

Rolling Polyhedra on Tessellations

Akira Baes  

Département d'Informatique, Université libre de Bruxelles, Belgium

Erik D. Demaine  



Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Martin L. Demaine  

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

Elizabeth Hartung  

Department of Mathematics, Massachusetts College of Liberal Arts, North Adams, MA, USA

Stefan Langerman  

Département d'Informatique, Université libre de Bruxelles, Belgium

Joseph O'Rourke  

Department of Computer Science, Smith College, Northampton, MA, USA

Ryuhei Uehara  

School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan

Yushi Uno 

Graduate School of Informatics, Osaka Metropolitan University, Japan

Aaron Williams  

Department of Computer Science, Williams College, Williamstown, MA, USA

Abstract

We study the space reachable by *rolling* a 3D convex polyhedron on a 2D periodic tessellation in the xy -plane, where at every step a face of the polyhedron must coincide exactly with a tile of the tessellation it rests upon, and the polyhedron rotates around one of the incident edges of that face until the neighboring face hits the xy plane. If the whole plane can be reached by a sequence of such rolls, we call the polyhedron a *plane roller* for the given tessellation. We further classify polyhedra that reach a constant fraction of the plane, an infinite area but vanishing fraction of the plane, or a bounded area as *hollow-plane rollers*, *band rollers*, and *bounded rollers* respectively. We present a polynomial-time algorithm to determine the set of tiles in a given periodic tessellation reachable by a given polyhedron from a given starting position, which in particular determines the roller type of the polyhedron and tessellation. Using this algorithm, we compute the reachability for every regular-faced convex polyhedron on every regular-tiled (≤ 4)-uniform tessellation.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Theory of computation \rightarrow Design and analysis of algorithms; Mathematics of computing \rightarrow Discrete mathematics

Keywords and phrases polyhedra, tilings

Digital Object Identifier 10.4230/LIPIcs.FUN.2022.6

Supplementary Material *Software*: <https://github.com/akirbaes/RollingPolyhedron/>
archived at `swh:1:dir:94cacdc90902f3129be316e189a4d64f94ad4537`

Funding *Stefan Langerman*: Directeur de Recherches du F.R.S.-FNRS.

Ryuhei Uehara: JSPS KAKENHI Grant Numbers JP18H04091, JP20H05961, JP20H05964, JP20K11673

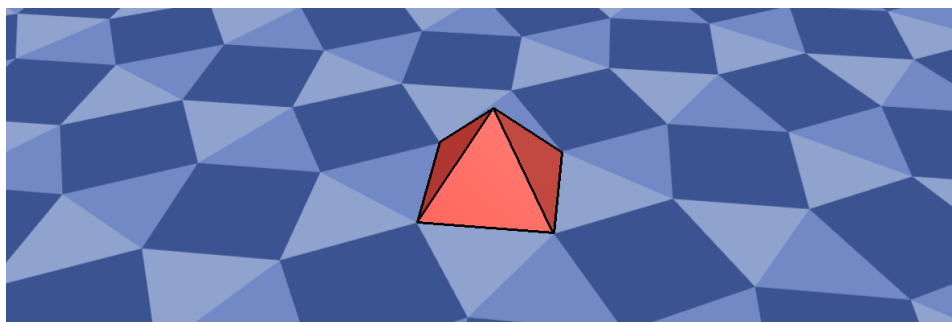


© Akira Baes, Erik D. Demaine, Martin L. Demaine, Elizabeth Hartung, Stefan Langerman, Joseph O'Rourke, Ryuhei Uehara, Yushi Uno, and Aaron Williams;
licensed under Creative Commons License CC-BY 4.0

11th International Conference on Fun with Algorithms (FUN 2022).
Editors: Pierre Fraigniaud and Yushi Uno; Article No. 6; pp. 6:1–6:16



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Screenshot from an interactive 3D rolling visualization program on the subject of this paper [3].

Acknowledgements Part of this work appeared in the first author’s Master’s Thesis. Part of this work was done at the 1st and 2nd Virtual Workshops on Computational Geometry (2020 and 2021). The authors would like to thank all participants of those workshops. Renders of prisms and antiprisms are by Robert Webb’s Stella software. Other polyhedron renders are from Wikimedia Commons under Creative Commons Attribution license.

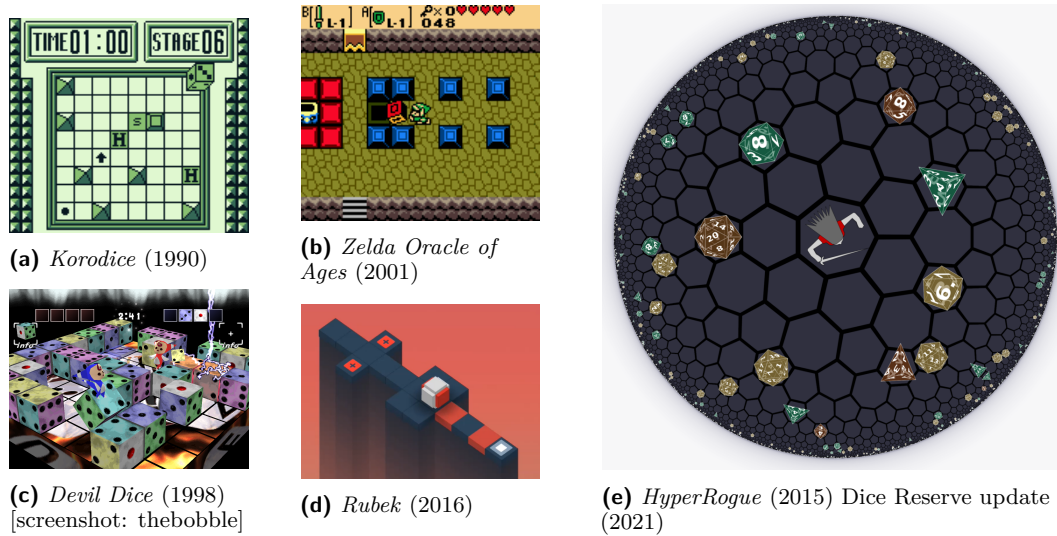
1 Introduction

Dice rolling puzzles feature a cube rolling around on the square grid. The goal is often to match a given face with a given tile. Such puzzles were popularized by Martin Gardner [11, 12, 13], and are featured in a variety of computer games, such as *Korodice* (Gameboy, 1990), *Super Mario 64* (Nintendo 64, 1996), *Devil Dice* (Playstation, 1998), *Legacy of Kain: Soul Reaver* (Playstation, 1999), *Legend of Zelda Oracle of Ages* (Gameboy Color, 2001), *Bombastic* (Playstation 2, 2002), *Legend of Zelda Spirit Tracks* (Nintendo DS, 2009), *Rubek* (Windows, 2016), *Roll The Box* (Mobile, 2021), and *The Last Cube* (Windows, 2022); see Figure 2. Cube rolling puzzles have been occasionally generalized to rolling other polyhedra on other grids. For example, computer game *HyperRogue* (Windows, 2015) involves hexagonal and heptagonal tiles in a hyperbolic space, and in its 2021 update, rolling tetrahedron, octahedron, or icosahedron dice on a triangular lattice; see Figure 2e. With various constraints, rolling puzzles can be NP-complete [6, 18], and when rolling multiple shapes, they can be PSPACE-complete [5, 16].

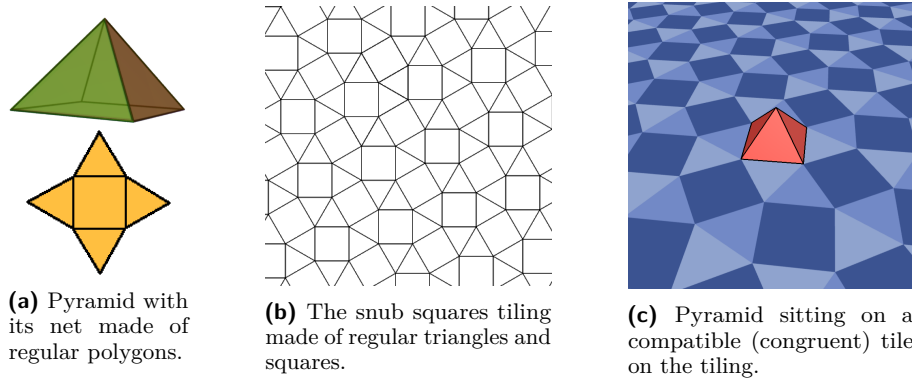
Previous work has explored rolling a polyhedron to reach any position and orientation in the plane [8, 4]. Akiyama [1] defined a *frame-stamper* as a regular polyhedron that covers the whole plane with a tiling by rolling the polyhedron in arbitrary directions, and a *tile-maker* as a polyhedron whose unfoldings all tile the plane. A more relaxed definition in [2] determines all *tessellation polyhedra* – regular-faced convex polyhedra that have at least one unfolding that tiles the plane.

1.1 Rolling Rollers

We formalize the concept of rolling any convex 3D polyhedron P on any tessellation T , which we imagine as lying in the (horizontal) xy -plane; refer to Figure 3. Recall that a plane tessellation is a partition of the plane into a collection T of polygons called *tiles* [15]. We restrict our attention to *edge-to-edge* tilings where two touching tiles share either a whole polygon edge or a vertex. When a tile of T is congruent to a face of P , we call them *compatible*.



■ **Figure 2** Cube and dice-rolling puzzles in video games.



■ **Figure 3** A polyhedron and a tessellation with compatible faces are required for rolling.

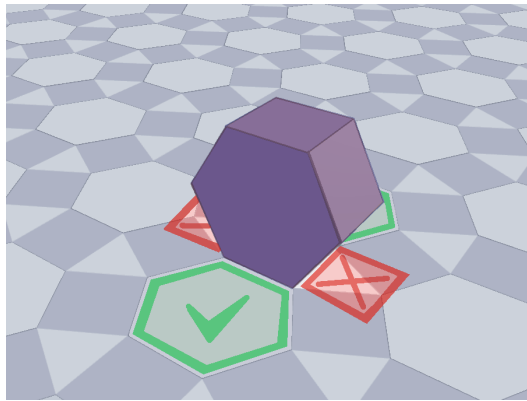
To start, we place the polyhedron P on the tessellation so that one of its faces *rests* on (i.e., coincides exactly with) a compatible tile. In a *rolling step*, we rotate the polyhedron about one of the edges of its resting face, until another face rests on the tessellation. For the roll to be *valid*, we insist that, at the end of the motion, the adjacent face of P across the rolling edge rests on another (adjacent) compatible tile. See Figure 4 for an example.

Valid sequences of rolls form paths in the *rolling graph* of possible configurations; see Section 2.2 for a formal definition. If the rolling graph contains a connected component that includes every tile of T , then we call the polyhedron a *plane roller* (denoted by the 🎲 icon) for that tessellation and starting position, as it can eventually roll to cover the entire plane. Other possibilities are 🎲 *hollow-plane rollers*, which cover a constant fraction of the plane while leaving holes; 🎲 *band rollers*, which cover an infinite area that is a vanishing fraction of the plane; and 🎲 *bounded rollers*, which are confined to a finite area.

1.2 Our Results

In this paper, we develop a polynomial-time algorithm to identify whether a polyhedron is a plane roller, hollow-plane roller, band roller, or bounded roller for a given plane tessellation and starting location, provided the tessellation is *periodic* meaning that its tiles have two linearly

6:4 Rolling Polyhedra on Tessellations



■ **Figure 4** Valid and invalid rolls, marked by green checks and red Xs respectively.



independent translational symmetries. The running time of our algorithm is polynomial in the number of faces of the polyhedron and the number of tiles in the fundamental domain of the the two translational symmetries. We essentially take advantage of the periodicity of the tessellation, coupled with the structure of the polyhedron, to prove that the resulting rolling graph also has a periodic structure that we can exploit.

We then apply this algorithm to completely categorize a natural finite set of interesting



■ **Figure 5** Screenshot of the rolling-pair reachable-area classification interactive table available at <https://akirabaes.com/polyrollly/resulttable/>.

special cases, compiled on the website <https://akirabaes.com/polyrolly/resulttable/> shown in Figure 5. For polyhedra, we consider the *regular-faced* convex polyhedra where every face is a regular polygon: the 5 Platonic solids [9], the 13 Archimedean solids [10], the 92 Johnson solids and their chiral variations [14, 17, 19], the n -prisms for $n \in \{3, 5, 6, 8, 10, 12\}$, and the n -antiprisms for $n \in \{4, 5, 6, 8, 10, 12\}$, as higher-sided polygons cannot be used to tile the plane [15]. For periodic plane tessellations, we consider all “ k -uniform” tilings for $k \geq 4$, as listed in [7]. A plane tessellation is *k-uniform* if its tiles are regular polygons and it is *k-isogonal*, meaning that there are k equivalence classes of vertices (called *orbits*) formed by applying all transformations in the symmetry group to the vertices. All k -uniform tilings are periodic [15].

Including chiral variations of polyhedra that have one, these cases consist of 129 polyhedra and 131 tilings. For each case, we tried all possible starting positions to find the largest connected reachable area, thereby characterizing every pair of polyhedron and tiling as  plane roller,  hollow-plane roller, band roller, or bounded roller. See Figures 6, 7, 8, and 9 for examples of each respective type, and Tables 2, 3, and 4 in Appendix A for a condensed view of all results.

The figures and tables use standard notation for k -uniform tilings based on vertex types [7]. The type of a regular-polygon tile is the number of its sides, and the type of a vertex is the clockwise cyclic order of tile types that surround a vertex. For a k -uniform tiling, there are finitely many vertex types, so the tiling can be labeled by the list of vertex types, with duplicate names differentiated by a subscript. See Figure 10.

The rest of this paper is organized as follows. Section 2 describes our algorithm. Section 3 shows how the results from this algorithm can also assist puzzle designers. Section 4 describes our implementation.

2 The Algorithm

2.1 Tilings

First we review some basics about tilings, following Grünbaum and Shephard [15].

There are uncountably infinitely many tilings, even when restricted to edge-to-edge tilings with regular polygons. For example, the tiling in Figure 6b can be modified to follow any binary sequence of triangle and square rows, and there are uncountably many such binary sequences. We restrict our attention to *periodic tilings* T , which have two linearly independent translational symmetries (say, \vec{a} and \vec{b}) that act on the tiles of T . What this means is that applying the translation vector \vec{a} (respectively \vec{b}) on any tile $t \in T$ produces another tile of T . The symmetry group generated by \vec{a} and \vec{b} decomposes the set of tiles of T into equivalence classes, also called *orbits*, where two tiles are in the same class if there is a symmetry in the group (an integer linear combination $i\vec{a} + j\vec{b}$ for some $i, j \in \mathbb{Z}$) that matches one to the other.

The tiling can then be described by a *fundamental domain* for the action of these symmetries. Figure 11 shows an example. The fundamental domain is a connected subset of the tiles (one tile for each orbit), which glued together form a *supertile* S . We denote by $|S|$ the number of tiles in the supertile. The supertile (and the tiles that compose it) can be repeated by the action of the two translations to obtain the original tiling. As S tiles the plane isohedrally by translation, its boundary can be decomposed into six pieces, denoted by $A, B, C, \bar{A}, \bar{B}, \bar{C}$, counterclockwise, where \bar{A}, \bar{B} , and \bar{C} are translations of A by the action of \vec{a} , B by the action of \vec{b} , and C by the action of $\vec{b} - \vec{a}$, respectively. See Figure 16 (right).

6:6 Rolling Polyhedra on Tessellations

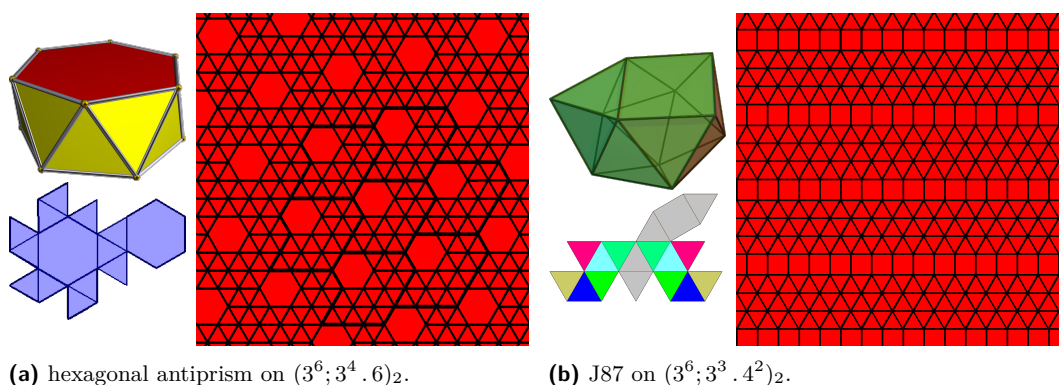



Figure 6 Examples of reachable-area patterns generated by  plane rollers which can reach the entire plane.

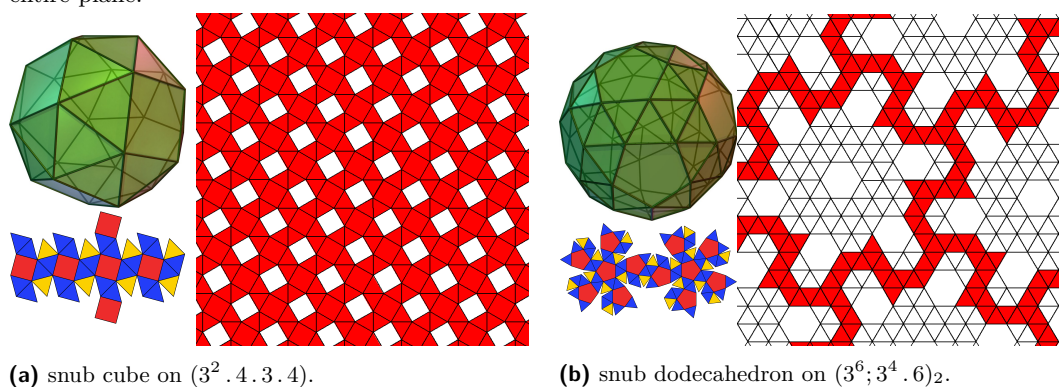



Figure 7 Examples of reachable area patterns generated by  hollow-plane rollers which reach a constant fraction of the plane while leaving holes.

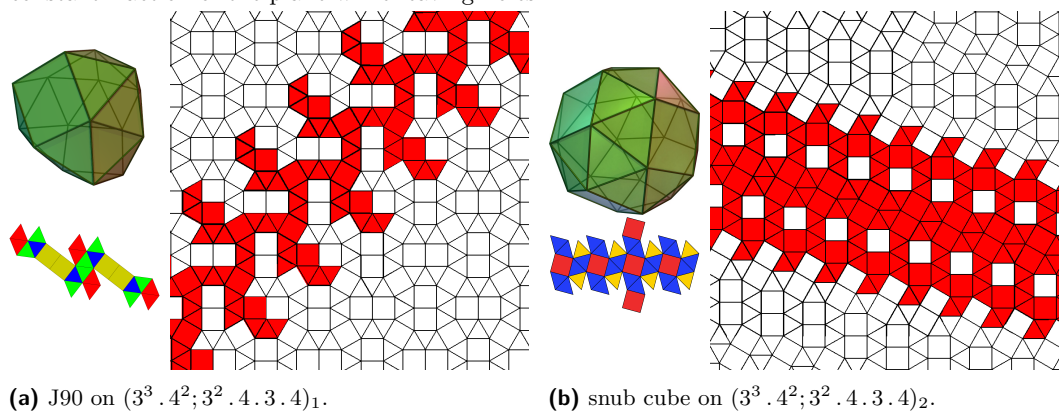



Figure 8 Examples of reachable areas patterns generated by  band rollers which reach an infinite area but a vanishing fraction of the plane, being restricted to an infinite band.

A copy of the supertile can be identified by its integer coordinates in the *basis* formed by the translation vectors \vec{a} and \vec{b} . That is, the copy (i, j) corresponds to the application of the translation $i\vec{a} + j\vec{b}$ to S . An individual tile t of the tiling T can then be uniquely identified by $\langle (i, j), s \rangle$: the coordinates (i, j) of the copy of S it is located in and its representative tile s within S . See Figure 12(a),(c)

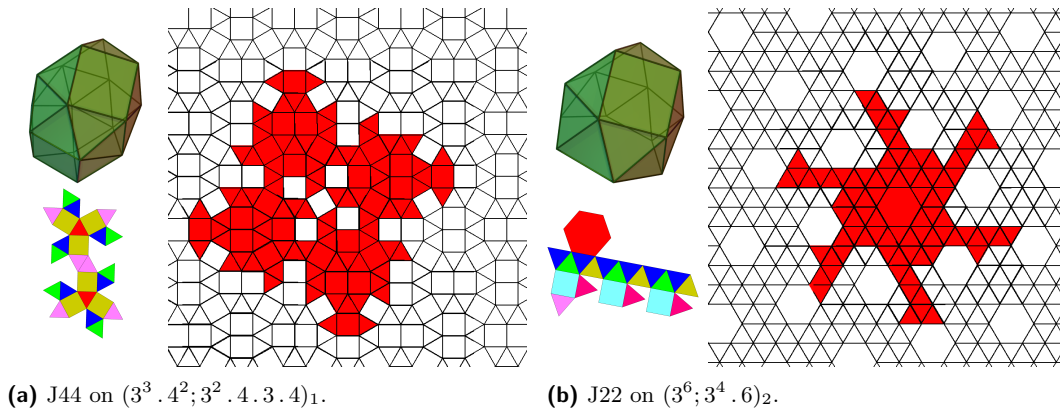


Figure 9 Examples of reachable area patterns generated by \blacktriangleright bounded rollers which are restricted to a finite area containing the start.

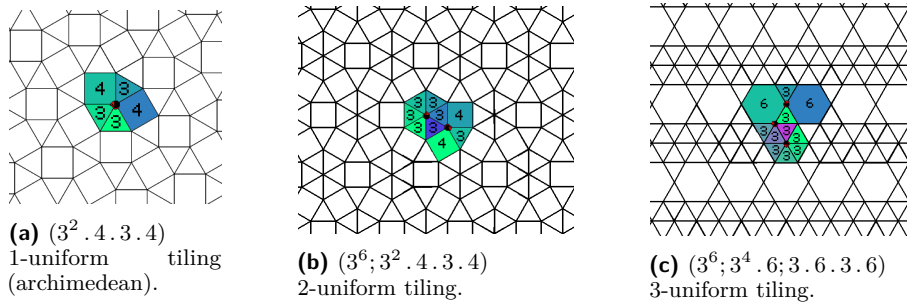


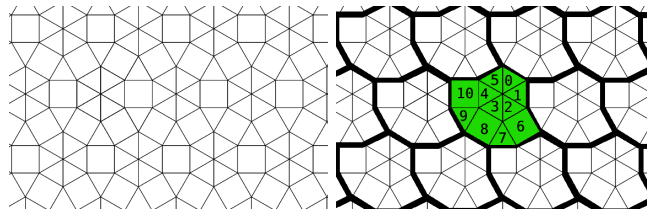
Figure 10 Examples of the naming convention of uniform tilings in the standardized “isogonal vertex type” notation, each point belonging to an orbit describing vertex types around it.

A tiling T can also be represented by its (infinite) dual graph G_T ,¹ where each tile is a vertex of G_T , and two vertices are connected by an edge if the two corresponding tiles are adjacent. When T is a periodic tiling, it is represented by the dual multigraph G_S of its supertile S . For tiles touching the boundary of S , we connect them to the tiles to which they are adjacent in the other copy or copies of the supertile, and mark the dual edges by $A, B, C, \bar{A}, \bar{B}$, or \bar{C} depending on the portion of the boundary they cross, see Figure 12(b). The graph G_S is in fact the quotient of G_T by the action of the symmetries \vec{a} and \vec{b} (also denoted $G_T/\{\vec{a}, \vec{b}\}$). The graph G_S can be used to navigate the tiling T or the graph G_T by updating the representation $\langle(i, j), s\rangle$ when moving to an adjacent tile. The tile s is updated to the adjacent tile s' in G_S , and the coordinates (i, j) need to be updated when crossing a boundary of the supertile S , using the edge marks.

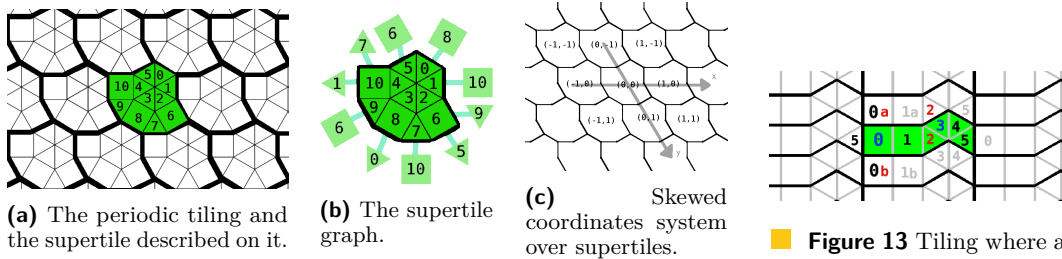
2.2 Rolling Graphs

Let P be a convex polyhedron in \mathbb{R}^3 . We denote by $|P|$ the number of faces of P . The face structure of P can be represented by its dual graph G_P where each face of P is a vertex in G_P and two vertices are connected by an edge if the two corresponding faces of P share an edge (Figure 14).

¹ This can be a multigraph, with parallel edges when two tiles are adjacent on more than one edge; see Figure 13.



■ **Figure 11** The same tiling as Figure 10b ($3^6; 3^2 \cdot 4 \cdot 3 \cdot 4$) in its supertile tiling representation.



(a) The periodic tiling and the supertile described on it.

(b) The supertile graph.

(c) Skewed coordinates system over supertiles.

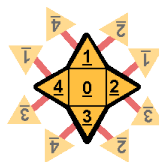
■ **Figure 13** Tiling where a multigraph is necessary; see tiles 3 and 2.

■ **Figure 12** Infinite tiling to supertile multigraph.

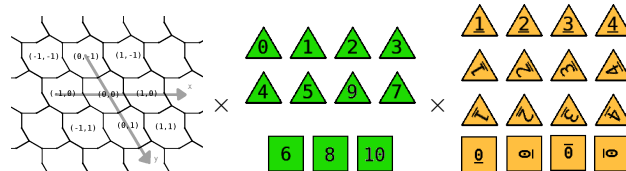
For a face $f \in P$ or a tile $t \in T$, denote by $|f|$ and $|t|$ its number of edges. We number the edges of every face f of polyhedron P counter-clockwise starting from one arbitrary edge that will serve as the reference edge. We do the same for every tile t of the supertile S (and the corresponding tessellation T), with one edge being the reference edge, and the next edges being numbered in clockwise order. A face $f \in P$ is *compatible* with $t \in T$ in the *orientation* o if $|f| = |t|$ and the counter-clockwise sequence of edge lengths and angles in f starting at edge number o matches exactly the clockwise sequence of edge lengths and angles in t starting from the reference edge. This means that f can be placed in the plane with edge number o overlapping with the reference edge of t so that the two polygons overlap perfectly.

We say polyhedron P *rests on the tile* t in the tessellation T with its face f at orientation o if f and t completely overlap and the edge number o of f overlaps the reference edge of t . The *position* of P is then represented by the tuple $\langle t, f, o \rangle$. When T is a periodic tiling with supertile S , and $t = \langle (i, j), s \rangle$ for $s \in S$, then this position can be written as $\langle (i, j), s, f, o \rangle$ (Figure 15). The *state* associated with this position is the tuple $\langle s, f, o \rangle$.

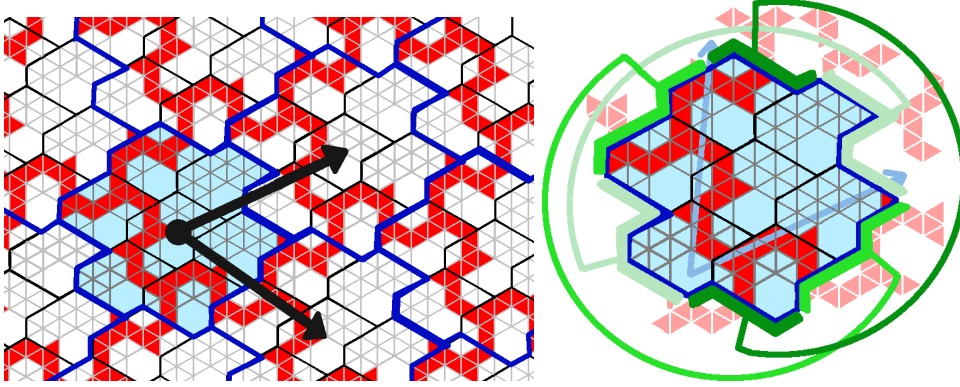
The *rolling graph* $G_{P,T}$ for P and T is an infinite graph whose vertex set is the set of all possible positions $\langle t, f, o \rangle$, and two nodes are connected by an edge if there is a valid roll between them. The positions adjacent to $\langle t, f, o \rangle$ can be easily explored by using the dual graphs of P and T . We write $\langle t, f, o \rangle \sim \langle t', f', o' \rangle$ if the two positions are connected by a path in the rolling graph. In that case, we say that the two positions are *reachable* from one another.



■ **Figure 14** Dual graph of a pyramid with information about the relative orientations of its faces.



■ **Figure 15** A vertex of the rolling graph is composed of $\langle (i, j), (tile, face, orientation) \rangle$



■ **Figure 16** By finding the symmetry vectors in a connected component, we can describe a compact representation of the connected component's periodic graph (over the rolling graph).

2.3 Symmetries of Rolling Graphs

In this section, we show that any large connected subgraph of the rolling graph $G_{P,T}$ has a translational symmetry. We start by bounding the number N of possible states $\langle s, f, o \rangle$ of a rolling graph.

$$\begin{aligned} N &= \sum_{s \in S} \sum_{f \in P} (\text{number of compatible orientations between } f \text{ and } s) \\ &\leq \sum_{s \in S} \sum_{f \in P} |f| \leq 6|S||P|. \end{aligned}$$

The last inequality is by Euler's formula. Note that the rolling graph in itself has the same translational symmetries as the tiling T , because the validity conditions are the same in both positions.

► **Fact 1.** *If $\langle (i, j), s_0, f_0, o_0 \rangle$ has a valid roll to $\langle (i+i_1, j+j_1), s_1, f_1, o_1 \rangle$, then $\langle (i', j'), s_0, f_0, o_0 \rangle$ has a valid roll to $\langle (i' + i_0, j' + j_0), s_1, f_1, o_1 \rangle$ for all $i', j' \in \mathbb{Z}$.*

This however does not mean that the same symmetries apply to the connected components of the rolling graph, that is, $\langle (i, j), s_0, f_0, o_0 \rangle$ and $\langle (i', j'), s_0, f_0, o_0 \rangle$ might not be reachable, even if the connected components are infinite. However, the following lemma shows that if two distinct reachable positions have the same state, then we obtain a translational symmetry on their connected components in the rolling graph.

► **Lemma 1.** *If $\langle (i, j), s, f, o \rangle \sim \langle (i+u, j+v), s, f, o \rangle$, then for all $\langle (i', j'), s', f', o' \rangle \sim \langle (i, j), s, f, o \rangle$, we have $\langle (i', j'), s', f', o' \rangle \sim \langle (i' + u, j' + v), s', f', o' \rangle$.*

That is, $u\vec{a} + v\vec{b}$ defines a translational symmetry on the connected component of $\langle (i, j), s, f, o \rangle$ in the rolling graph.

Proof. Write the path from $\langle (i', j'), s', f', o' \rangle$ to $\langle (i, j), s, f, o \rangle$ in the rolling graph as $\langle (i, j), s, f, o \rangle = \langle (i+i_0, j+j_0), s_0, f_0, o_0 \rangle, \dots, \langle (i+i_k, j+j_k), s_k, f_k, o_k \rangle = \langle (i', j'), s', f', o' \rangle$. Since, by Fact 1, $\langle (i+u+i_\ell, j+u+j_\ell), s_\ell, f_\ell, o_\ell \rangle$ to $\langle (i+u+i_{\ell+1}, j+u+j_{\ell+1}), s_{\ell+1}, f_{\ell+1}, o_{\ell+1} \rangle$ is a valid roll, we can construct the path $\langle (i', j'), s', f', o' \rangle = \langle (i+i_k, j+j_k), s_k, f_k, o_k \rangle, \dots, \langle (i+i_0, j+j_0), s_0, f_0, o_0 \rangle = \langle (i, j), s, f, o \rangle \sim \langle (i+u, j+v), s, f, o \rangle = \langle (i+u+i_0, j+v+j_0), s_0, f_0, o_0 \rangle, \dots, \langle (i+u+i_k, j+v+j_k), s_k, f_k, o_k \rangle = \langle (i' + u, j' + v), s', f', o' \rangle$ ◀

► **Lemma 2.** *There is an algorithm which, in $O(|P||S|)$ time either finds a base of the translational symmetries of the connected component of the rolling graph containing a given position $\langle(i, j), s, f, o\rangle$, or decides that the connected component is of finite size.*

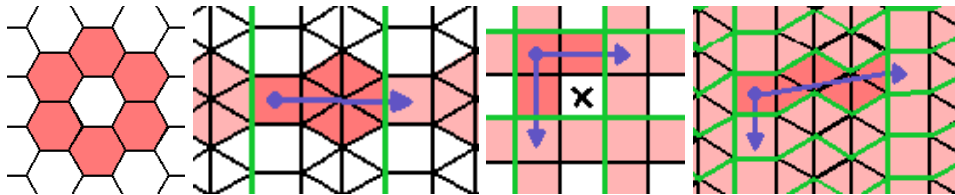
Proof. Run a depth first search on the rolling graph starting from $\langle(i, j), s, f, o\rangle$, for N steps. If the depth first search stops, then the connected component containing $\langle(i, j), s, f, o\rangle$ in the rolling graph is of finite size. Otherwise, by the pigeonhole principle, we have found two positions with the same state. By Lemma 1, we obtain a translational symmetry $u\vec{a} + v\vec{b}$ of the connected component.

Next, factor the rolling graph by this symmetry vector, that is, $G_{P,T}/\{u\vec{a} + v\vec{b}\}$ identifies any pair of positions $\langle(i, j), s, f, o\rangle$ and $\langle(i + ku, j + kv), s, f, o\rangle$ for all $k \in \mathbb{Z}$. Run again a depth first search in $G_{P,T}/\{u\vec{a} + v\vec{b}\}$ starting from $\langle(i, j), s, f, o\rangle$, for N steps. If the depth first search stops, then there are only a finite number of orbits for this symmetry vector, and so only one translational symmetry in this connected component. Otherwise, again by the pigeonhole principle and Lemma 1, we have found a second linearly independent translational symmetry $u'\vec{a} + v'\vec{b}$ for this connected component. ◀

The algorithm in the above lemma finds a basis of two, one or zero translational symmetries in the connected component. We can factor the rolling graph by those symmetries by identifying symmetric tiles. As the symmetries are multiples of the supertile symmetries, this is easily done by performing a coordinate change from the (i, j) coordinates to coordinates in the new basis. When there is no symmetry, the algorithm identifies a bounded connected component in $G_{P,T}$. When there is one symmetry vector, the algorithm finds a finite number of orbits for this symmetry. Finally, when there are two symmetry vectors in the basis, the factored rolling graph $G_{P,T}/\{u\vec{a} + v\vec{b}, u'\vec{a} + v'\vec{b}\}$ is of size polynomial in N and the connected component can be explored completely by depth first search. In all three cases, a compact representation of the connected component has been found. In the two latter cases, it takes the form of a polynomially-sized fundamental domain and one or two translational symmetry vectors.

2.3.1 Results on reachability

The arguments above show how to identify the connected components in the rolling graph. In order to find the set of tiles that can be reached from a starting position, we only need to look at the first part $(i, j), s$ of the positions in the connected component. Because this is a projection, it preserves the symmetry vectors. We obtain the following classification for the reachable area.



■ **Figure 17** No vector, one vector, two vectors but fail to cover, two vectors and full cover.

- If the rolling graph does not have symmetry vectors, the reachable area is bounded and P on T starting at $\langle t, f, o\rangle$, is a *bounded roller*.

- If the rolling graph only has one linearly independent vector, the reachable area is a band and P on T starting at $\langle t, f, o \rangle$ is a 🌀 *band roller*.
- If the rolling graph has two linearly independent vectors, the reachable area extends infinitely in all directions. If not every tile t is present in the reachable supertiles, the reachable tiles forms a plane with holes and P on T starting at $\langle t, f, o \rangle$ is a 🏠 *hollow-plane roller*.
- If every tile t is present in the reachable supertiles, the reachable tiles cover the entire plane and P on T starting at $\langle t, f, o \rangle$ is a 🏠 *plane roller*.

3 Toolbox for Puzzle Designers

As mentioned in the Introduction, a rolling puzzle game typically includes a playing area with obstacles and/or paths, a polyhedron that will navigate that space, a starting position, and a goal position. The starting and/or goal positions sometimes specify a specific polyhedron face to match with a specific tile, in addition to just the tile. Once a polyhedron and a tessellation have been selected, there are several additional properties that can facilitate puzzle design. The rolling graph defined above can also be used to compute them.

3.1 Properties

Unused tiles in the playing area

The first and most crucial piece of information is provided directly by the reachability computed in the previous section. For the puzzle to be solvable, the goal tile should be in the reachable area from the start tile. Also, when the game includes interactive elements, they cannot not be usefully placed on tiles that cannot be reached (except as misdirection).

Unused faces on the polyhedron

For face-matching puzzles, determining which faces of the polyhedron are usable in the puzzle is also important. Some faces might not be compatible with the tiling, while others might not appear in the connected rolling graph despite being compatible. For example, puzzle designers should avoid placing the goal on a polyhedron face that cannot be rolled on. Unused faces of the polyhedron can easily be detected while computing the reachable area.

Guaranteed starting points

When using a plane roller, we must select a starting state (tile, face, and orientation) from which the polyhedron can reach the whole plane. This task can be simplified by selecting a *guaranteed starting point*, which has the property that every tile in the plane is reachable from that starting tile, no matter what polyhedron face and orientation is used as a starting state.

► **Definition 3.** *Given a plane roller pair (P, T) , a tile $t \in T$ is a guaranteed starting point if, for every $f \in P$ with $|f| = |t|$, and for every $o \in f$, we have P on T starting at $\langle t, f, o \rangle$ is a plane roller.*

► **Definition 4.** *Given a rolling pair (P, T) with reachable area RA , a tile $t \in T$ is a guaranteed starting point if, for every $f \in P$ with $|f| = |t|$, for every $o \in f$, and for every $t_i \in RA$, there is a face f_i and orientation o_i such that $\langle t, f, o \rangle \sim \langle t_i, f_i, o_i \rangle$.*

Which faces reach which tiles: face-completeness

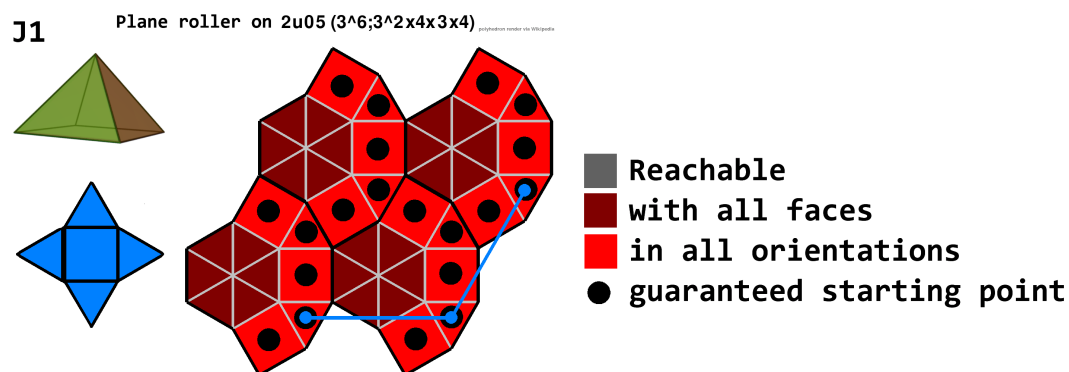
In a face-matching rolling puzzle game, the objective is to reach a specific tile with a specific face on the polyhedron (often marked by a different color). In some cases, not every face of a particular shape can reach every tile. When using a polyhedron/tiling pair in a puzzle game, it can help to know which face can reach which tile. We can track specific tiles that can be reached by every compatible face during our computation. We call such tiles *face-complete* tiles. Refer to Figure 18.

► **Definition 5** (face-complete tile). A rolling pair (P, T) with starting state $\langle t_0, f_0, o_0 \rangle$ has a face-complete tile $t \in T$ if all compatible faces of the polyhedron can roll on t with some orientation, that is, for all $f \in P$ with $|f| = |t|$, there is an orientation o such that $\langle t, f, o \rangle \sim \langle t_0, f_0, o_0 \rangle$.

► **Definition 6** (face-orientation-complete tile). A tile is face-orientation-complete if it can be visited with all compatible faces in every orientation within a connected component.

3.2 Puzzlemaker's Reference Image

We can combine all of the above results into one image that serves as a reference point for puzzlemakers. Figure 18 shows an example. This image allows one to select a tessellation/polyhedron pair very easily depending on the puzzle's needs.



■ **Figure 18** Puzzlemaker's reference image: Left: polyhedron and its used faces (net). Right: Tiling and tile properties.

The full set of reference images can be found on our website: <https://akirabaes.com/polyroll/>.

4 Implementation

The roller classification algorithm was implemented in Python 3.8 and is available on GitHub at <https://github.com/akirabaes/RollingPolyhedron/>. It uses NumPy and SymPy for creating a minimal linearly independent base, and pygame to produce images. The implemented version performs further manipulations, such as aggregating connected rolling graph states grouped by supertile into superstates, to lower processing time and avoid dealing with individual tile positions calculations by only looking at the supertile cartesian coordinates. The result table can be consulted at <https://akirabaes.com/polyroll/resulttable/>.

We defined the supertiles of each tiling by hand in a custom periodic tessellation drawing tool, as we lacked code to automatically convert vertex-type orbits (isohedral, edges) notation to dual-graph supertile (isogonal, tiles) notation, but we did have a list of n -uniform tessellation drawings [7].

An interactive 3D visualization of the rolling logic was implemented by Université libre de Bruxelles Computer Science Bachelor students [3]; see Figure 1.

5 Open Problem

It is left to determine, for the 87 polyhedra out of the 129 considered that did not generate a plane roller with the 131 considered tilings, if there exists a tiling on which they would be able to roll on the 2D plane.

■ **Table 1** Considered polyhedra which did not generate a plane roller with considered tilings.

dodecahedron, truncated cube, truncated octahedron, rhombicuboctahedron, truncated cuboctahedron, snub cube, snub cube c, icosidodecahedron, truncated dodecahedron, truncated icosahedron, rhombicosidodecahedron, truncated icosidodecahedron, snub dodecahedron, snub dodecahedron c, j2, j4, j5, j6, j7, j9, j18, j19, j20, j21, j23, j24, j25, j32, j33, j34, j35, j36, j38, j39, j40, j41, j42, j43, j45, j45 c, j46, j46 c, j47, j47 c, j48, j48 c, j49, j52, j53, j55, j57, j58, j59, j60, j61, j63, j64, j66, j67, j68, j69, j70, j71, j72, j73, j74, j75, j76, j77, j78, j79, j80, j81, j82, j83, j91, j92, triangular prism, pentagonal prism, hexagonal prism, octagonal prism, decagonal prism, dodecagonal prism, pentagonal antiprism, octagonal antiprism, decagonal antiprism, dodecagonal antiprism

References


- 1 Jin Akiyama. Tile-makers and semi-tile-makers. *The American Mathematical Monthly*, 114(7):602–609, 2007.
- 2 Jin Akiyama, Takayasu Kuwata, Stefan Langerman, Kenji Okawa, Ikuro Sato, and Geoffrey C. Shephard. Determination of all tessellation polyhedra with regular polygonal faces. In *International Conference on Computational Geometry, Graphs and Applications*, pages 1–11, 2010.
- 3 Rachel Aouad Albashara, Luca Insisa, Quentin Magron, Dan Ngongo, Dang Phi L Pham, and Simon Yousfi. Rouler des polyèdres sur des tessellations. Université libre de Bruxelles Computer Science Bachelor *Printemps des Sciences* showcase, 2022. URL: <https://akirabaes.com/polyrollly/printempsdessciences2022>.
- 4 Antonio Bicchi, Yacine Chitour, and Alessia Marigo. Reachability and steering of rolling polyhedra: a case study in discrete nonholonomy. *IEEE Transactions on Automatic Control*, 49(5):710–726, 2004.
- 5 Kevin Buchin and Maike Buchin. Rolling block mazes are pspace-complete. *Journal of Information Processing*, 20(3):719–722, 2012. doi:10.2197/ipsjjip.20.719.
- 6 Kevin Buchin, Maike Buchin, Erik D. Demaine, Martin L. Demaine, Dania El-Khechen, Sándor Fekete, Christian Knauer, André Schulz, and Perouz Taslakian. On rolling cube puzzles. In *Proceedings of the 19th Canadian Conference on Computational Geometry (CCCG 2007)*, Ottawa, Canada, August 2007.

6:14 Rolling Polyhedra on Tessellations

- 7 D. Chavey. Tilings by regular polygons – II: A catalog of tilings. *Computers & Mathematics with Applications*, 17(1–3):147–165, 1989.
- 8 Yacine Chitour, Alessia Marigo, Domenico Prattichizzo, and Antonio Bicchi. Rolling polyhedra on a plane, analysis of the reachable set. In *Algorithms for Robotic Motion and Manipulation (WAFR 1996)*, pages 277–285. CRC Press, 1997.
- 9 Euclid. *Elements*, volume I, Circa 300 BCE.
- 10 Judith V. Field. Rediscovering the archimedean polyhedra: Piero della francesca, luca pacioli, leonardo da vinci, albrecht dürer, danielle barbaro, and johannes kepler. *Archive for History of Exact Sciences*, 50(3/4):241–289, 1997.
- 11 Martin Gardner. *Martin Gardner’s Sixth Book of Mathematical Diversions from Scientific American*. W H Freeman, San Francisco, 1971. Chapter 8.
- 12 Martin Gardner. *Mathematical Carnival*. Borzoi / Alfred A Knopf, New York, 1975. Chapter 9, Problem 1: The red-faced cube.
- 13 Martin Gardner. *Time Travel and Other Mathematical Bewilderments*. W H Freeman, New York, 1988. Chapter 9, Problem 8: Rolling cubes.
- 14 B. Grünbaum and N. W. Johnson. The faces of a regular-faced polyhedron. *Journal of the London Mathematical Society*, 1(1):577–586, 1965.
- 15 Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman and Company, 1987.
- 16 Markus Holzer and Sebastian Jakobi. On the complexity of rolling block and Alice mazes. In *Proceedings of the 6th International Conference on Fun with Algorithms*, pages 210–222, 2012.
- 17 Norman W Johnson. Convex polyhedra with regular faces. *Canadian Journal of Mathematics*, 18:169–200, 1966.
- 18 Akihiro Uejima and Takahiro Okada. NP-completeness of rolling dice puzzles using octahedral and icosahedral dices (in Japanese). *IEICE Trans. Fund.*, J94-A(8):621–628, 2011.
- 19 Viktor Abramovich Zalgaller. Convex polyhedra with regular faces. *Zapiski Nauchnykh Seminarov POMI*, 2:5–221, 1967.

A Result Tables

tetrahedron with (3^6) • **cube** with (4^4) • **octahedron** with (3^6) • **icosahedron** with (3^6) • **truncated tetrahedron** with $(3^6; 3^2.6^2)$ • **cuboctahedron** with $(3^2.4.3.4)$, $(3^6; 3^2.4.3.4)$, $(3^3.4^2; 3^2.4.3.4)1$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j1** with $(3^2.4.3.4)$, $(3^6; 3^2.4.3.4)$, $(3^3.4^2; 3^2.4.3.4)1$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j3** with $(3^6; 3^2.4.3.3.4; 3.4^2.6)$, $(3^6; 3^2.4.3.4; 3.4^2.6; 3.4.6.4)$ • **j8** with (4^4) , $(3^6; 3^3.4^2; 4^4)1$, $(3^6; 3^3.4^2; 4^4)3$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j10** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j11** with (3^6) • **j12** with (3^6) • **j13** with (3^6) • **j14** with $(3^6; 3^3.4^2)1$ • **j15** with $(3^6; 3^3.4^2)1$ • **j16** with $(3^6; 3^3.4^2)1$ • **j17** with (3^6) • **j22** with $(3^6; 3^4.6)1$, $(3^6; 3^4.6; 3.6.3.6)2$, $(3^6; 3^4.6; 3.6.3.6)3$, $(3^6; 3^6; 3^4.6^2)$ • **j26** with $(3^2.4.3.4)$, $(3^3.4^2; 3^2.4.3.4)2$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j27** with $(3^3.4^2)$, $(3^3.4^2; 3^2.4.3.4)1$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^3.4^2; 3^2.4.3.4; 3^2.4.3.4)$ • **j28** with $(3^3.4^2)$, $(3^3.4^2; 4^4; 4^4)1$ • **j29** with $(3^2.4.3.4)$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j30** with $(3^3.4^2)$ • **j31** with $(3^2.4.3.4)$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j37** with (4^4) • **j44** with $(3^6; 3^2.4.3.4; 3^2.4.3.4)$, $(3^3.4^2; 3^2.4.3.4; 3^2.4.3.4)$ • **j44 chiral** with $(3^6; 3^2.4.3.4; 3^2.4.3.4)$ • **j50** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j51** with (3^6) • **j54** with $(3.4.6.4)$ • **j56** with $(3.4.6.4)$ • **j62** with (3^6) • **j65** with $(3.6.3.6)$ • **j84** with (3^6) • **j85** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^3.4^2)1$, $(3^6; 3^3.4^2; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j86** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 4^4)1$, $(3^6; 3^3.4^2; 4^4)2$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j87** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **j88** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 4^4)1$, $(3^6; 3^3.4^2; 4^4)2$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^3.4^2)1$, $(3^6; 3^3.4^2; 3^3.4^2)2$ • **j89** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 4^4)3$, $(3^6; 3^3.4^2; 4^4)4$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^3.4^2)1$, $(3^6; 3^3.4^2; 3^3.4^2)2$ • **j90** with (3^6) , $(3^6; 3^3.4^2)1$, $(3^6; 3^3.4^2)2$, $(3^6; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 4^4)1$, $(3^6; 3^3.4^2; 4^4)2$, $(3^6; 3^6; 3^3.4^2)1$, $(3^6; 3^6; 3^3.4^2)2$, $(3^6; 3^3.4^2; 3^3.4^2)1$, $(3^6; 3^3.4^2; 3^3.4^2)2$, $(3^6; 3^2.4.3.4; 3^2.4.3.4)$, $(3^6; 3^3.4^2; 3^2.4.3.4; 4^4)$ • **square antiprism** with $(3^3.4^2)$ • **hexagonal antiprism** with $(3^4.6)$, $(3^6; 3^4.6)1$, $(3^6; 3^4.6)2$, $(3^4.6; 3^2.6^2)$, $(3^6; 3^4.6; 3^2.6^2)2$, $(3^6; 3^4.6; 3.6.3.6)1$, $(3^6; 3^4.6; 3.6.3.6)2$, $(3^6; 3^4.6; 3.6.3.6)3$, $(3^6; 3^6; 3^4.6^2)$, $(3^6; 3^4.6; 3^4.6)$, $(3^4.6; 3^4.6; 3.6.3.6)1$, $(3^4.6; 3^4.6; 3.6.3.6)2$, $(3^6; 3^4.6; 3^2.6^2; 3.6.3.6)$, $(3^4.6; 3^2.6^2; 3^2.6^2; 3.6.3.6)$

■ **Table 2**  Plane-roller polyhedra and tilings (42 polyhedra and 145 pairings).

6:16 Rolling Polyhedra on Tessellations

tetrahedron (x7) • octahedron (x7) • icosahedron (x7) • truncated tetrahedron (x12)
 • cuboctahedron (x4) • truncated cube • truncated octahedron (x10) •
 rhombicuboctahedron (x8) • truncated cuboctahedron (x6) • snub cube (x13)
 • snub cube chiral (x12) • truncated icosahedron (x4) • rhombicosidodecahedron (x4)
 • truncated icosidodecahedron (x5) • snub dodecahedron (x5) •
 snub dodecahedron chiral (x4) • j1 (x6) • j3 (x8) • j7 (x5) • j8 • j10 (x15) •
 j11 (x6) • j12 (x7) • j13 (x7) • j14 (x15) • j15 (x15) • j16 (x15) • j17 (x7) • j18 (x7) •
 j19 (x4) • j22 • j26 (x2) • j27 (x13) • j28 (x8) • j29 (x2) • j30 (x6) • j31 • j35 (x11) •
 j37 (x3) • j38 (x7) • j44 (x6) • j44 chiral (x5) • j45 (x2) • j45 chiral (x2) • j49 (x8)
 • j50 (x16) • j51 (x7) • j53 (x6) • j54 (x14) • j55 (x10) • j56 (x16) • j57 (x14) •
 j62 (x4) • j65 (x3) • j66 • j72 (x4) • j74 (x10) • j75 (x6) • j76 (x4) • j78 (x4) •
 j79 (x6) • j81 (x4) • j84 (x7) • j85 (x16) • j86 (x16) • j87 (x18) • j88 (x18) • j89 (x20)
 • j90 (x16) • triangular prism (x12) • hexagonal prism (x18) • octagonal prism
 • dodecagonal prism (x4) • square antiprism (x5) • hexagonal antiprism (x2) •
 dodecagonal antiprism (x2)

■ **Table 3** 🛠️ Hollow-plane-roller polyhedra and tilings (76 polyhedra and 588 pairings).

tetrahedron (x35) • cube (x41) • octahedron (x35) • icosahedron (x35) •
 truncated tetrahedron (x34) • cuboctahedron (x3) • truncated octahedron (x15)
 • rhombicuboctahedron (x43) • snub cube (x7) • snub cube chiral (x8) •
 truncated icosahedron (x13) • rhombicosidodecahedron (x2) • snub dodecahedron (x7)
 • snub dodecahedron chiral (x7) • j1 (x7) • j3 (x2) • j7 (x43) • j8 (x39) • j9 (x42) •
 j10 (x29) • j11 (x31) • j12 (x35) • j13 (x35) • j14 (x42) • j15 (x42) • j16 (x42) • j17 (x35)
 • j18 (x43) • j19 (x41) • j20 (x42) • j21 (x42) • j22 (x30) • j23 (x30) • j24 (x30) • j25 (x30)
 • j26 (x7) • j27 (x17) • j28 (x46) • j29 (x4) • j30 (x15) • j31 (x3) • j35 (x44) • j36 (x49) •
 j37 (x46) • j38 (x44) • j39 (x47) • j40 (x42) • j41 (x42) • j42 (x42) • j43 (x42) • j44 (x32)
 • j44 chiral (x33) • j45 (x31) • j45 chiral (x32) • j46 (x32) • j46 chiral (x32) • j47 (x30) •
 j47 chiral (x30) • j48 (x30) • j48 chiral (x30) • j49 (x20) • j50 (x27) • j51 (x35) • j54 (x8)
 • j55 (x10) • j56 (x15) • j57 (x11) • j62 (x17) • j65 (x25) • j67 • j72 • j73 (x5) • j76 •
 j77 (x5) • j80 (x5) • j84 (x35) • j85 (x32) • j86 (x27) • j87 (x28) • j88 (x23) • j89 (x21)
 • j90 (x25) • triangular prism (x45) • pentagonal prism (x42) • hexagonal prism (x36)
 • octagonal prism (x42) • decagonal prism (x42) • dodecagonal prism (x43) •
 square antiprism (x37) • pentagonal antiprism (x30) • hexagonal antiprism (x40) •
 octagonal antiprism (x30) • decagonal antiprism (x30) • dodecagonal antiprism (x30)

■ **Table 4** 🛠️ Band-roller polyhedra and tilings (94 polyhedra and 2623 pairings).