Journal for Geometry and Graphics
Volume 30 (2026), No. 1, 1–13

Polyhedral Realization as Deltahedra Using the Subgraph Isomorphism Test

Naoya Tsuruta

Utsunomiya University, Utsunomiya, Japan
naoya@is.utsunomiya-u.ac.jp

Abstract. This paper extends our previous work presented at the 21st International Conference on Geometry and Graphics on realizing 3-connected triangulations as deltahedra using subgraph isomorphism testing. The method classifies graphs into composite deltahedra, formed by joining two deltahedra, and non-composite deltahedra, corresponding to 4-connected triangulations. Composite deltahedra are realized analytically by augmenting known components, while non-composite deltahedra require numerical optimization. We introduce algorithmic optimizations based on a key geometric property: degree-3 vertices in deltahedra necessarily form tetrahedra with their neighbors. This property occurs in over 97 percent of graphs with 10 or more vertices, enabling targeted search strategies. Our optimized algorithm achieves approximately 2 times speedup across all graph sizes through case-based search, graph isomorphism testing, and histogram filtering. We extend the experimental scope to 12 vertices, encompassing 7,595 graphs, and provide comprehensive performance evaluation. Examples of realized forms are presented, with particular focus on non-composite deltahedra.

Key Words: deltahedra, polyhedral realization, subgraph isomorphism, triangulated graphs

MSC 2020: 51M04 (primary), 68R10

1 Introduction

Polyhedral realization is the reconstruction of a polyhedron from a given graph structure. Studies have explored whether reconstruction is possible [4], and developed reconstruction methods focusing on symmetry, the shape of polygonal faces, and vertex coordinates [1, 3, 8]. Calculating the coordinates of each unknown vertex is not an easy task.

We have been working on realizing a graph with equilateral triangle faces, that is the reconstruction of a deltahedron from a graph. A deltahedron is a polyhedron whose faces are congruent equilateral triangles. There are only eight convex ones, but various families

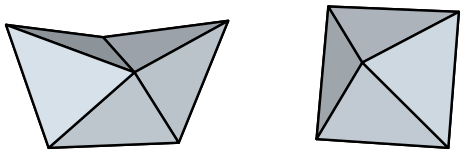


Figure 1: Examples of composite deltahedra (tetrahedron on triangular dipyramid) and noncomposite deltahedra (octahedron) with six vertices.

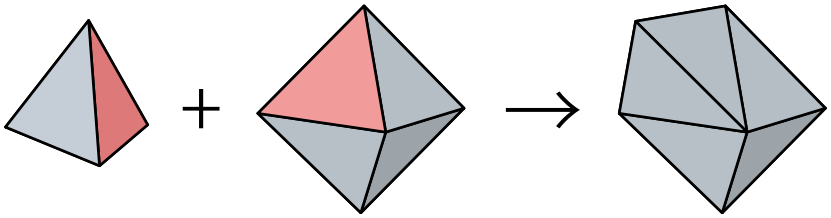


Figure 2: Generation of composite deltahedra.

of non-convex deltahedra and equilateral triangle surfaces have been investigated. We have provided reconstructed forms of simple planar graphs with less than 10 vertices [11].

Previous studies have utilized numerical optimization methods to construct non-convex deltahedra [5, 7, 12]. However, due to the possibility of small errors in the final shape of the reconstructed polyhedron, it is preferable to construct it analytically if possible. When counting simple planar graphs, many graphs contain smaller graphs as subgraphs. This implies that most of the deltahedron to be reconstructed is composed of multiple smaller deltahedra, and if the vertex coordinates of each smaller deltahedron are known, obtaining the coordinates of the larger deltahedron becomes straightforward.

In our previous work [9], we proposed an approach to reconstruct deltahedra from triangulated graphs using subgraph isomorphism testing. That baseline method exhaustively searches for decompositions by testing all possible component sizes. While successful in realizing composite deltahedra up to 11 vertices, the computational efficiency relied on naive enumeration.

In this paper, we extend that approach by exploiting structural properties of triangulated graphs to achieve substantial algorithmic improvements. Specifically, we identify that degree-3 vertices in deltahedra necessarily form tetrahedra with their neighbors—a geometric constraint that occurs in over 97% of graphs with 10 or more vertices. Based on this observation, we develop an optimized realization algorithm that achieves approximately 2× speedup through targeted search strategies. We also extend the experimental scope to 12 vertices, encompassing 7,595 graphs, and provide comprehensive performance evaluation. Several reconstructed examples are presented, with particular focus on non-composite deltahedra corresponding to 4-connected triangulated graphs.

2 Composite and Noncomposite Deltahedra

Here, composite and noncomposite deltahedra are defined. A composite deltahedron comprises two deltahedra that are joined so that one of their faces aligns with the other. Joining them ensures that each polyhedron is not deformed but maintains its original shape. Further, the faces concealed on the inside by the joint are removed. Figure 2 shows an example of a composite deltahedron. Put differently, a composite deltahedron can be decomposed into two

Table 1: Numbers of the 3- and four (4)-connected triangulated graphs and the difference.

# of vertices	3-connected	4-connected	Difference
4	1	-	-
5	1	0	1
6	2	1	1
7	5	1	4
8	14	2	12
9	50	4	36
10	233	10	223
11	1249	25	1224
12	7595	87	7508

or more smaller deltahedra by cutting at the single face, whereas noncomposite deltahedra cannot be decomposed.

As all the faces of a deltahedron are equilateral triangles, the upper bound of the number of possible deltahedra is limited to the number of three-connected (3-connected) triangulated graphs. However, not all 3-connected graphs can form a deltahedron, and this may result in intersecting faces, duplicated vertices or zero-volume.

Thus, by excluding the graphs containing smaller triangulated graphs that are subgraphs (isomorphic to themselves) from the set of 3-connected graphs, we are left with graphs that can form noncomposite deltahedra. This process is equivalent to increasing the graph connectivity by 1. Therefore, the number of remaining graphs is equal to the number of 4-connected graphs (Table 1). For example, Figure 1 shows two 3-connected triangulated graphs with six vertices and one 4-connected triangulated graph, corresponding to composite and noncomposite deltahedra, respectively. The sequence in the difference column represents the number of possible composite deltahedra (excluding tetrahedron). The number of 3-connected triangulated graphs is significantly larger than that of the 4-connected triangulated graphs, and this is mostly because a tetrahedron can be augmented in numerous ways to form composite deltahedra. Conversely, 4-connected graphs may form interesting shapes that cannot be obtained from simple joining operations.

2.1 Analysis of Degree-3 Vertex Distribution

Before describing our realization algorithm, we analyze a structural property of 3-connected triangulated graphs that provides opportunities for algorithmic optimization. This analysis extends our previous work [9] by identifying and exploiting geometric constraints inherent to deltahedra.

2.1.1 Geometric Property of Degree-3 Vertices

In a deltahedron, every face is a congruent equilateral triangle. A vertex v of degree 3 is incident to exactly three such faces, which share v as a common vertex. Let v_1, v_2, and v_3 denote the three neighbors of v. Since each face is an equilateral triangle and all edges have unit length, the four vertices $\{v, v_1, v_2, v_3\}$ necessarily form a regular tetrahedron.

This geometric constraint has an important implication for composite deltahedra. When a composite deltahedron G contains a degree-3 vertex v, one of its components must be the

Table 2: Prevalence of degree-3 vertices in 3-connected triangulated graphs

# of vertices	Total graphs	Graphs with degree-3	Percentage
4	1	1	100.00%
5	1	1	100.00%
6	2	1	50.00%
7	5	4	80.00%
8	14	12	85.71%
9	50	45	90.00%
10	233	221	94.85%
11	1,249	1,215	97.28%
12	7,595	7,465	98.29%

tetrahedron formed by $\{v, v_1, v_2, v_3\}$. Consequently, removing v and its three incident edges from the graph of G yields a subgraph with exactly $n-1$ vertices, which corresponds to the graph structure of the other component.

This observation suggests a targeted search strategy: rather than testing subgraph isomorphism for all possible component sizes from 4 to $n-1$ vertices, we can directly search for the $(n-1)$-vertex component when a degree-3 vertex exists.

2.1.2 Statistical Distribution

To assess the practical impact of this property, we analyzed the distribution of degree-3 vertices across all 3-connected triangulated graphs generated by plantri [2]. Table 2 presents the results for graphs with 4 to 12 vertices.

The data reveals several notable patterns. First, the proportion of graphs containing at least one degree-3 vertex increases monotonically with the number of vertices, reaching 98.29% for 12-vertex graphs. For graphs with 10 or more vertices, over 94% contain at least one degree-3 vertex. This high prevalence suggests that an algorithm designed to efficiently handle the degree-3 case would apply to the vast majority of graphs in practical applications.

2.1.3 Algorithmic Implications

The geometric and statistical properties of degree-3 vertices lead to three key algorithmic optimizations:

1. **Reduced search space:** For graphs with degree-3 vertices ($> 97\%$ of cases when $n \geq 10$), we need only search for components with exactly $n-1$ vertices, rather than testing all sizes from 4 to $n-1$.
2. **Faster isomorphism testing:** Since removing a degree-3 vertex and its edges yields a subgraph with the exact size of the target component, we can use graph isomorphism testing rather than subgraph isomorphism testing. Graph isomorphism is computationally less expensive as it does not require exploring different vertex mappings to find a subset match.
3. **Candidate pruning:** The degree sequence of the $(n-1)$-vertex subgraph provides additional filtering constraints. We can compute a degree histogram and test isomorphism only against graphs with matching histograms, substantially reducing the candidate set.

These observations form the foundation for the optimized realization algorithm presented in Section 3.

3 Realization Method

Our realization method realizes a graph as a deltahedron. The input graph is generated using plantri [2]. We also gather 3D coordinate data for known convex deltahedra, which will be referenced during the realization process. All edges of the deltahedra are normalized to unit length.

Given an input graph G with n vertices, the algorithm searches for a graph G_1 that is subgraph isomorphic to G from the set of graphs with $n-1$ or fewer vertices. If such a subgraph isomorphic graph G_1 exists, the deltahedron P_1 corresponding to G_1 is one of the two components. To find the other component P_2, the algorithm seeks graph G_2 that is isomorphic to the remaining part of the graph. Finally, after determining the joining face through vertex and edge mappings, the two components are augmented at the face to construct a composite deltahedron. If no subgraph exists, the algorithm employs numerical optimization to realize a non-composite deltahedron. Algorithm 1 presents this baseline approach from our previous work [9].

After identifying components P_1 and P_2, the algorithm verifies that their augmentation reproduces the original graph G. Let $\{p_1, p_2, p_3\}$ and $\{q_1, q_2, q_3\}$ denote the vertices of the joining faces in G_1 and G_2, respectively. The algorithm tests all three cyclic alignments of these vertices by merging the graphs and checking isomorphism with G. For symmetric components such as tetrahedra and octahedra, all alignments yield identical structures, while non-symmetric components require explicit verification. This verification step refines the baseline approach from our conference publication [9], which implicitly assumed correct alignment. While this assumption holds for highly symmetric components that dominate the graph distribution, explicit verification ensures correctness for all component types.

Assuming either G_1 or G_2 is a graph that cannot be realized as a deltahedron (we call this a *nondeltahedral graph*), then G also cannot be realized. Therefore, the computation can be omitted. Nondeltahedral graphs also appear in larger graphs, generating nonrealizable families, as the self-intersection is not resolved by the joining operations. Notably, a realized polyhedron with coplanar faces is not a deltahedron within the definition; however, if it appears as a subgraph of a larger graph, the coplanar faces may be eliminated by joining another deltahedron.

3.1 Optimized Method

Building upon the degree-3 vertex analysis from Section 2.1, we now present an optimized algorithm that exploits the geometric properties of triangulated graphs. The key innovation is a case-based approach that applies different search strategies depending on the presence of degree-3 vertices.

Algorithm 2 distinguishes two cases:

Case 1: Graphs with degree-3 vertices (over 97% for $n \geq 10$). When a degree-3 vertex v exists, removing it yields a subgraph with exactly $n-1$ vertices. This enables three optimizations: (1) searching only $(n-1)$-vertex graphs rather than all sizes from 4 to $n-1$, (2) using graph isomorphism instead of subgraph isomorphism, and (3) filtering candidates by degree histogram before isomorphism testing.

Algorithm 1 Realization process

```
function REALIZATION(G)
    n ← number of vertices in G
    for each G_1 with n − 1 to 4 vertices do
        if G_1 is subgraph isomorphic to G then
            P_1 ← deltahedron whose structure is G_1
            G_s ← Polyhedral graph of remaining part
        end if
    end for
    m ← number of vertices in G_s
    for each G_2 with m vertices do
        if G_2 is subgraph isomorphic to G_s then
            P_2 ← deltahedron whose structure is G_2
            Test cyclic alignments of joining face
            if Valid alignment found matching G then
                P ← Augmentation of P_1 and P_2
                return P
            end if
        end if
    end for
    Apply numerical optimization                    ▷ Non-composite case
    return P
end function
```

Case 2: Graphs without degree-3 vertices (less than 3% for $n \geq 10$). For these graphs, the algorithm searches from $(n-2)$ down to 6 vertices using subgraph isomorphism for the first component G_1. Importantly, once G_1 is found, the size and structure of the remaining subgraph G_s are uniquely determined. This allows the second component G_2 to be identified using graph isomorphism with histogram filtering, rather than the exhaustive subgraph isomorphism search employed in Algorithm 1. Thus, even for this minority of graphs, the optimized method improves upon the baseline approach. As with Algorithm 1, the joining face alignment is verified by testing all three cyclic permutations, which is particularly important for non-symmetric components in this case.

The distinction between these cases enables targeted optimization strategies that exploit the specific structural properties of each graph type, leading to substantial performance improvements as demonstrated in Section 4.

4 Results

We have applied our method to graphs with up to 12 vertices, and realized all the composite deltahedra. Most of them comprised combinations of tetrahedra and octahedra (Figure 3). The code under each figure is composed of two numbers and a character. The numbers denote the number of vertices and the graph index. The index follows the sequence generated by Plantri [2] and is categorized into one of the following three groups: "D" for deltahedra, "P" for those with coplanar faces, and "N" for the nonrealizable ones with self-intersections.

Table 3 presents the number of generated deltahedra and polyhedra containing self-intersecting faces. Classifications are based on numerical computation, except for deltahedra

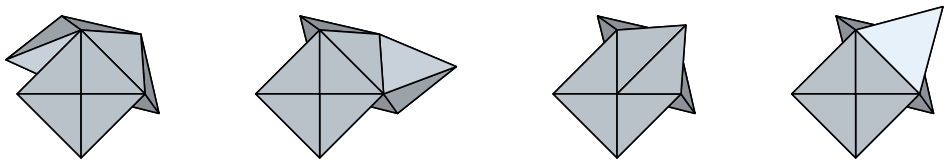
Algorithm 2 Optimized Realization Process

```
function RealizationOptimized(G)
    n ← number of vertices in G
    v3 ← vertex of degree 3 in G (or null if none exists)
    if v3 ≠ null then                          ▷ Case 1: Has degree-3 vertex (> 97% for n ≥ 10)
        G′ ← G with v3 and its incident edges removed
        H ← degree histogram of G′
        candidates ← graphs with (n − 1) vertices matching H
        for each G1 in candidates do
            if G′ is graph isomorphic to G1 then
                P1 ← deltahedron whose structure is G1
                f ← face corresponding to neighbors of v3
                P ← augment P1 with tetrahedron at face f
                return P
            end if
        end for
    else                                       ▷ Case 2: No degree-3 vertex (< 3% for n ≥ 10)
        for each G1 with (n − 2) down to 6 vertices do
            if G1 is subgraph isomorphic to G then
                P1 ← deltahedron whose structure is G1
                Gs ← polyhedral graph of remaining part
                m ← number of vertices in Gs
                Hs ← degree histogram of Gs
                candidates2 ← graphs with m vertices matching Hs
                for each G2 in candidates2 do
                    if Gs is graph isomorphic to G2 then
                        P2 ← deltahedron whose structure is G2
                        Test cyclic alignments of joining face
                        if Valid alignment found matching G then
                            P ← Augmentation of P1 and P2
                            return P
                        end if
                    end if
                end for
            end if
        end for
    end if
    Apply numerical optimization               ▷ Non-composite case
    return
end function
```

constructed exclusively from convex components. There may be other realizations without self-intersections for the remaining cases, as discussed in Section 4.2.

Here, we focused on the noncomposite deltahedra and discussed their geometric properties. Figure 4 shows examples of the realization of the 4-connected triangulated graphs. The codes contain the indexes in the sequence of the 3-connected graph. The figures were ordered alphabetically, following the sequence of the 4-connected triangulated graphs.

We assumed that Figure 4(b) was the smallest near-miss deltahedron. The errors in the edge lengths were up to 3%, and a further convergence was not possible unless the self-



(a) $11_{770}, P$ (b) $11_{771}, P$ (c) $11_{772}, P$ (d) $11_{773}, D$

Figure 3: Examples of the 11-vertex composite deltahedra.

Table 3: Number of self-intersecting polyhedra.

# of vertices	# of deltahedra	# of self-intersecting polyhedra[†]
4	1	0
5	1	0
6	2	0
7	5	0
8	14	1
9	50	3
10	233	19
11	1249	130
12	7595	1088

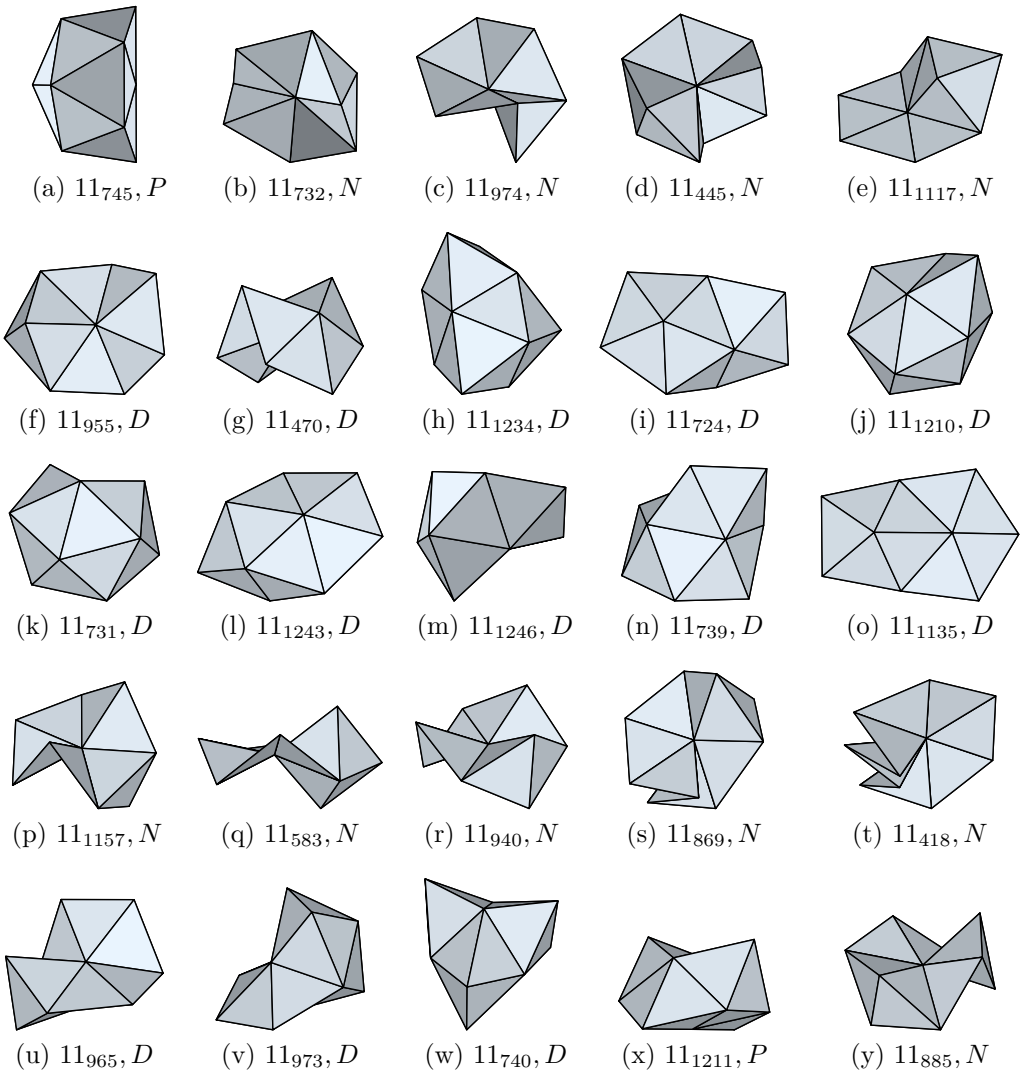
[†]Classifications are based on numerical computation, except for deltahedra constructed exclusively from convex components.

intersection of faces was allowed. We attempted to rigorously verify the non-realizability of this graph using Mathematica's `Resolve` function and `FindInstance` with coordinate hints derived from the numerical solution; however, both approaches were unsuccessful. As a reference for computational cost, `Resolve` required 0.079 seconds for the tetrahedron (graph 4_0) and resolved tetrahedral compounds up to 8 vertices within one second; however, the octahedron (graph 6_1) required 197 seconds, and graphs containing non-tetrahedral structures were aborted due to memory exhaustion (32 GB RAM). This suggests that computational cost depends not only on the number of vertices but also on the degree distribution of the graph. A rigorous proof of non-realizability for this near-miss case remains an open problem. The other nine polyhedra labeled N exhibited completely overlapping faces.

Notably, Figure 4(e) can be decomposed into a tetrahedron, an octahedron, and a flat shape with eight vertices. However, this one is not a composite deltahedron, as it requires the simultaneous separation of multiple faces during the decomposition. Although Figure 4(m) appears to comprise three tetrahedra and a triaugmented triangular prism, it is not. Simply joining the three tetrahedra onto a triaugmented triangular prism generates a deltahedron with 12 vertices (Figure 5(a)). Therefore, assuming the realization form can be decomposed, it cannot be distinguished based on the graph structure alone.

Figure 4(a) is called an edge-contracted icosahedron, as it exhibits the coplanar faces and icosahedron-like surface. Further, Figure 4(x) can be decomposed into two square pyramids and two triamond antiprisms (TAP 2,2) [6].

An observation of the other examples, reveals that unique shapes can be created by joining the three connecting tetrahedra in Figure 6. Figures 4(l) and (m) can be viewed as different approaches for augmenting this tetrahedral component at alternative locations on



(a) $11_{745}, P$ (b) $11_{732}, N$ (c) $11_{974}, N$ (d) $11_{445}, N$ (e) $11_{1117}, N$

(f) $11_{955}, D$ (g) $11_{470}, D$ (h) $11_{1234}, D$ (i) $11_{724}, D$ (j) $11_{1210}, D$

(k) $11_{731}, D$ (l) $11_{1243}, D$ (m) $11_{1246}, D$ (n) $11_{739}, D$ (o) $11_{1135}, D$

(p) $11_{1157}, N$ (q) $11_{583}, N$ (r) $11_{940}, N$ (s) $11_{869}, N$ (t) $11_{418}, N$

(u) $11_{965}, D$ (v) $11_{973}, D$ (w) $11_{740}, D$ (x) $11_{1211}, P$ (y) $11_{885}, N$

Figure 4: Realizations of the noncomposite deltahedra with 11 vertices. Labels are based on numerical computation.

(a) $12_{7558}, D$ (b) $11_{1246}, D$

Figure 5: Composite deltahedron and similar noncomposite deltahedron.

the triangular pyramid, indicating its inherent adaptability.

4.1 Limitations of the Variations of the Realized Form

Some deltahedra have multiple realization forms. For example, the composite deltahedron 10_{230} (Figure 7) comprises a tetrahedron and a triaugmented triangular prism, and five realiza-

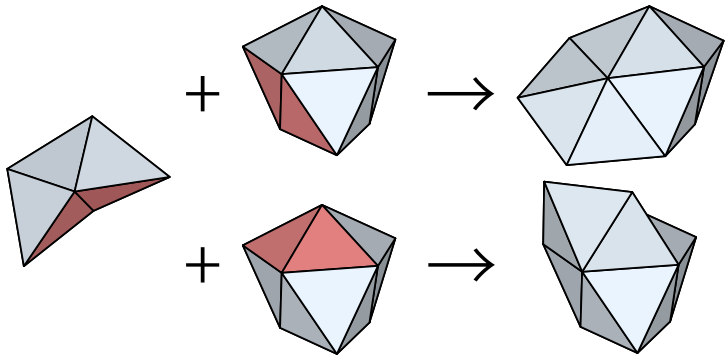


Figure 6: Joining the three tetrahedra components with deformation.

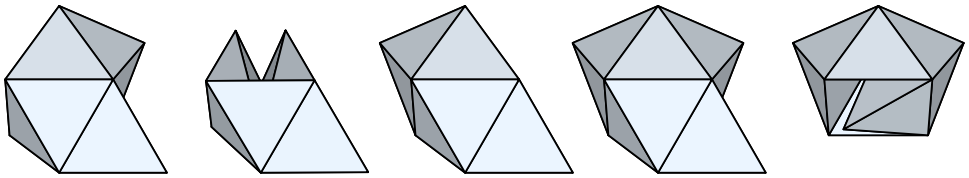


Figure 7: Realization forms of the graph 10_{230}.

tion forms are shown. Similarly, deltahedron 11_{1236}, belonging to the family of this composite deltahedron, also has multiple forms, two of which are shown in Figure 8(b). However, another deltahedron, 11_{1235}, has face intersections in one of the tucking forms (Figure 8(a), right), resulting in fewer valid realizations. Note that additional realization forms may exist beyond those shown here.

Owing to the difficulty of enumerating all the possible forms, our algorithm did not consider multiple realization forms. Nevertheless, there might be cases where tucking or inverse augmentation could eliminate self-intersection instances.

4.2 Computational Performance Evaluation

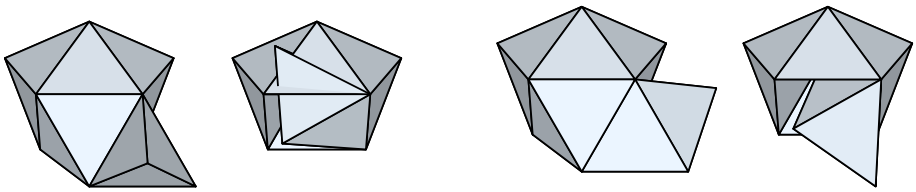
To assess the effectiveness of the optimization strategy presented in Section 3, we compared Algorithm 1 and Algorithm 2 across graphs with 5 to 12 vertices.

4.2.1 Experimental Setup

Both algorithms were implemented in Java using the VF2 algorithm via the JGraphT library for graph and subgraph isomorphism testing. All experiments were conducted on Intel Core i9-10900X, 32GB RAM PC with each vertex size tested five times to assess variability. The reported times include all preprocessing (index building from planar code files) and realization computations, excluding file output operations as these are identical across both algorithms.

4.2.2 Performance Comparison

Table 4 presents the computational performance comparison for graphs with 10 to 12 vertices, where the computational cost becomes substantial. For smaller graphs (5–9 vertices), the optimized algorithm achieves speedup ranging from 1.81× to 1.95×, with absolute execution times remaining under one second for both methods.



(a) 11_{1235} (self-intersection in the right) (b) 11_{1236}

Figure 8: Deltahedra with shared components showing different realizability depending on realization form.

Table 4: Computational performance comparison for graphs with 10–12 vertices

Vertices	Graphs	Baseline	Optimized	Speedup
10	233	1.29 s	676 ms	1.91×
11	1,249	39.4 s	20.3 s	1.94×
		(0.66 min)	(0.34 min)	
12	7,595	1,671s	821 s	2.03×
		(27.8 min)	(13.7 min)	

The results demonstrate that the optimization provides substantial practical benefits for larger graphs. For the 11- and 12-vertex cases (1,249 and 7,595 graphs, respectively), the optimization saves approximately 19.1 seconds and 14.2 minutes (from 27.8 to 13.7 minutes). The speedup factor remains highly consistent across graph sizes 6–12 (1.81–2.03×), with the absolute time savings increasing significantly with problem size, making the optimization particularly valuable for comprehensive enumeration of larger deltahedra.

The optimized method also demonstrates improved consistency across trials. For instance, the 10-vertex case shows a standard deviation of 6.2 ms (baseline) versus 1.3 ms (optimized), and the 12-vertex case shows 1.9 s versus 292 ms, representing up to 5-fold improvement in execution stability.

4.2.3 Discussion

The experimental results validate the effectiveness of exploiting the degree-3 vertex property. The optimization achieves efficiency gains through three mechanisms:

1. **Reduced search space:** Testing only $(n-1)$-vertex graphs for 98% of cases, rather than all sizes from 4 to $n-1$
2. **Faster isomorphism testing:** Using graph isomorphism rather than subgraph isomorphism, which is computationally less expensive
3. **Candidate filtering:** Degree histogram matching eliminates non-viable candidates before isomorphism testing

The consistent speedup across vertex sizes 6–12 (1.81–2.03×) demonstrates the robustness of the optimization strategy. The degree-3 vertex property remains highly prevalent ($> 97\%$) even for larger graphs, ensuring consistent performance gains. For very small graphs (5 vertices), the optimization overhead slightly outweighs the benefits, but this represents a negligible absolute time difference (<0.2 ms) for a case that is rarely of practical interest.

5 Conclusion

In this study, we extended our previous work [9] on polyhedral realization as deltahedra by introducing algorithmic optimizations based on structural properties of triangulated graphs.

We identified that degree-3 vertices, which occur in over 97% of graphs with 10 or more vertices, necessarily form tetrahedra with their neighbors. Exploiting this geometric property, we developed an optimized algorithm that achieves approximately 2× speedup across all graph sizes through case-based search strategies, graph isomorphism testing, and histogram filtering. The optimization is particularly effective for larger graphs, where the computational cost is substantial.

We extended the experimental scope to 12 vertices and provided comprehensive performance evaluation demonstrating both improved efficiency and execution time stability. The results validate that exploiting geometric constraints can yield substantial practical improvements in polyhedral realization algorithms.

All computational results, including 9,150 realized deltahedra with their 3D coordinates have been released as an open dataset [10]. The dataset contains both composite deltahedra and non-composite deltahedra, along with metadata including realization status, geometric properties, and component relationships. This dataset enables further research on polyhedral geometry and provides a foundation for future investigations of deltahedra properties.

Future work includes further investigation of non-composite deltahedra, as 4-connected triangulated graphs represent a geometrically interesting subset, and enumeration of multiple realization forms for graphs admitting them. Rigorous verification of realizability for non-composite deltahedra, particularly near-miss cases, via symbolic computation methods also remains an open problem.

References

[1] J. Bokowski and M. Cuntz: *Hurwitz's regular map* $(3, 7)$ *of genus 7: A polyhedral realization.* Art Discrete Appl. Math. **1**(1), 2017. doi: 10.26493/2590-9770.1186.258.

[2] G. Brinkmann and B. D. McKay: *Fast generation of planar graphs (expanded edition).* MATCH Commun. Math. Comput. Chem. **58**(2), 323–357, 2007. ISSN 0340-6253.

[3] D. Eppstein: *On Polyhedral Realization with Isosceles Triangles.* Graphs and Combinatorics **37**(4), 1247–1269, 2021. ISSN 0911-0119, 1435-5914.

[4] M. Firsching: *Realizability and inscribability for simplicial polytopes via nonlinear optimization.* Math. Program. **166**(1), 273–295, 2017. ISSN 0025-5610, 1436-4646.

[5] P. Gailiunas: *Twisted Domes.* Bridges: Mathematical Connections in Art, Music, and Science 45–52, 2004.

[6] R. Kaufman: *Convex Triamond Regular Polyhedra.* https://www.interocitors.com/polyhedra/Triamonds/. [Accessed 07-04-2025].

[7] S. Nishimoto, T. Horiyama, and T. Tachi: *Geodesic Folding of Regular Tetrahedron.* J. Geom. Graphics **26**(1), 81–100, 2022.

[8] G. Rote: *Realizing Planar Graphs as Convex Polytopes*. In M. Van Kreveld and B. Speckmann, eds., *Graph Drawing*, vol. 7034, 238–241. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-25877-0, 978-3-642-25878-7.

[9] N. Tsuruta: *Polyhedral Realization as Deltahedra Using Subgraph Isomorphism Test*. In K. Takenouchi, ed., *ICGG 2024 - Proceedings of the 21st International Conference on Geometry and Graphics*, 108–115. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-71225-8.

[10] N. Tsuruta: *Deltahedra Realization Dataset*, 2025. doi: 10.5281/zenodo.18344653.

[11] N. Tsuruta, J. Mitani, Y. Kanamori, and Y. Fukui: *Random Realization of Polyhedral Graphs as Deltahedra*. J. Geom. Graphics **19**(2), 227–236, 2015.

[12] E. Wohlleben and W. Liebermeister: *Shapes and Deformations of Polyhedral Rings Formed by Corpuscle Elements*. J. Geom. Graphics **16**(1), 59–67, 2012.

Received January 23, 2026; final form March 30, 2026.