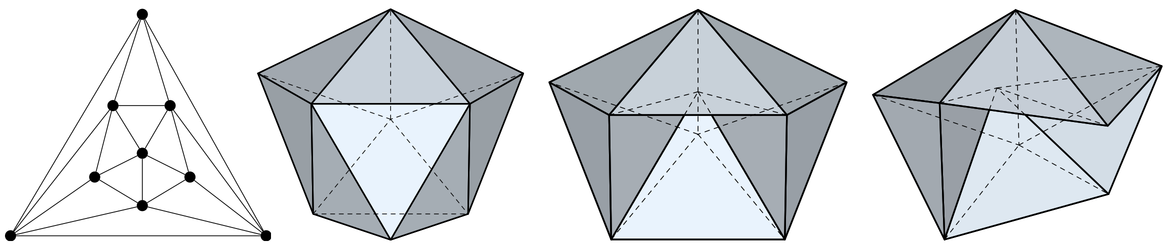


Figure 3: One convex shape and two non-convex shapes form the graph 9_{50}

limit the allowable number of vertices. We choose this limit to be ten. As an upper bound, there are 306 polyhedral graphs which have ten or less vertices. We decided that it is enough for the first step of this realization problem.

3.1. Graph embedding

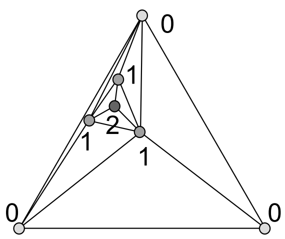
Several methods have been proposed for embedding planar graphs. We used PLESTENJAK's algorithm, which is based on a spring model [10]. The size of the graph is small enough that it is practical to calculate it. This algorithm chooses a base face and fixes the positions of its vertices in the 2D plane. The remaining vertices are placed inside the base face. We choose the base face randomly and place it so that it forms an equilateral triangle with edges of unit length. The algorithm calculates the *periphericity* p_v of each vertex when placing the inner vertices. Periphericity is a kind of centrality and indicates the distance from the outer polygon. The periphericity p_v of the outer triangle is 0, and the p_v of the vertices adjacent to them is 1. The periphericity p_v increases as going toward inside.

These periphericities are used when generating the initial polyhedron which is needed for the iterative deformation. Figure 4a shows an example of an embedded graph. In this step, the base face is chosen randomly. The results of the following steps are different, depending on the initial choice.

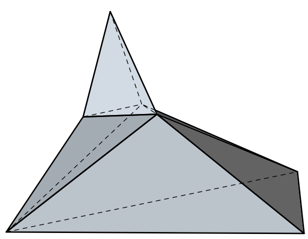
3.2. Realization of deltahedra

First, we generate a polyhedron with non-equilateral triangles from an embedded graph. Although this polyhedron will be deformed to a deltahedron, it should be close to a deltahedron. We obtain the heights h_v for the vertices corresponding to each p_v by using the following formula:

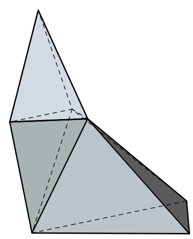
$$h_v = Cp_v,$$



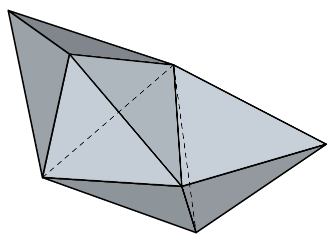
(a) Embedding



(b) Lifting



(c) Shrinking the base face



(d) Deformed shape

Figure 4: Realization process

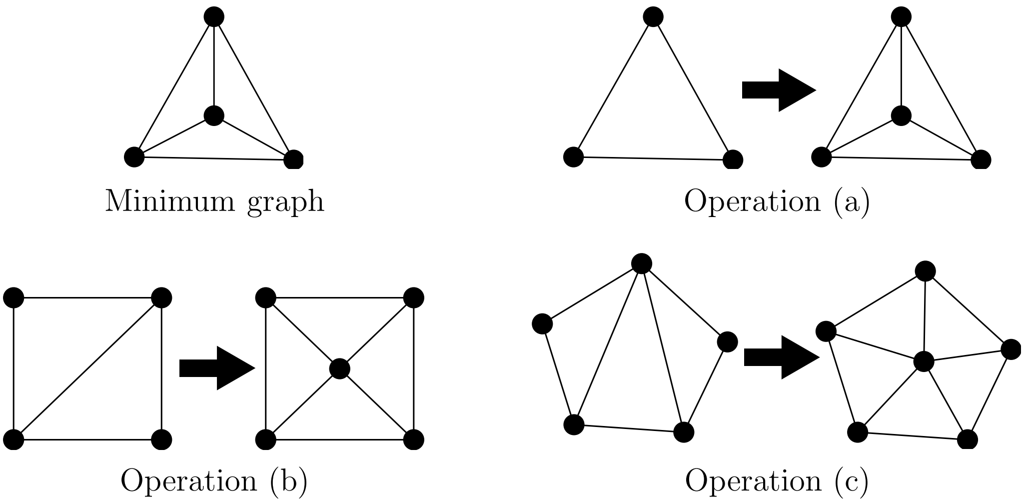


Figure 5: Operations for the generation of simple triangulations

where C is a constant value that defines the height of the pyramid-like model shown in Figure 4b. Then the base face is shrunk to reduce the differences in the lengths of the edges (Figure 4c).

Finally, we transform the generated polyhedron to form a deltahedron by using a numerical optimization. We define a penalty function F which measures the difference in length of the edges, and then we minimize it. We set

$$F := \sum_i (L(E_i) - 1.0)^2,$$

where $L(E_i)$ is the length of E_i. We use the LEVENBERG-MARQUARDT algorithm for the iterations and the GAUSS-SEIDEL method for solving the linear equations within each iteration. The resulting polyhedron may include intersecting faces. If this happens we try a different embedding or we reconfigure the positions of the vertices manually.

The convergence of this method is not guaranteed. However, there exists a vertex configuration that makes the lengths of all edges equal when self-intersections are ignored. The proof is as follows:

There are only three operations for generating all triangulations [2],

a) adding a vertex of degree 3,
b) removing an edge and adding a vertex of degree 4, and
c) removing two edges and adding a vertex of degree 5.

The minimum four-vertex graph is realized as a tetrahedron. The graph and the operations are shown in Figure 5. When these operations are applied, the resulting polyhedron can be fit into the interior of the tetrahedron because our iterative deformation algorithm does not care about intersecting faces. The operation (a) adds a vertex that separates a face into three faces. The additional faces can be realized as an excavation of a tetrahedron. The additional faces generated by operations (b) and (c) will be realized by overlapping coplanar faces covering other faces. These operations change the positions of the original vertices, however, the shape will still fit into the tetrahedron.

Table 1: Number of deltahedral graphs

Vertices	4	5	6	7	8	9	10
Graphs	1	1	2	5	14	50	233
Deltahedral graphs	1	1	2	4	9	36	154
Non-deltahedral graphs	0	0	0	1	5	14	79

4. Results

Here we present the experimental results of the realizations with up to 10 vertices. Table 1 shows the numbers of deltahedral and non-deltahedral graphs. We can see that more than 50% of the graphs are deltahedral. The percentage of non-deltahedral graphs is gradually decreasing. To confirm this trend, it will be necessary to investigate larger graphs. Figures 6 and 7 show the constructed polyhedra with seven and eight vertices, respectively. Note that each figure shows one of various possible polyhedra. We did not generate all the possible realization shapes for each graph. Fortunately, if we do not allow faces to intersect, the variations are small in graphs with ten or fewer vertices. It is easy to identify manually whether a graph has different shapes. For example, only 8_{13} admits different shapes (made by tucking one of the pyramid) in Figure 7. In difficult cases, we manually generated *good* initial polyhedra and performed an iterative deformation.

The lengths of all the edges of the constructed polyhedra are very close to 1. Although there is an error caused by the numerical calculations, the maximum difference between the mean edge length and each edge was under 10^{-5}. In most cases, the iteration process converged in a few seconds on a PC with 2.9 GHz Intel Core i7 CPU. The time differed depending on the initial polyhedron produced by the graph lifting.

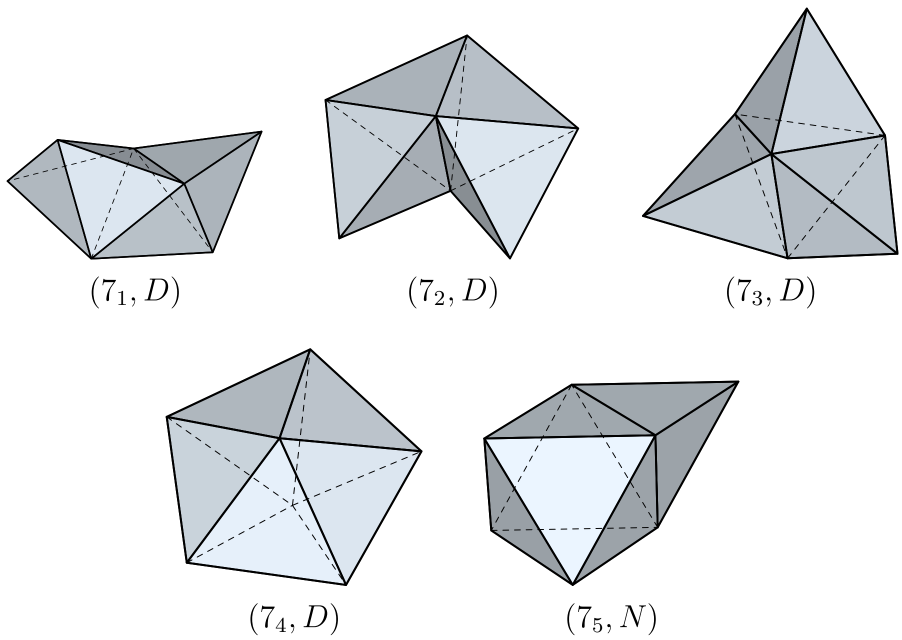
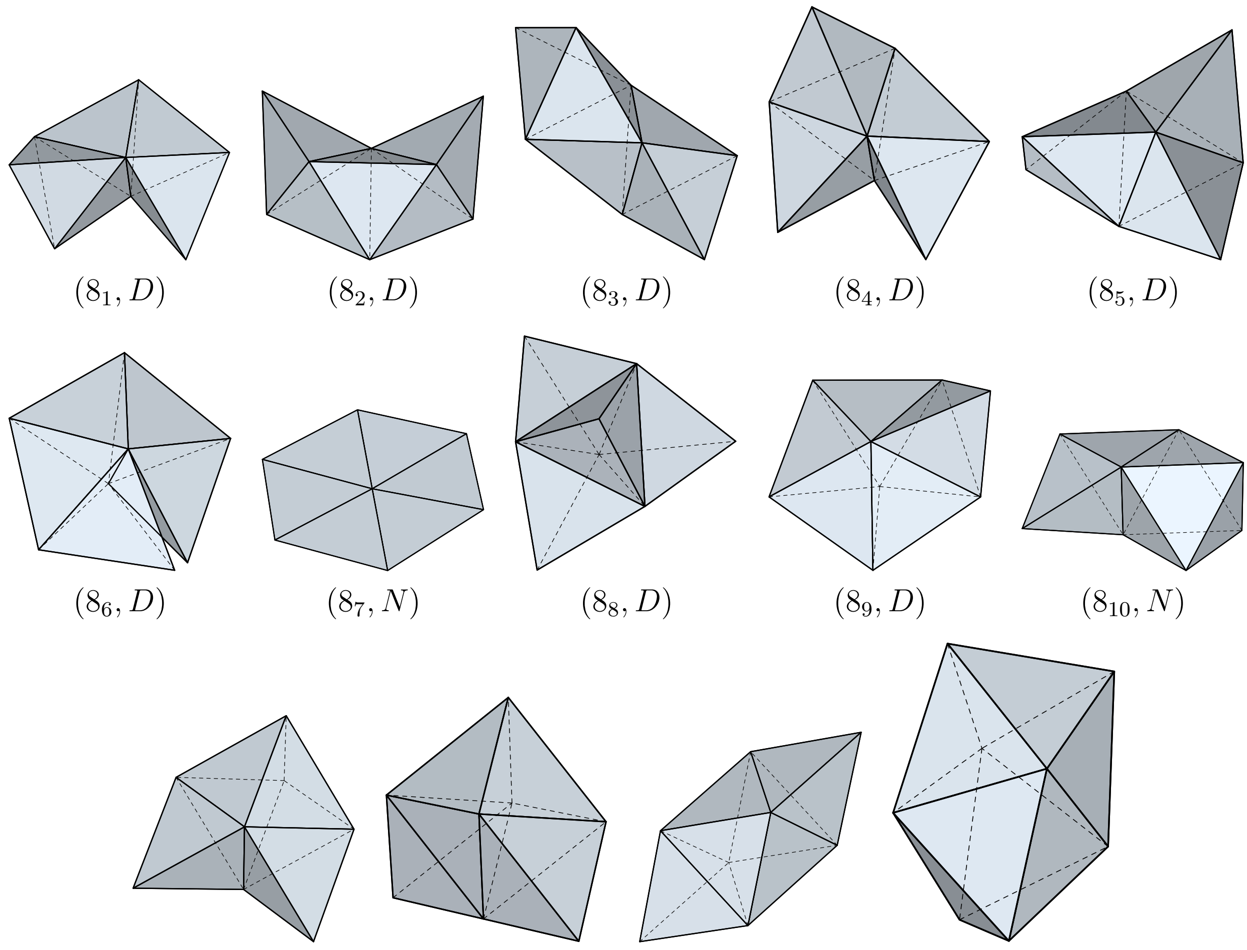


Figure 6: Constructed polyhedra with $V = 7$

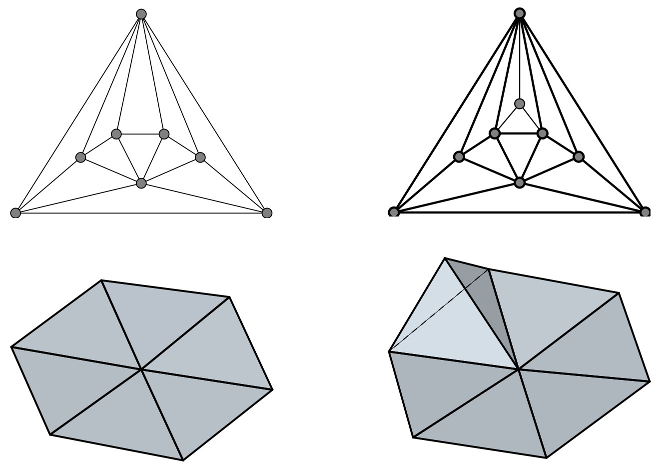


$(8_1, D)$ $(8_2, D)$ $(8_3, D)$ $(8_4, D)$ $(8_5, D)$

$(8_6, D)$ $(8_7, N)$ $(8_8, D)$ $(8_9, D)$ $(8_{10}, N)$

$(8_{11}, N)$ $(8_{12}, N)$ $(8_{13}, N)$ $(8_{14}, D)$

Figure 7: Constructed polyhedra with $V = 8$



$(8_7, N)$ $(9_{26}, N)$

Figure 8: Graphs with and without the appendage tetrahedra, and the associated polyhedra

5. Finding the infinite family of non-deltahedral graphs

Some graph structures cause face intersections. For example, in the case of the graph 8_7 the shape is completely flat (see Figure 7). Similar shapes appear also in larger graphs. We can find the graphs that have particular non-realizable structures by comparing graphs.

Figure 8 shows graphs which contain the same partial structures and their realized polyhedra. As shown in Figure 8, when we form a larger deltahedron by connecting two smaller deltahedra along a single face, the original shapes do not change and the graph of the appended deltahedron can be embedded inside a connecting face. This operation can be detected by solving the subgraph isomorphism problem. Hence, we can obtain a family of non-deltahedral graphs from one non-deltahedral graph by solving it, without the need to realize the polyhedron.

Figure 10 shows an example of a non-deltahedral family. We used graph 8_7 as a seed and obtained ten non-deltahedral graphs with nine or ten vertices. We used a simple backtracking algorithm for the subgraph isomorphism [15]. For nine and ten vertices the computation time was 140 ms and 2500 ms, respectively.

This subgraph isomorphism only detects the connection of two deltahedra along a single face. Figure 9 shows a comparison of a connection along one face with another, where more faces are involved. These shapes look similar, but the isomorphism of the subgraphs only detects the deformation from the left to the central object. The right one is a seed of another non-deltahedral family. For the transition from the center to the right, an octahedron is attached along two faces. In this case, the original shape does not change, but in general, attaching a deltahedron along at least two faces causes a deformation of the original shape.

The graph 7_5 can be realized as a polyhedron that does not contain self-intersections. Such non-deltahedral graphs that are realizable as polyhedra cannot be used as seeds of the subgraph isomorphism problem. A larger graph may be realizable as a deltahedron because the attachment changes the dihedral angles of the edges around the core face.

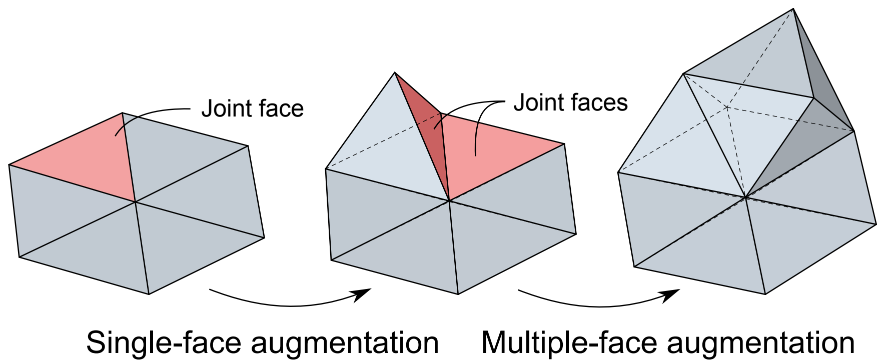


Figure 9: Connection with single faces and with multiple faces

6. Conclusions

We described a method for realizing a simple cubic polyhedral graph as a deltahedron. Not all simple cubic polyhedral graphs can be realized as deltahedra, due to self-intersections or dihedral angles of 180°. We also showed that the infinite family of non-deltahedral graphs is obtained by solving the subgraph isomorphism problem. This eliminates the non-deltahedral graphs from the set of cubic polyhedral graphs without the need for realization, and it may

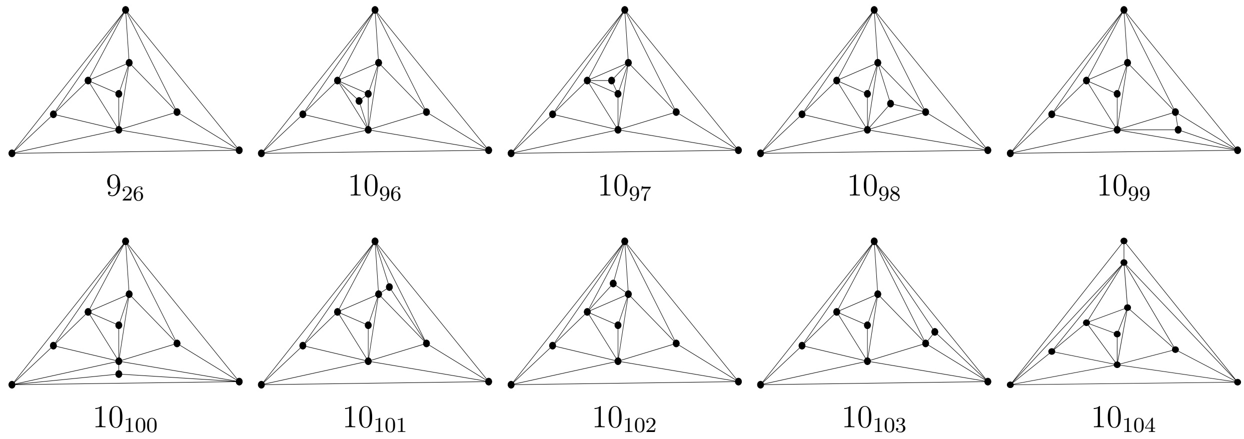


Figure 10: Non-deltahedral family of graphs 8_7 with up to ten vertices

be useful for finding the deltahedral ones. Our future work is to improve the discrimination approach by employing this detection method.

The remaining problem is to determine how non-realizability can be characterized. In order to do this, we need the vertex coordinates for checking the dihedral angles. Our deltahedral realization problem is similar to the polyhedral realization problem. We hope that by combining our iterative deformation process with other realization or detection methods [12, 7], we will obtain a method that creates a robust realization of deltahedra.

It is also necessary to investigate graphs of higher genus. Do all triangulated surfaces with non-zero genus admit a deltahedral realization? Our realization process is not applicable for surfaces with non-zero genus; however, an iterative deformation may be useful. It will be an interesting challenge to find the smallest deltahedron with $g > 0$, such as a toroidal deltahedron.

References

[1] J. Bokowski, A. Guedes de Oliveira: *On the Generation of Oriented Matroids.* Discrete Comput. Geom. **24**/2-3, 197-208 (2000).

[2] R. Bowen, St. Fisk: *Generation of Triangulations of the Sphere.* Math. Comp. **21**/98, 250–252 (1967).

[3] G. Brinkmann, B.D. Mckay: *Fast generation of planar graphs.* MATCH-Commun. Math. Co. **58**, 323–357 (2007).

[4] H.M. Cundy: *Deltahedra.* Math. Gaz. **36**, 263–266 (1952).

[5] H. Freudenthal, B.L. van der Waerden: *On an Assertion of Euclid.* Simon Stevin **25**, 115–121 (1947).

[6] B. Grünbaum: *Convex Polytopes.* 2nd ed., Graduate Texts in Mathematics, vol. 221, Springer, New York 2003.

[7] F.H. Lutz: *Enumeration and Random Realization of Triangulated Surfaces.* In A.I. Bobenko, J.M. Sullivan, P. Schröder, G.M. Ziegler (eds.): Discrete Differential Geometry, Oberwolfach Seminars, vol. 38, Birkhäuser, Basel 2008, pp. 235–253.

[8] J. McNeill: *Isohedral Deltahedra.* `http://www.orchidpalms.com/polyhedra/rhombic/isohedral\_deltahedra.htm`.

[9] G. Olshevsky: *Breaking Cundy's Deltahedra Record.* `http://www.interocitors.com/polyhedra/papers/Polytopics28-BreakingCundy'sDeltahedraRecord.pdf`.

[10] B. Plestenjak: *An Algorithm for Drawing Planar Graphs.* Software Pract. Exper. **29**/11, 973–984 (1999).

[11] G. Rote: *Realizing Planar Graphs As Convex Polytopes.* Proceedings of the 19th International Conference on Graph Drawing GD'11, Springer-Verlag, Berlin, Heidelberg 2012, pp. 238–241.

[12] L. Schewe: *Nonrealizable Minimal Vertex Triangulations of Surfaces: Showing Nonrealizability Using Oriented Matroids and Satisfiability Solvers.* Discrete Comput. Geom. **43**/2, 289–302 (2010).

[13] G.C. Shephard: *Isohedral Deltahedra.* Period. Math. Hungar. **39**/1-3, 83–106 (2000).

[14] C.W. Trigg: *An Infinite Class of Deltahedra.* Math. Mag. **51**/1, 55–57 (1978).

[15] J.R. Ullmann: *An Algorithm for Subgraph Isomorphism.* J. ACM **23**/1, 31–42 (1976).

Received August 8, 2014; final form November 11, 2015