A Graph Theory Approach to Permutation Puzzle Design

By Tanner Frisby

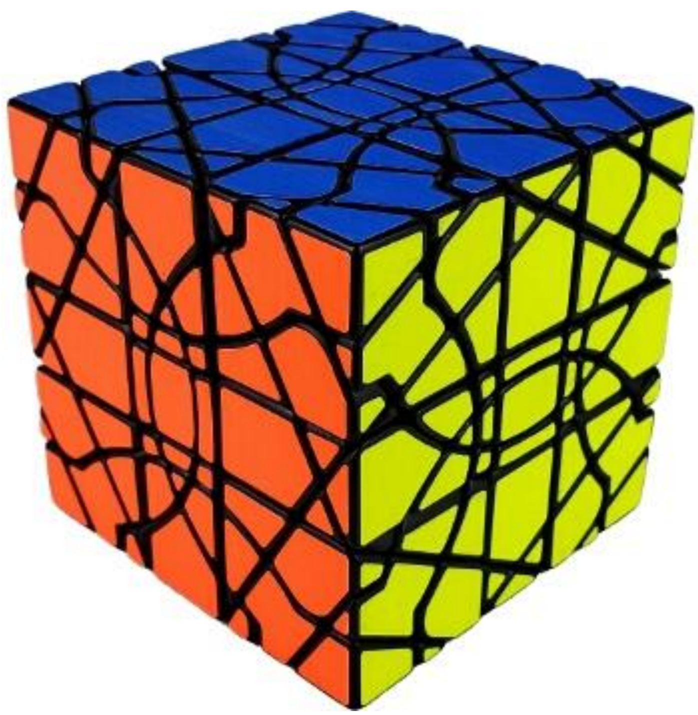
Contents

Introduction

A traditional design formula for creating twisty puzzles, albeit far from the only formula, is roughly as follows:

1. Select a solid
2. Define an axis system based on the faces/edges/vertices of the selected solid
3. Define the cut curve geometry (planar, conic, spherical, etc.)
4. Define the depth(s) of the cut(s)
5. Select a degree of (un)bandaging
6. Select the exterior solid
7. Map the axis system to the exterior solid (bump/ghost/etc)

Using Mark Gagliano's embodiment of a giga-hexaminx from 2019 as an example:



1. Regular dodecahedron (regular icosahedron)
2. Axis system based on faces (vertices)
3. Conic transitioning to cylindrical
4. Two shallow cut layers per axis, resulting in 13 piece-types
5. Doctrinaire, no bandaging
6. Cube
7. All vertices of the cube are placed on vertices of the dodecahedron (all vertices of the cube are placed on the center of a face of the icosahedron)

Steps 1 and 2 of the example result in an axis system with 12 axes, each of which are adjacent to 5 other axes at a dihedral angle of ~116.57 degrees.

Most twisty puzzles have an axis systems defined based on “named solids”, which are *solids* having unique characteristics and have therefore been classified. Platonic solids are solids whose faces are all identical regular polygons meeting at the same dihedral angles. Archimedean solids are convex uniform polyhedra composed of regular polygons meeting in identical vertices (excluding platonic solids, prisms and antiprisms, and the pseudo-rhombicuboctahedron). Catalan solids are the duals of Archimedean solids. Prisms, antiprisms, dipyramids, Johnson solids… the list goes on. While many named solids result in twisty puzzles with enjoyable characteristics, there exist axis systems which result in enjoyable characteristics but aren’t derived from named solids. The purpose of this paper is to explore an alternative axis system design methodology and provide a basic framework for implementing the methodology.

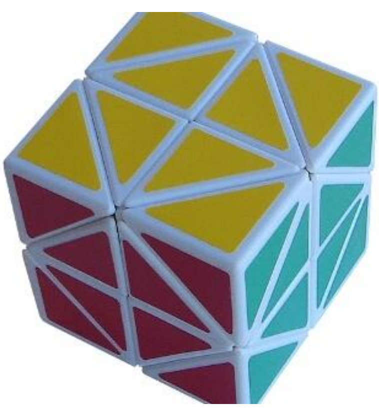
Axis System Characteristics

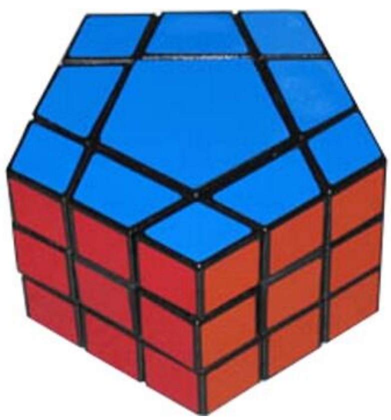
Some of the axis system characteristics which influence the resulting puzzle are listed below:

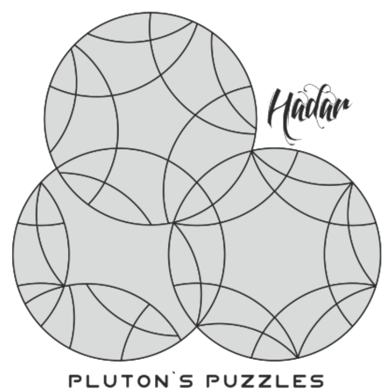
Quantity of Dihedral Angles

Most permutation puzzles have either one or two dihedral angles. Puzzles with more than one dihedral angle tend to jumble but the converse is not necessarily true.

The Bevel / Helicopter cube by Katsuhiko Okamoto / Adam G Cowen is the first example of a jumbling puzzle and has a single dihedral angle of 120 deg. Uwe Meffert / Lee Tutt’s “Pentagonal Prism” is the first examples of a two-dihedral-angled axis system in a twisty puzzle, 90 deg and 108 deg. Dmitry Andreev’s Hadar is the first example of a doctrinaire axis system where all three axes are separated by different lengths.

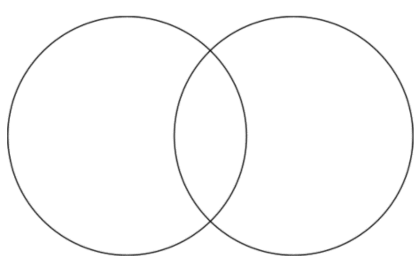




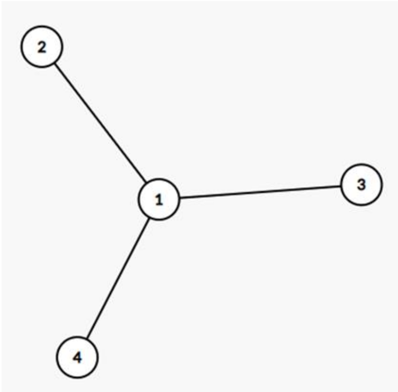


Number of Stops per Each Axis

If an axis is constrained (by the puzzle’s current state) to only have one stop, then the axis is trivial; while it can spin, it can only spin back to its starting position. For example, a trivial intersecting circles puzzle in the shape of a Venn diagram would have two axes, each with one stop.

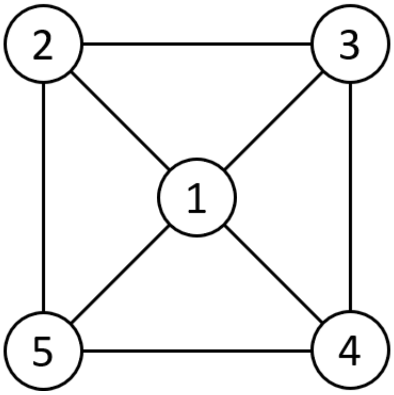


Axes which only ever have two stops may also be trivial as the scramble sequence could have limited possibility of branching. Axes with two stops can be bridges for piece swapping between otherwise independent subsets of axes, which can inflate move count and tedium without adding much to the solving experience. This simplest bridgeless axis system can be found in the "Flower" puzzle produced by Clever Toys. The axis system can be represented by a bridgeless order four tree.



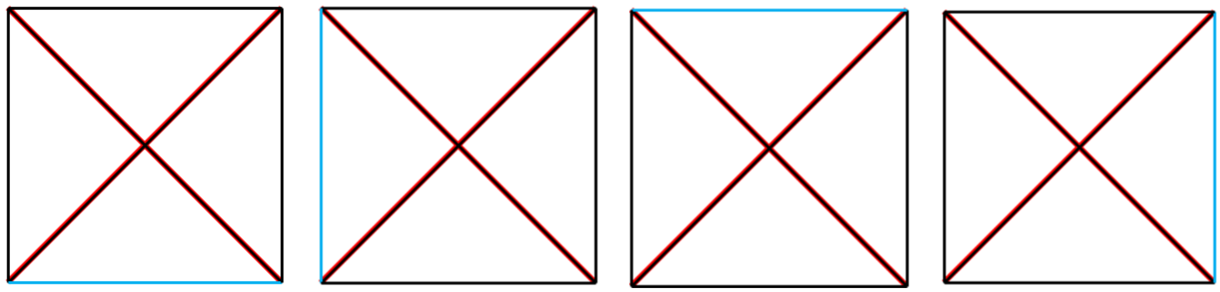
Axes with three or more stops tend to be more desirable from a solver's standpoint.

The number of stops on puzzles with an axis system having more than one dihedral angle is influenced by the distribution of axes pairs at each dihedral angle. Let a vertex (V1) of a simplicial planar graph representing an axis system have degree four and each of V1's adjacent vertices are also adjacent to two other V1-adjacent vertices. Let each vertex represent an axis and each edge represent the dihedral angle between a pair of adjacent axes.

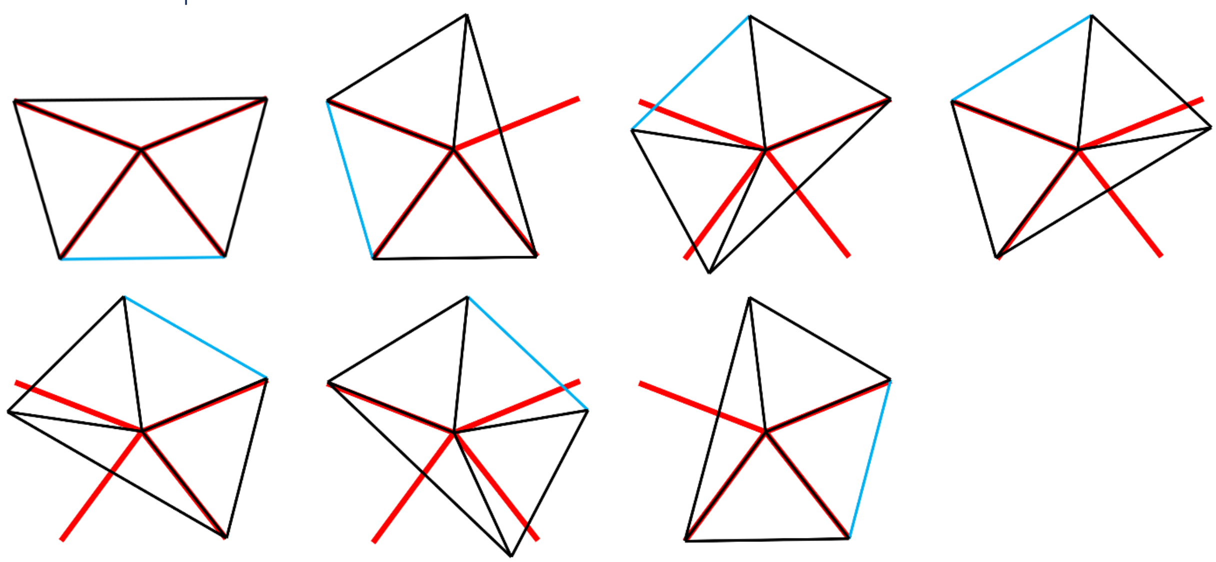


Let the axis system have up to two dihedral angles, angles "A" and "B". The sequences of dihedral angles between pairs of adjacent vertices about V1 determines the stops when rotating about V1; the four stop sequence possibilities for a fourth-degree axis are given below.

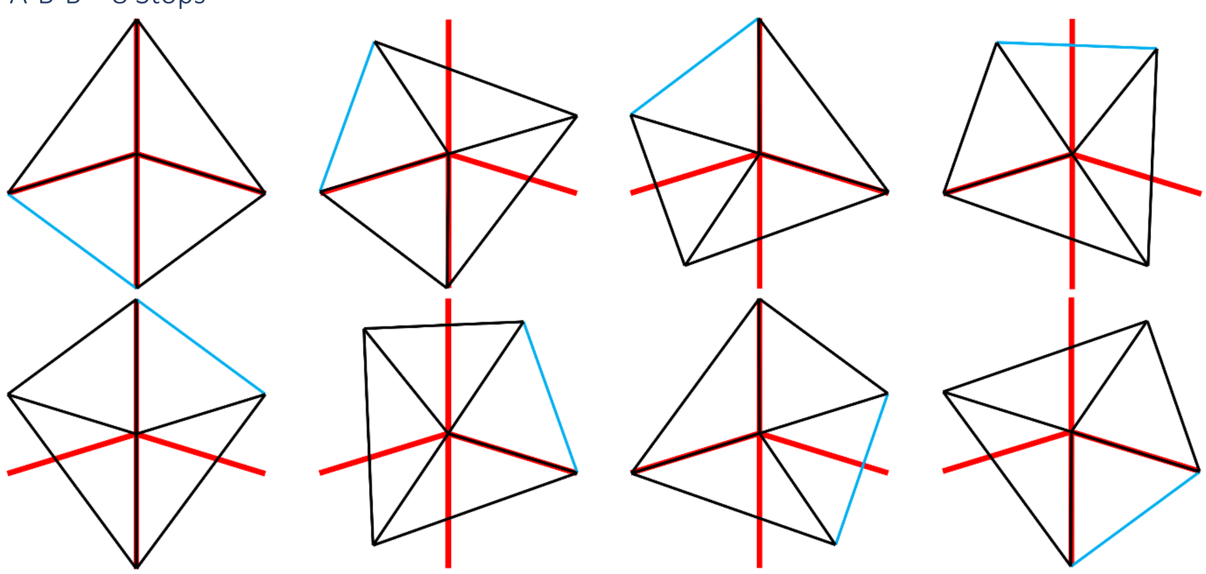
A-A-A-A – Four Doctrinaire Stops



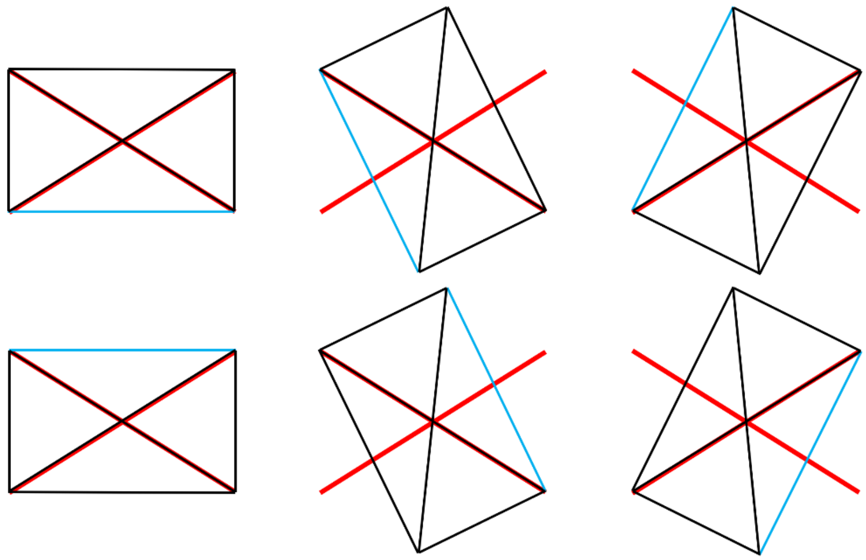
A-A-A-B – 7 Stops



A-A-B-B – 8 Stops

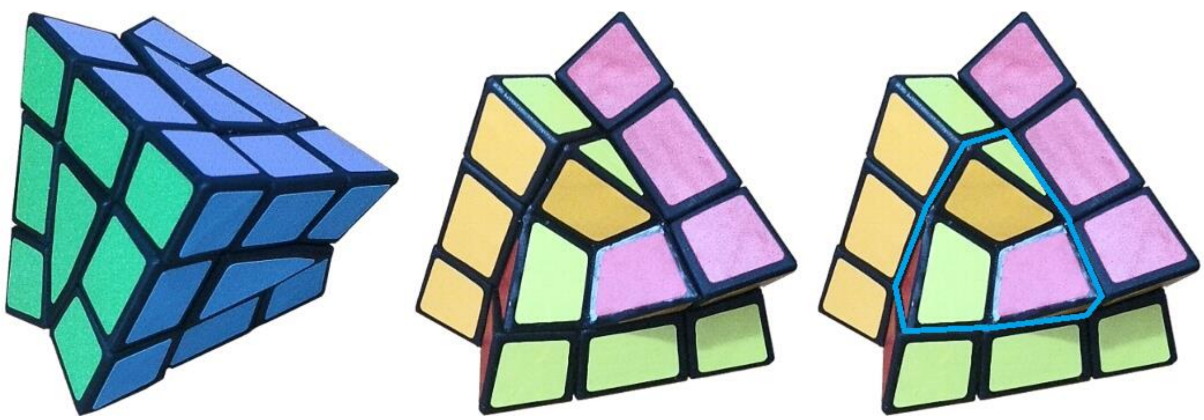


A-B-A-B – 6 Stops

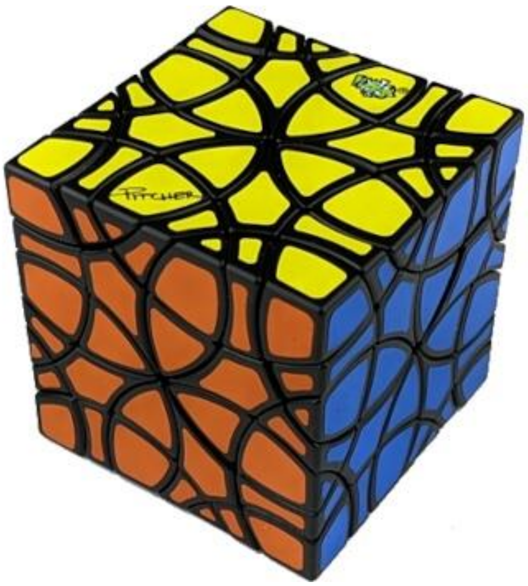


Axis Density

The distribution of axes is not required to be maximally diffuse. Dead space (the characteristic space of a non-maximally diffuse axis system) can be a degree of freedom to enable otherwise over-constrained axis systems, but can come at the expense of a lower average stop count. One example of dead space enabling an axis system can be found on Fenz / Evgeniy Grigoriev's "Bermuda Flower" Series. Highlighted in cyan in the third picture below is the dead space present on the "Bermuda Flower – Earth" variant. A fourth axis placed in the dead space can maintain the axis system's singular dihedral angle with at most 2 of the 3 axes. Increasing the singular dihedral angle to approximately 70.529 degrees (tetrahedral) or decreasing it to 60 degrees (planar 3-axis) would provide a maximally diffuse axis system with a singular dihedral angle.



Alternatively, dead space can be utilized to reduce the number of axes while preserving piece-types. This tends to dramatically cut down on a puzzle's tedium ratio and can provide a more difficult solving experience. One example of this tedium-reduction dead space is presented in David Pitcher's "Andromeda" series. "Face of Andromeda", "Edge of Andromeda", and "Corner of Andromeda" all maintain the same piece-types found in "Andromeda". Due to the reduction in adjacent axes, algorithms which work on "Andromeda" are not guaranteed to work on the reduced versions. This typically results in an increase in solving difficulty.



Generating an Axis System

Let the narrowest of the dihedral angles be denoted by “A_1” and the widest dihedral angle be denoted by “A_n”. Let each axis be defined by a vertex coincident to the axis and unit length away from the axis system’s origin.

Required Constraints

Introduce the following constraints to each vertex:

1. All other vertices whose distance to the current vertex is less than or equal to Sin(180-A_n) shall have a distance of Sin(180-A) with respect to the current vertex, where “A” is one of the axis system’s dihedral angles.
2. The current vertex must be a distance of Sin(180-A) away from at least one other vertex.
3. The current vertex must have a unit length with respect to the origin.

Other constraints may be added to elicit desired properties, but the above constraints are the minimum required. Two methods of satisfying the constraints will be discussed: a brute force method and a guided method. The brute force method is ideal for discovering all possible variants of an axis system defined by given constraints. The guided method is ideal for identifying near-miss solutions (fudge-able puzzles), steering the solver to a specific embodiment, and for recognizing patterns/associations for future implementations.

Brute Force Method

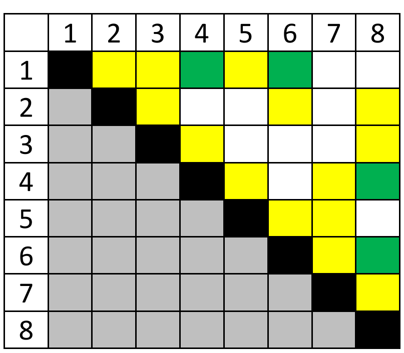
The brute force method attempts to solve for every possible axis system that could arise from a given quantity of axes and dihedral angles. The solve time increases very quickly as axis count increases, especially if other constraints aren't programmed in. The methodology is as follows:

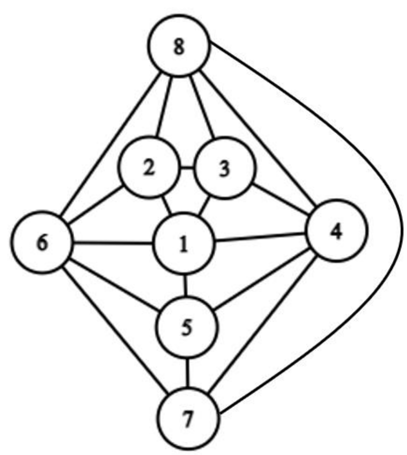
1. Identify all simplicial planar n^{th}-order graphs, where "n" is the desired quantity of axes.
2. Create d^E variants of each identified graph where "d" is the maximum number of dihedral angles and "E" is the size of the graph. This accounts for all possible distributions of dihedral angles on edges.
3. Generate a system of equations based on each graph to solve for the dihedral angle(s).
    a. If the system can't be solved, then an axis system with the given constraints doesn't exist.
    b. If the system solves with finitely many solutions for each dihedral angle, then the axis system exists and is maximally diffuse.
    c. If the system solves with infinitely many solutions for at least one dihedral angle, then the axis system exists.

Designing an Axis System Through Brute Force

For this example, attempt to design an 8-axis maximally-diffuse jumbling puzzle which is not defined by any named solid:

The following is an adjacency matrix representation of an interesting optimization of the design criteria. Green and yellow squares denote dihedral angles of A and B respectively.





The graph is simplicial (since it can be completely expressed in the above adjacency matrix format) and planar per the Kuratowski's and Wagner's theorems. One planar representation of the graph is displayed above.

From a permutation puzzle perspective, some special traits of this axis system are as follows:

1. Has the densest quantity of edges (18) possible for 8 axes: e = 3v – 6. Maximizing edges for a given number of vertices is sufficient, but not necessary, for proving the axis system will be maximally diffuse.
2. Is bridgeless (which follows from having the densest quantity of edges).
3. Has more than one dihedral angle (2), which tends to produce jumbling axis systems.
4. Has at least one iso-dihedral Hamiltonian cycle (example: 1,2,8,3,4,7,6,5). This should result in an axis system with deep scrambles.

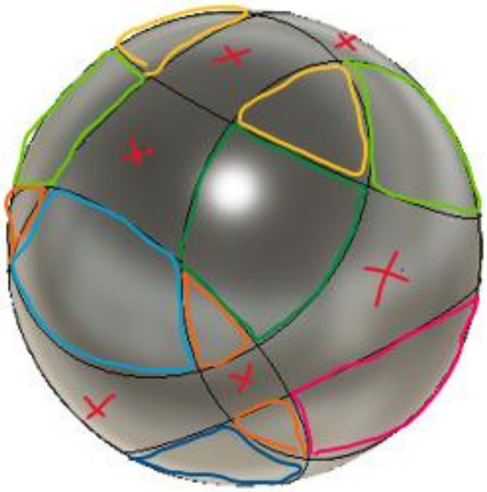
The system of equations defining this axis system solves to exactly one pair of dihedral angles A and B, approximately 82.929 deg and 106.307deg respectively.

Embodying the Axis System in a Puzzle

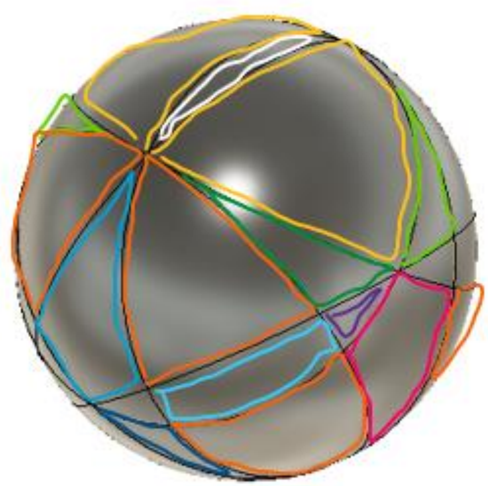
Following the design formula outlined in the introduction, but using the above axis system instead of steps 1 and 2:

3. Conic
4. Single shallow cut creating 8 piece-types
5. No (un)bandaging for one embodiment and 1st-level of unbandaging for a second embodiment
6. Cube
7. 5th-degree vertices intersect equatorial edges of the cube, 4th-degree vertices placed to give 180-degree axial symmetry about the polar axis of the cube

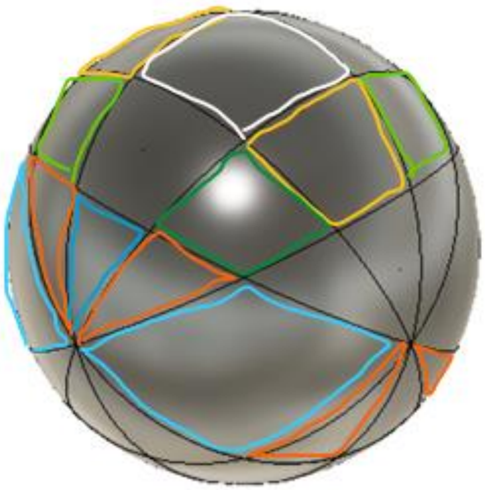
Furthermore, a deep cut puzzle based on this axis system appears relatively simple to manufacture; 4 shallow cut piece-types persist through to the deep cut embodiment.



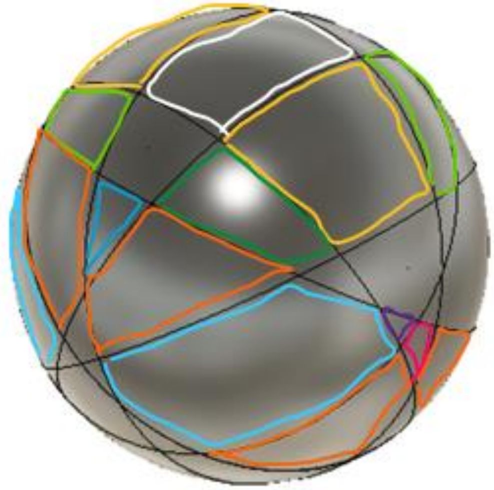
57



73.7



90



85

Guided Method

One implementation of a guided method is to utilize parametric modeling software to enforce and release constraints until the axis system meets its predefined requirements. The following outlined process should be repeatable in any parametric modeling software. The generalized steps are as follows:

1. Seed the necessary number of points about a unit radius sphere.
   Note: A too uniformly dispersed seed could progress easier into a named solid, while a bunched-up seed could be harder to bloom into an axis system without dead space. A good balance for even dispersion while maintaining disorder could be to map vertices of a named convex polyhedron whose vertices do not all land on its circumscribed sphere (Johnson solids for example).
2. Constrain each seeded vertex to have a unit length with respect to the origin.
   Note: This could be accomplished by dimensioning the distance between each point and the origin to a length of one unit or by constraining each vertex to be coincident to a unit sphere centered at the axis system's origin.
3. Map the edges of the convex polyhedron.
   Note: It is not always easy to determine upfront which edges should be expressed and this process may require iteration.

4. Constrain pairs of expressed edges to be equal (but currently undefined) in length. Continue constraining until you have assigned all of the expressed edges to one of A- groups, where A- is the number of dihedral angles.

   Note: As this process unfolds, it may become apparent that expressed edges from section 3 will not be edges of the convex polyhedron. These edges should be removed. Conversely, it may become apparent that two unlinked vertices will have an expressed edge between them on the final convex polyhedron. These vertices should be linked with an edge.
5. Iterate between steps 3 and 4 until an axis system sufficiently meeting the design goals has been created.

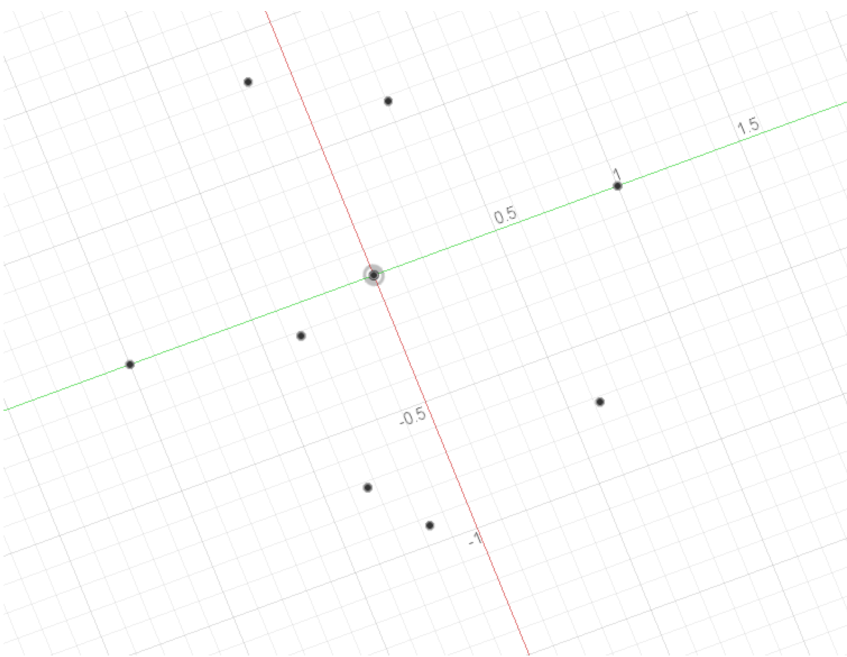
   Note: If the resulting system has no dead space, then the previously undefined edge lengths will now be defined. The dihedral angle(s) supplement can be found by converting the chord lengths to angles between adjacent axes (or by directly measuring through the CAD package). If the resulting system does have dead space, the lengths of each A- group may be varied to produce a puzzle of preferable characteristics. Alternatively, the addition of another vertex or the merger of existing vertices could result in a puzzle with no dead space. These modifications usually require further iteration between steps 3 and 4.

Designing an Axis System with the Guided Method

For this example, attempt to design an 8-axis maximally-diffuse jumbling puzzle which is not defined by any named solid:

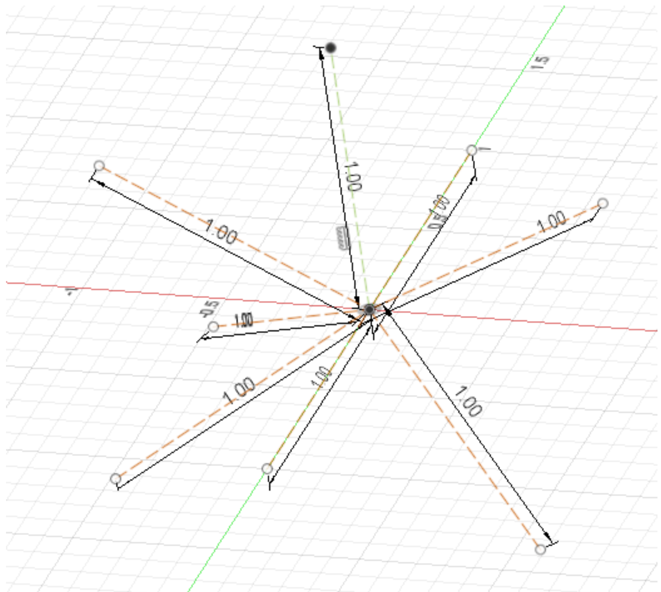
Seed the Axis System Vertices

8 vertices are placed in 3D-space from a seed.



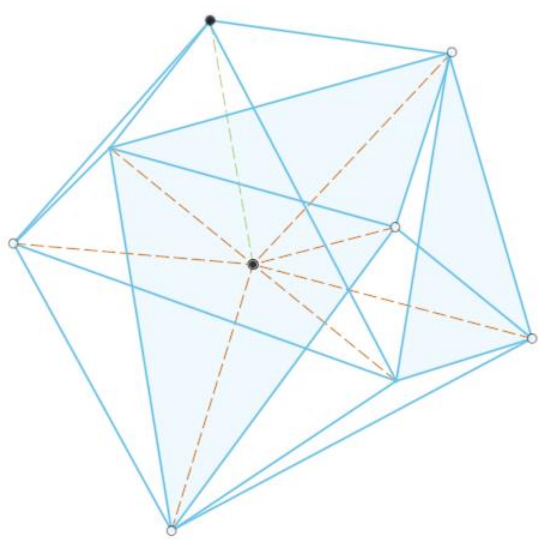
Constrain the Vertices

A reference line with a length of one unit is used to connect each seeded vertex to the origin. One reference line is optionally constrained coincident to the Z-axis. A second vertex is optionally constrained coincident to the X-plane (normal to the red axis). These two conditions constrain rigid body rotations about the origin, which simplifies the guided process.



Map the Edges of the Convex Polyhedron

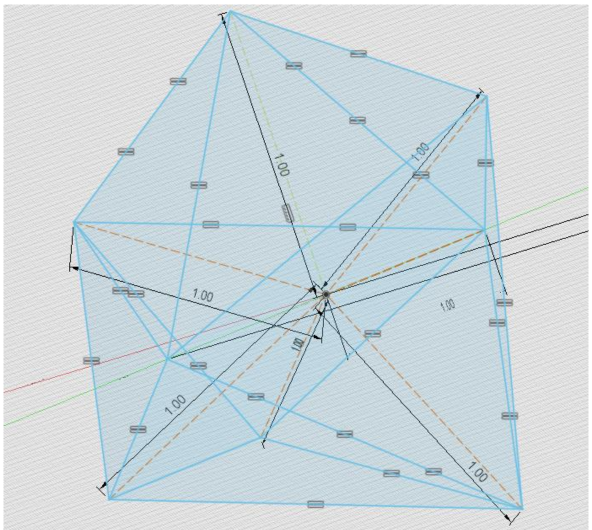
The edges of the complete K_8 graph that will be expressed as edges of the resulting convex polyhedron are mapped. This expression of edges could change as the vertices move during the constraining process. The wireframe is shown below (the origin is between the adjacently shaded triangles and the viewer):

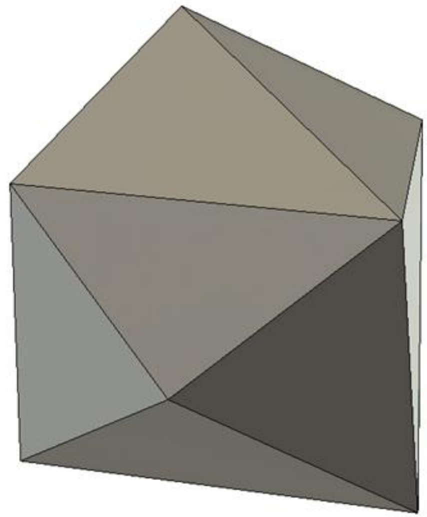


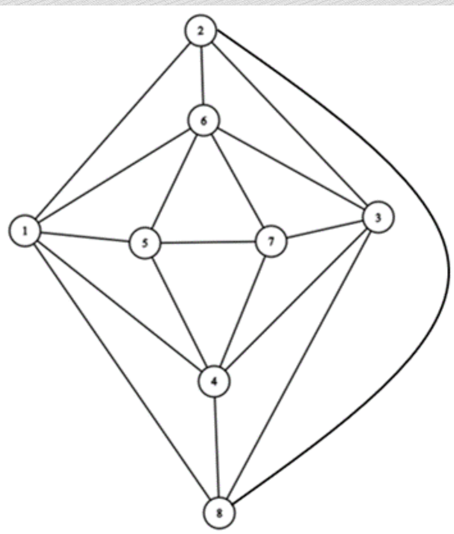
Constrain Edges to Solve for Dihedral Angles

One pair of similarly long edges are constrained to be of equivalent length. For an axis system with two dihedral angles, a second independent set of edges are constrained to be of equivalent length. Incrementally, all other edges are constrained to be of equivalent length to one of the two sets of edge lengths. If the last edge constrains successfully, a maximally diffuse solution has been achieved.

In this particular example, the length of the last edge was fully defined upon constraining the next-to-last edge. Furthermore, the length of the last edge was not a member of either edge length set.







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

Dihedral angles are 120 deg for yellow, 90 deg for green, and 84.7 deg for edge 4,8.

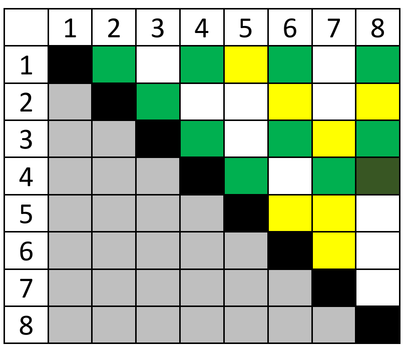
Constraining the Final Edge - Troubleshooting

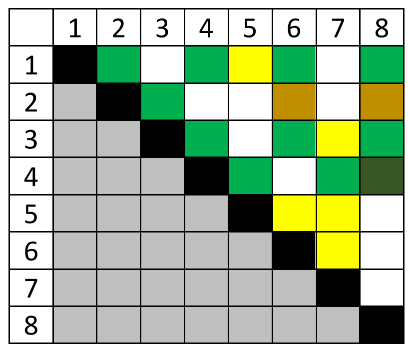
If the last edge does not constrain successfully to any existing set of edge lengths, the designer has a few options:

1. If the system is not fully constrained, adjust the length of each set to cause the unconstrained edge to be optimally close to the length of an existing set. Depending on how close the discrepancy is, this could be a viable [fudged](#) axis system.
2. If the system is fully constrained, introduce at least one additional dihedral angle constrained to be optimally close to an existing dihedral angle. Depending on how close the discrepancy is, this could be a viable fudged axis system.
3. Exchanging some of the members of one or more sets of constrained edges may enable the last edge to become a member of an existing edge length set.
4. If the unconstrained edge is the longest mapped edge, removing the unconstrained edge will yield an axis system with dead space.
5. Introduce an additional degree of freedom in the form of an extra dihedral angle by creating an additional iso-length edge set. Members of previously constrained edges may be moved to different sets.

Fudging

An optimal fudged solution which maintains the axis system's mirror symmetry is produced by also fudging edges 2,4 and 2,8. The adjacency matrix on the left is the previous matrix, the matrix on the right is the optimally fudged matrix.

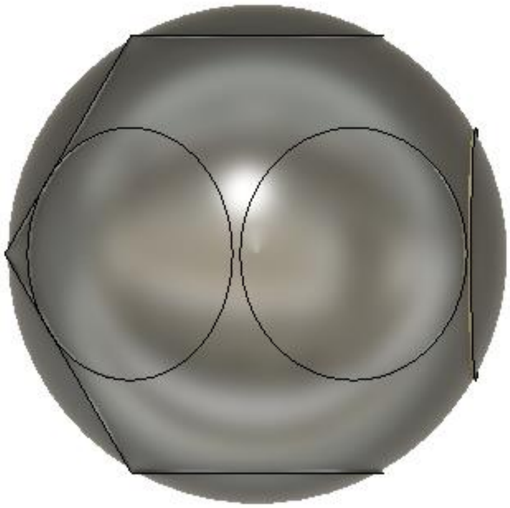
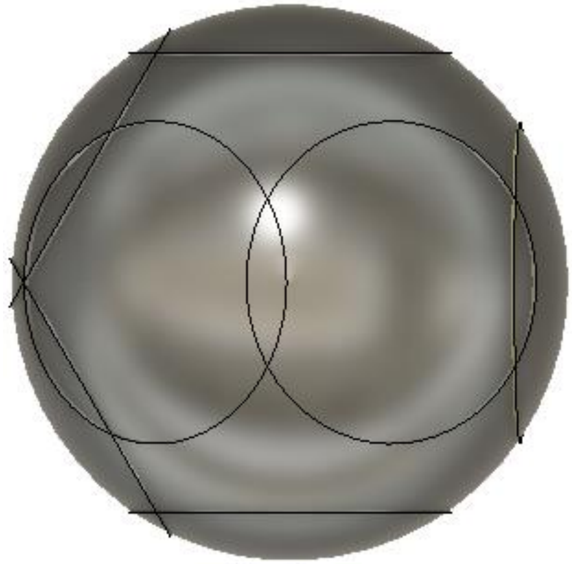
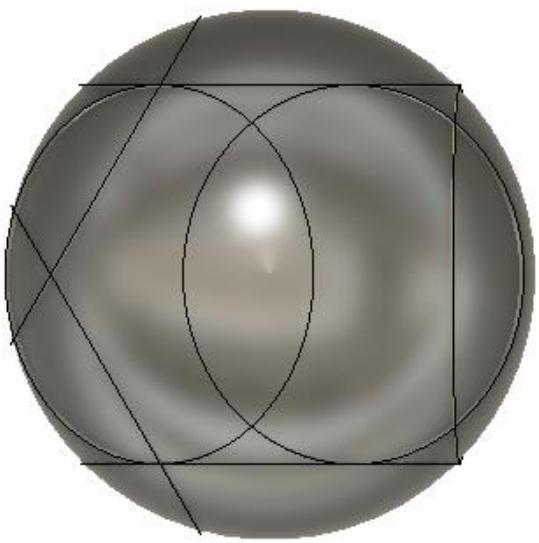
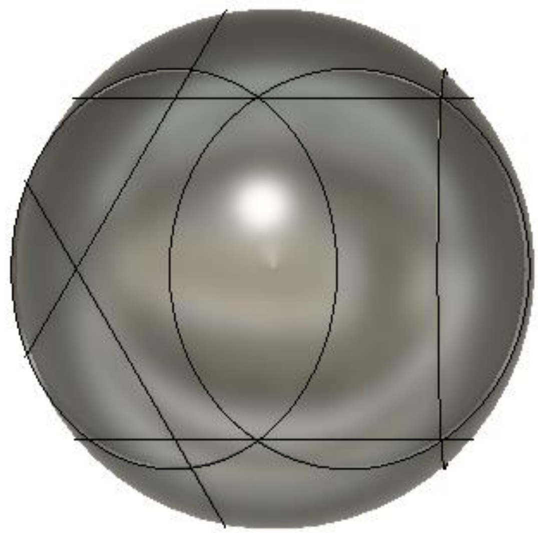
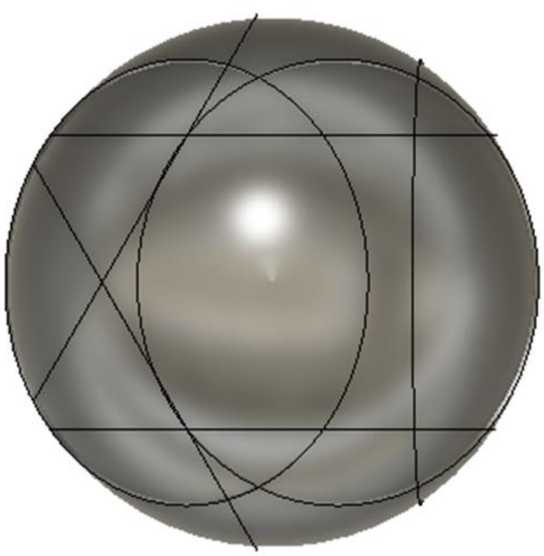
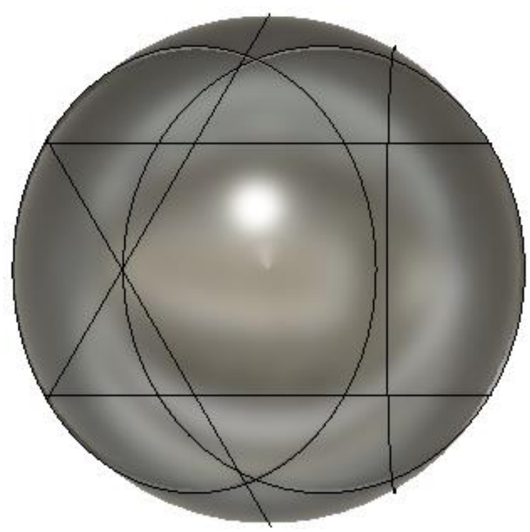


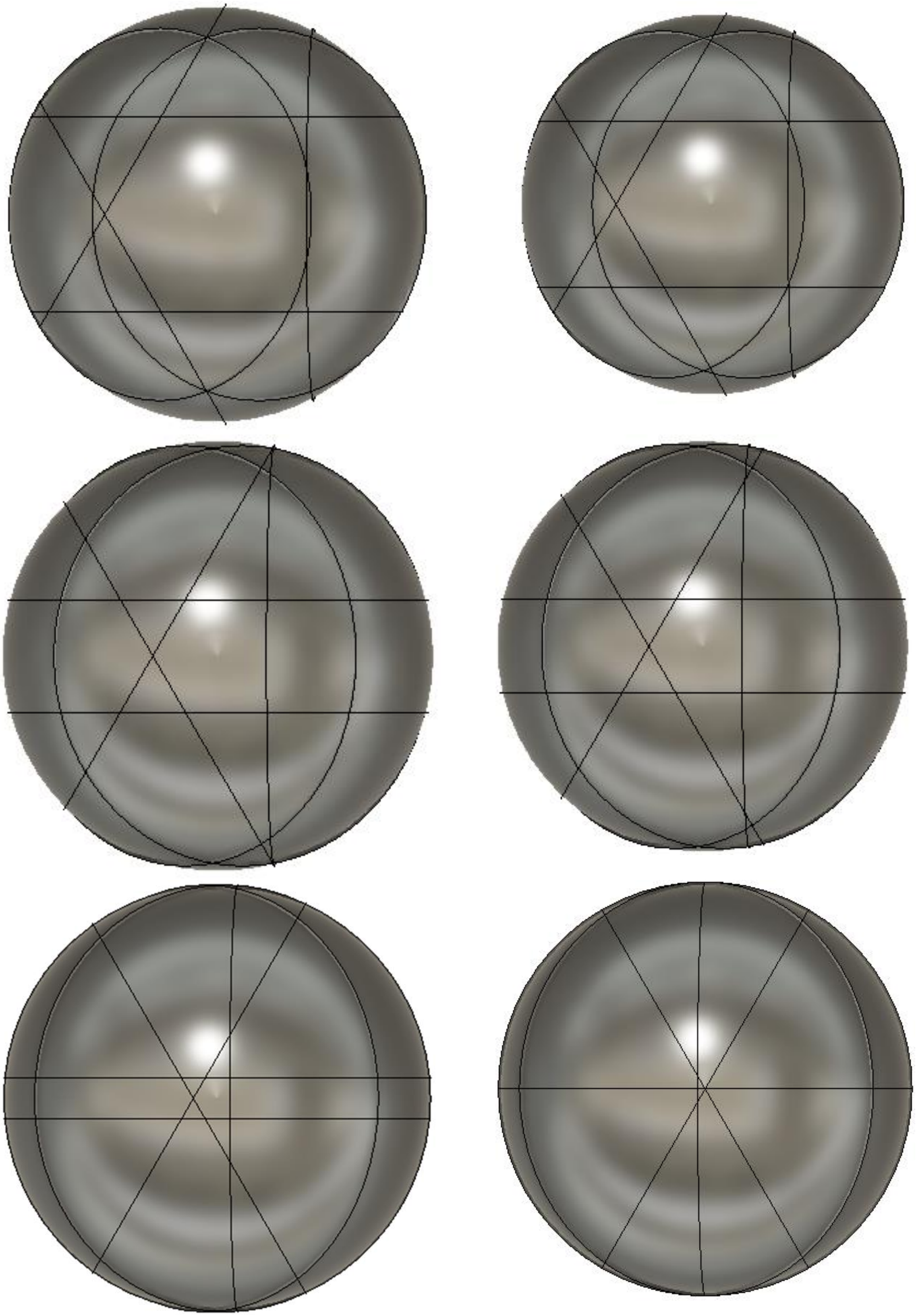


Yellow and green maintain dihedral angles of 120 and 90 degrees respectively with fudged yellow and fudged green now having dihedral angles of 118.25 and 88.25 degrees respectively. Importantly, a dihedral angle fudged to be smaller will provide slightly loose pieces between the fudged axes which is desirable compared to pieces having a slightly tighter fit. This fudged axis system is viable.

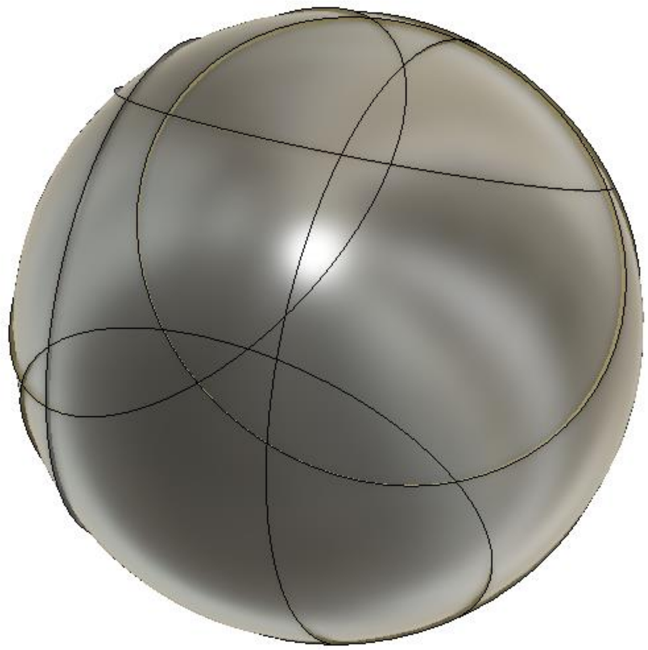
Cut Depths

The piece-type critical angles in degrees are approximately: 30, 35, 45, 49.5, 56.5, 60.5, 62.5, 64.5, 75, 77, 84.5, 90.

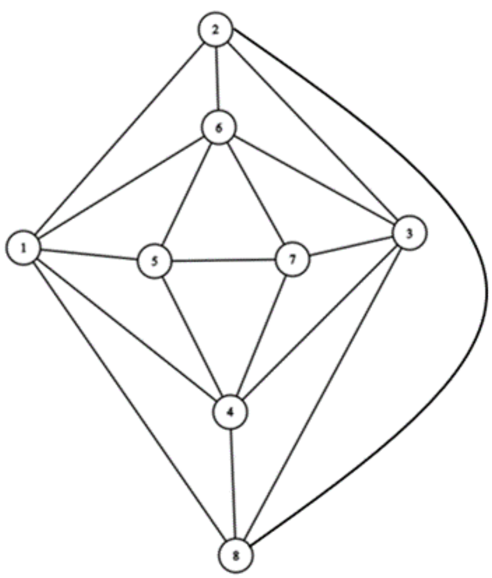


A puzzle with cut depths ~53 or ~76 degrees could make for an interesting puzzle. Both depths suffer from a disparity in piece size as well as multiple very small piece-types. This makes manufacture (especially at a reasonable size) difficult. Below is a 53-degree embodiment.

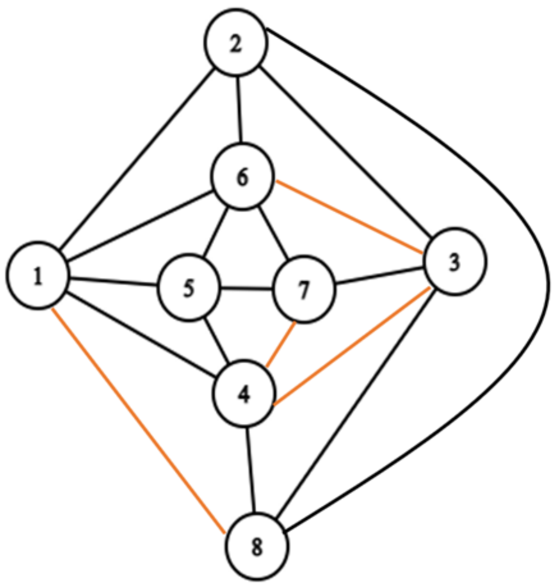


Reconstraining the Edges

An axis system with only two dihedral angles based off of the example graph can be created by adjusting the members of the two sets of edge lengths. Below is the original example followed by the re-membered adjacency matrix; each edge that changed sets is marked with an "X".

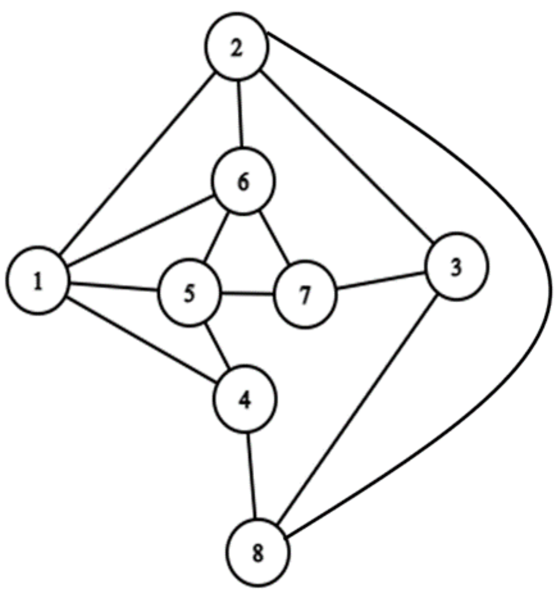


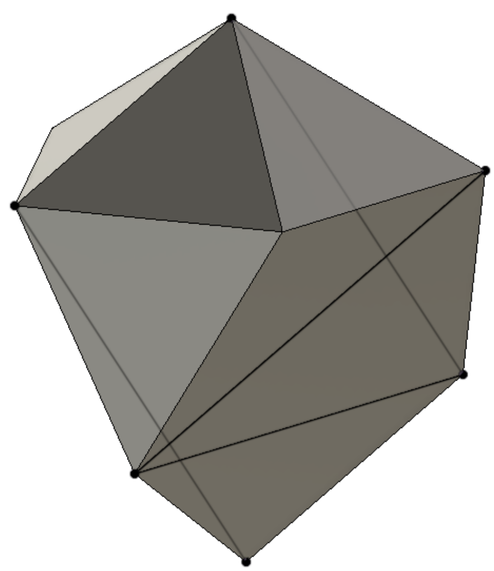
	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								



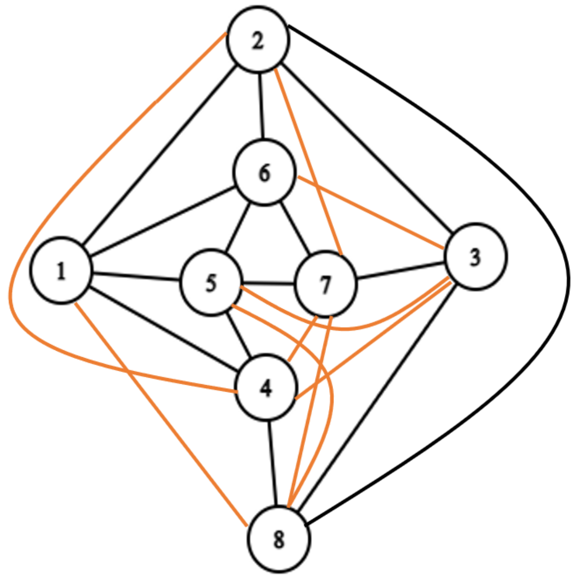
	1	2	3	4	5	6	7	8
1				X		X		
2						X		X
3								X
4					X			X
5								
6								
7								
8								

Under the new set of constraints, not all of the convex polyhedron's faces are triangular. The highlighted orange edges of the above graph indicate edges which are now coincident to a non-triangular face of the polyhedron.



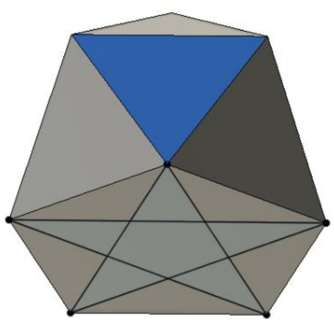


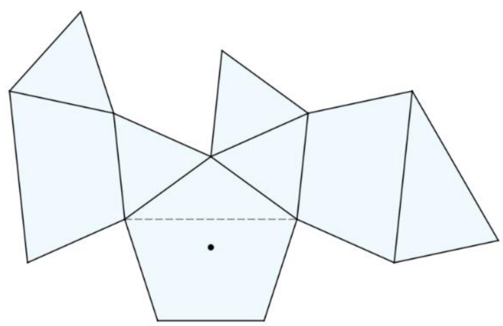
Notably, the regular pentagonal face 3,8,4,5,7 has an edge length and chord length of the two dihedral lengths. Non-expressed (face-coplanar) edges are marked with an "X" in the adjacency matrix below.

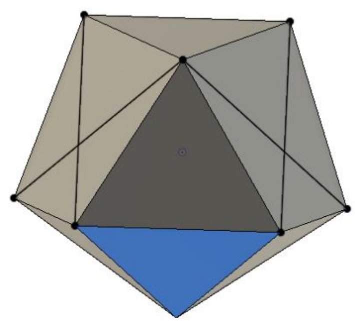


	1	2	3	4	5	6	7	8
1								
2				X			X	
3					X			
4								
5								X
6								
7								X
8								

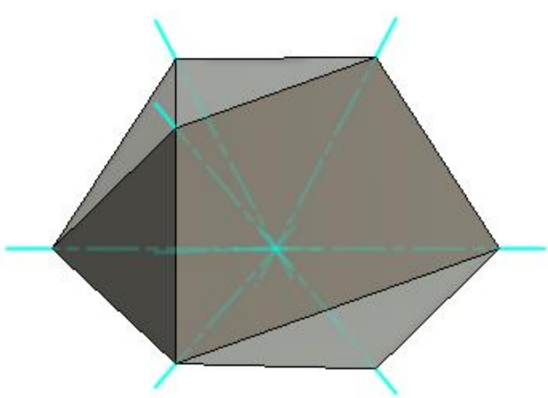
Including the non-expressed edges, this graph would have 23 of the 28 edges of the complete K_8 graph. This incredibly interconnected axis system is a strong candidate for the basis of an interesting permutation puzzles. The net is given below; it is composed of a regular pentagon, 4 regular triangles, two trapezoids formed by truncating a regular pentagon, and one isosceles triangle.

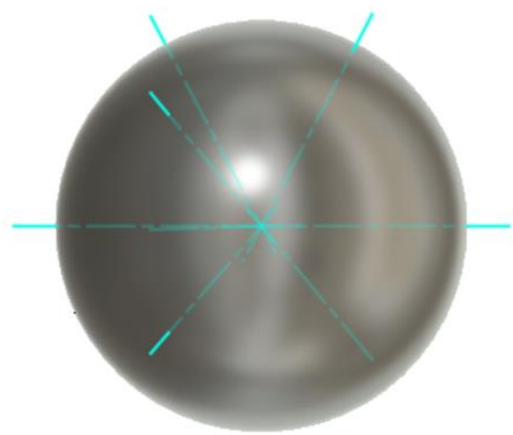




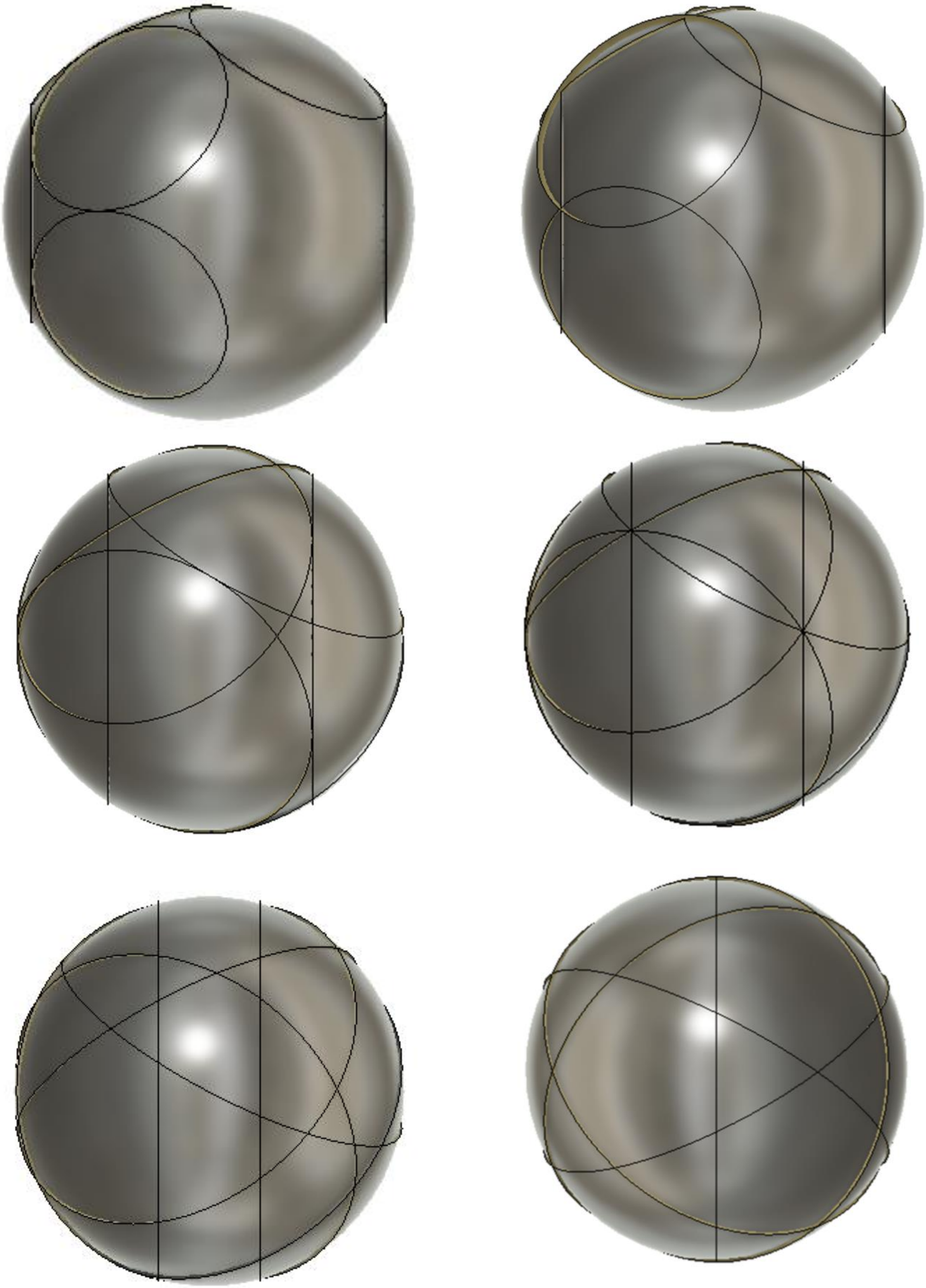


Next, the axis system is applied to a ball:

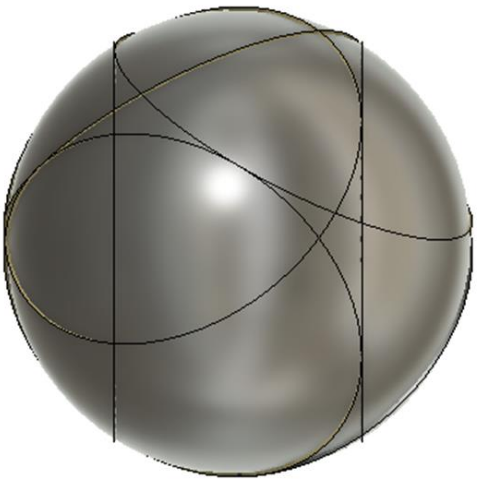


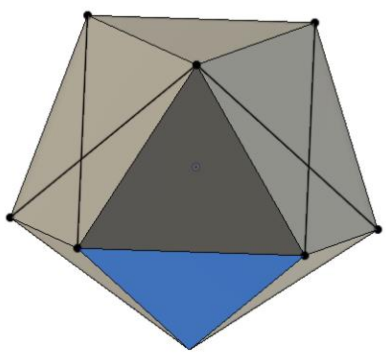


The piece-type critical revolve angles in degrees are approximately 31.7, 37.4, 58.3, 63.4, (75), and 90.



The savvy observer may recognize the source of this axis system's unique properties. Namely, this novel 8-armed jumbling axis system is neither novel nor jumbling; it is a face-centered platonic dodecahedral axis system (or vertex centered platonic icosahedron) with 4 axes removed (corresponding to vertices 2, 4, 5, and 7 extended through the origin). However, an argument can be made that this proposed design method *did* produce an enjoyable axis system as the above non-trivial embodiments are spherical bandaged versions of the Megaminx, Pyraminx Crystal, Starminx, and Pentultimate.

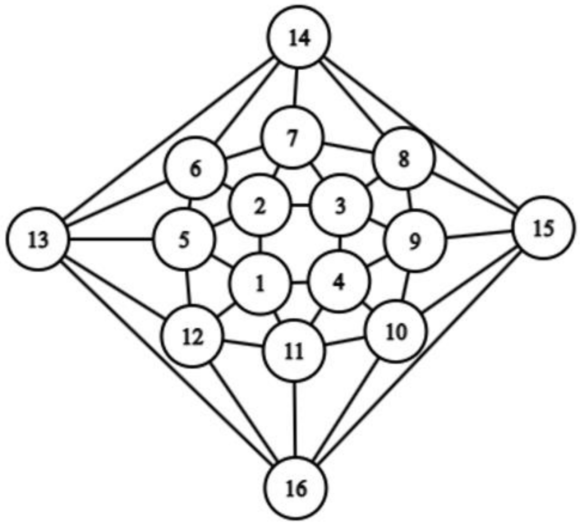




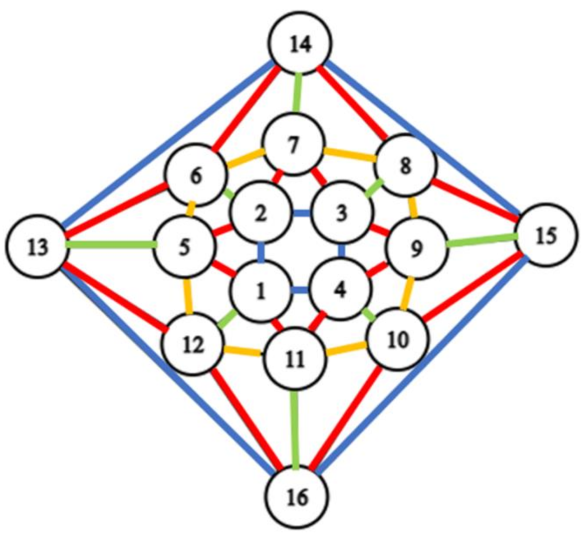
Highly Symmetric Axis System, Constraint Methods

On highly symmetric axis systems, it can be possible to partition edges by symmetry groups. These symmetry groups can then be further grouped into dihedral groups when an axis system is being designed with more than one dihedral angle.

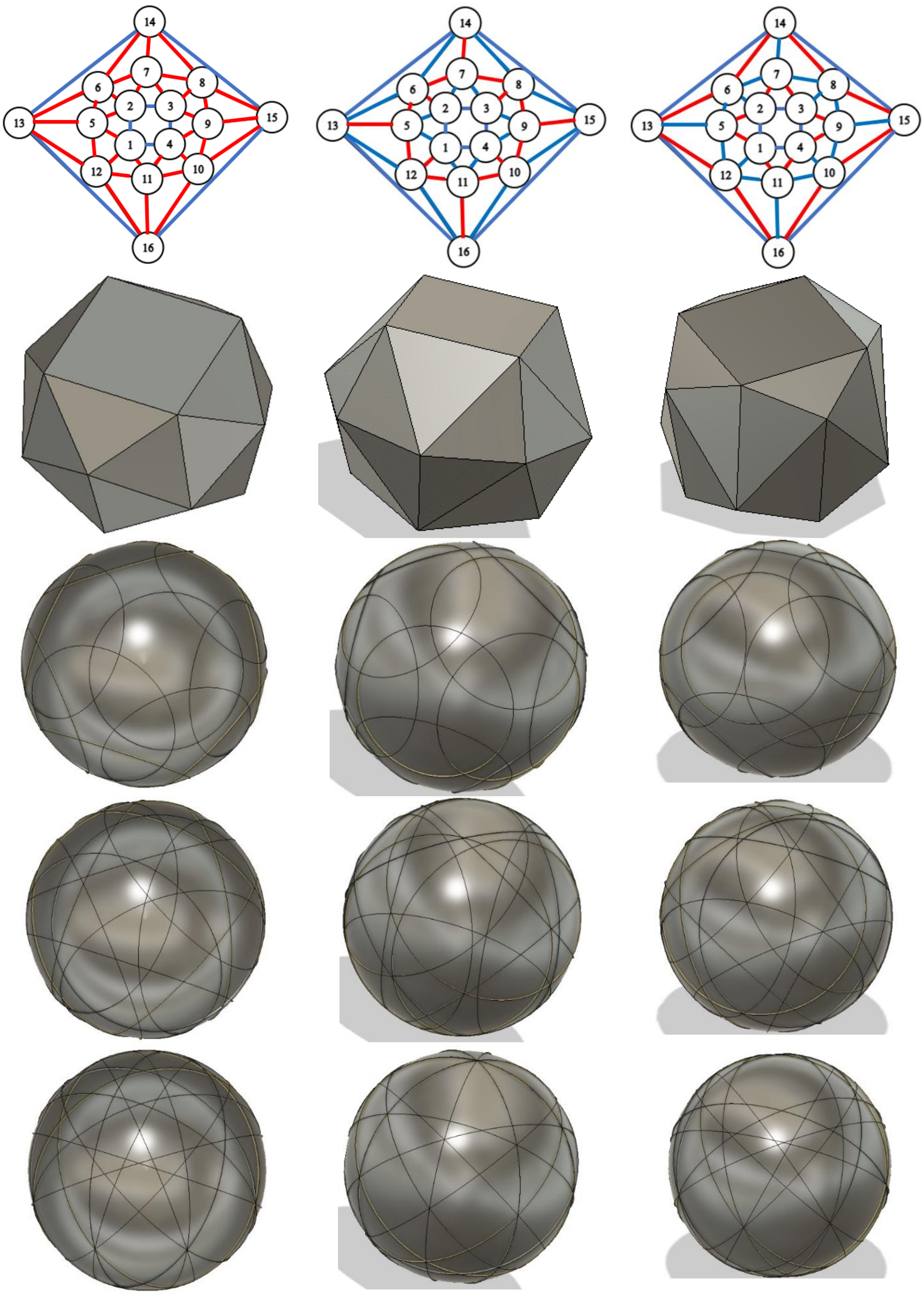
Take for example the following simplicial planar graph of an axis system which does not converge to a solution when constrained to a single dihedral angle:



One possible symmetrical edge sub-grouping is as follows:



Three possible sets of dihedral edge length pairs based on the above subgroups are given below, along with a polyhedral mapping and cut depth spheres at 40 (shallow), 65 (medium), and 90 degrees (deep):


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

Glossary

Fudging

Adjusting the geometry of a twisty puzzle to make something fit or turn that would otherwise not fit or turn. Twistypedia Entry

Cut

Part of the cut curve that is visible at the outside of the puzzle. A cut can be an active cut or a stored cut. A cut can be a straight cut or a curvy cut. Twistypedia Entry

Cut Curve

The revolved curve that separates one turning part of a twisty puzzle from another. Twistypedia Entry

Shallow cut

Cut curve that includes the rotation axis and covers less than 180 degrees from the origin. Twistypedia Entry

Deep cut

A puzzle cut where the cutting surface divides two isomorphic groups. Twistypedia Entry

Doctrinaire

A puzzle where if you were to remove all the coloration then every single position would look exactly the same. Twistypedia Entry

Jumbling

A twisty puzzle 'jumbles' when it cannot be unbandaged into a doctrinaire puzzle. Twistypedia Entry

Bandaging

Restricting moves of a puzzle. Classical bandaging is achieved by "gluing" pieces together. Twistypedia Entry

Piece-type

A member of the set of unique pieces of a given permutation puzzle.

Stop

A location in which the current axis may cease rotation while possibly permitting rotation on an adjacent axis dependent on the state of the scramble.

Bibliography

Glossary of Graph Theory. (2022). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Glossary_of_graph_theory

Twisty Puzzles Museum. (2022). Retrieved from Twisty Puzzles: www.twistypuzzles.com

Van Deventer, O. (2019, December 19). *Twistypedia*. Retrieved from Puzzles by Oskar Van Deventer: http://twistypedia.oskarvandeventer.nl/index.php/Twistypedia