

Finding a Hamiltonian Path in a Cube with Specified Turns is Hard

ZACHARY ABEL^{1,a)} ERIK D. DEMAINE^{2,b)} MARTIN L. DEMAINE^{2,c)}
 SARAH EISENSTAT^{2,d)} JAYSON LYNCH^{2,e)} TAO B. SCHARDL^{2,f)}

Received: xx xx, xxxx, Accepted: xx xx, xxxx

Abstract: We prove the NP-completeness of finding a Hamiltonian path in an $N \times N \times N$ cube graph with turns exactly at specified lengths along the path. This result establishes NP-completeness of *Snake Cube* puzzles: folding a chain of N^3 unit cubes, joined at face centers (usually by a cord passing through all the cubes), into an $N \times N \times N$ cube. Along the way, we prove a universality result that zig-zag chains (which must turn every unit) can fold into any polycube after $4 \times 4 \times 4$ refinement, or into any Hamiltonian polycube after $2 \times 2 \times 2$ refinement.

Keywords: Hamiltonian path, NP-Completeness, puzzle, Snake Cube puzzle

1. Introduction

Snake Cube puzzles [6–8] are physical puzzles consisting of a chain of unit cubes, typically made out of wood or plastic, with holes drilled through to route an elastic cord. The cord holds the cubes together, at shared face centers where the cord exits/enters the cubes, but permits the cubes to rotate relative to each other at those shared face centers. **Fig. 1** shows photographs of a wooden Snake Cube from its initial to solved state. As in most physically existing Snake Cube puzzles, it consists of 27 cubes and the goal is to make a $3 \times 3 \times 3$ cube.

The origin of Snake Cube puzzles seems to be unknown; the simplicity of the idea may have led to many copies and variations. Singmaster’s puzzle collection [9] lists his earliest purchase of such a puzzle as 1990 in Paris. Jaap’s Puzzle Page [8] lists two tradenames for different versions of the puzzle: Kev’s Kubes (wooden, made by Trench Puzzles), and Cubra Finger Twisting Puzzler (plastic, made by Parker Hilton Ltd circa 1998 [5], later acquired by Falcon Games) which has five color-coded variations—Mean Green, Bafflin’ Blue, Twist yer ’ead Red, ’Orrible Orange, and Puzzlin’ Purple. **Fig. 2** shows a packaged Cubra Puzzlin’ Purple. The two Snake Cube puzzles we own (including Fig. 1) happen to match the geometry of the Cubra Bafflin’ Blue puzzle.

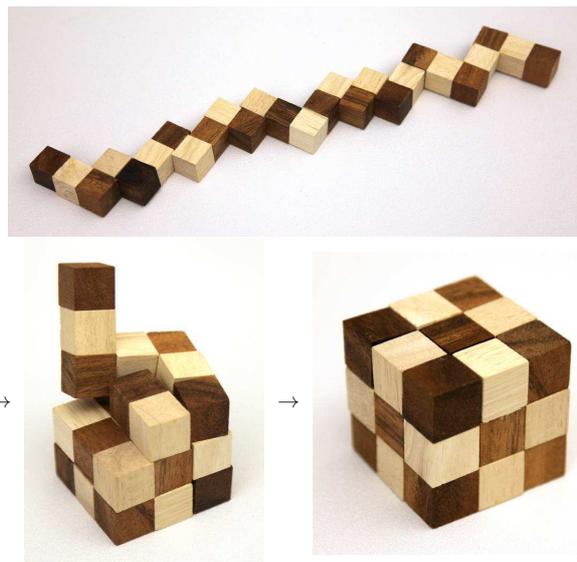


Fig. 1 A commercially available Snake Cube puzzle consisting of 27 cubes in the pattern SSTSSTSTTTTSTTTTSTTTTSS.



Fig. 2 The commercially available Cubra Puzzlin’ Purple puzzle consisting of 27 cubes in the pattern STTTTSTTTTSTTTTSTTTTSTTTTSS. Photo from the Slocum Collection [5] courtesy of The Lilly Library, Indiana University, Bloomington, Indiana.

¹ MIT Department of Mathematics, 77 Massachusetts Ave., Cambridge, MA 02139, USA

² MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA

a) zabel@math.mit.edu

b) edemaine@mit.edu

c) mdemaine@mit.edu

d) seisenst@mit.edu

e) jaysonl@mit.edu

f) neboat@mit.edu

Larger $4 \times 4 \times 4$ puzzles are sold under the name “King Snake” [1]. A custom $5 \times 5 \times 5$ puzzle was built at the Smith College Mathematics Department [6].

1.1 Our results

In this paper, we study the natural generalization of Snake Cube puzzles to a chain of N^3 cubes whose goal is to fold into an $N \times N \times N$ cube. The puzzle can be specified by a sequence of N^3 binary symbols (S or T), each representing a cube whose cord either passes straight through the cube and thus prevents the chain from turning (S), or turns 90° at the cube and thus forces the chain to turn (T).

We prove that it is NP-complete to decide whether such a puzzle has a folded state in the shape of an $N \times N \times N$ cube. Our NP-hardness reduction is from the classic 3-PARTITION problem, whose goal is to divide $3n$ integers, v_1, \dots, v_{3n} , into n triples of equal sum. We represent each integer v_i as a sequence of cubes that must effectively fold into a narrow “peg” of width proportional to v_i . In sections 4 and 5, we build an infrastructure for these pegs to fit into, consisting of n “slots” whose widths are proportional (with the same constant as pegs) to the target sum of each triple. Finally, in section 8, we connect the infrastructure and pegs together with “filler” that lets the pegs move freely relative to each other and the infrastructure.

To develop “filler” gadgets for our hardness proof, we also prove universality results about zig-zag puzzles, which consist solely of turning cubes (T). Specifically, in section 6, we show that the zig-zag puzzle of $(4N)^3$ unit cubes can fold into the shape of any N -cube polycube after $4 \times 4 \times 4$ refinement. Furthermore, in the special case of Hamiltonian polycubes (whose dual graph has a Hamiltonian path), we show how to reduce the required refinement to just $2 \times 2 \times 2$.

1.2 Related work

A few research papers have already been motivated by Snake Cube puzzles. Scherphuis [8] wrote a computer program to exhaustively enumerate all 11,487 possible $3 \times 3 \times 3$ Snake Cube puzzles, and found that 3,658 have unique solutions. (For example, the puzzle in Fig. 1 has a unique solution, but the puzzle in Fig. 2 has six solutions.) Ruskey and Sawada [7] characterized when an MNK -cube zig-zag puzzle can fill an $M \times N \times K$ box, and generalized this result to d dimensions and toroidal space. McDonough [6] analyzed cube Hamiltonian paths that form nontrivial knots, and multiple spanning paths that form nontrivial links.

2. Definitions

For convenience, we work on the standard, Cartesian coordinate system, identifying the positive x -, y -, and z -directions with right, forward, and up, respectively. We divide space into unit cubes, which we call *cells*, according to the integer lattice, \mathbb{Z}^3 . Each cell is identified by its left, back, bottom corner; in other words, the *cell at* (x, y, z) has opposite corners (x, y, z) and $(x + 1, y + 1, z + 1)$. A cell has six faces, and we identify co-located faces on adjacent

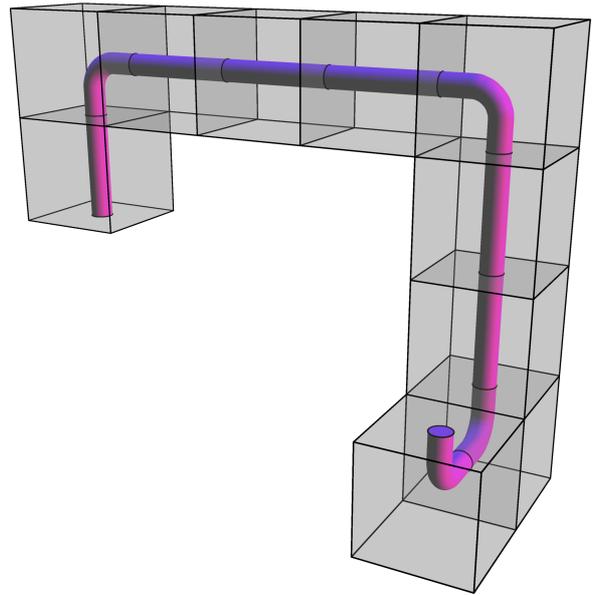


Fig. 3 A depiction of the puzzle $STS^3TS^2T^2$. In terms of real Snake Cube puzzles, the blue and purple pipe illustrates the cord inside the cubes to distinguish straight cubes from turn cubes.

cells. For example, we identify the right face of cell (x, y, z) with the left face of cell $(x + 1, y, z)$.

An (*abstract*) *puzzle* of cubes is a sequence of Ss and Ts, representing *straight* cells and *turn* cells, respectively. A *configuration* of a length- n abstract puzzle is a sequence of distinct cells c_1, \dots, c_n and faces f_0, \dots, f_n , such that

- for each $1 \leq i \leq n$, f_{i-1} and f_i are distinct faces of cell c_i , and
- if the i th symbol in the puzzle is an S then f_{i-1} and f_i are opposite faces on cell c_i ; if this symbol is a T, faces f_{i-1} and f_i are adjacent.

In other words, such a configuration is a *path of cells* whose straight and turn cells correspond to the specification of the abstract puzzle. Faces f_0 and f_n are called the *starting face* and *ending face* of the configuration, respectively.

A puzzle may be notated simply as a sequence of symbols S and T and superscripts to indicate symbol repetition. For example, Fig. 3 shows the puzzle $STS^3TS^2T^2$. We introduce a *run-length encoding* shorthand as follows: For positive integers d_1, \dots, d_r , we define

$$\langle d_1, \dots, d_r \rangle = S^{d_1-1}TS^{d_2-2}T \dots TS^{d_{r-1}-2}TS^{d_r-1}$$

if $r > 1$ and $\langle d_1 \rangle = S^{d_1}$; the integers d_i describe the lengths of the *bars*, or maximal subsets of collinear cubes. For example, the puzzle in Fig. 3 is abbreviated as $\langle 2, 5, 4, 2, 1 \rangle$. Note that 1s may appear in this notation only at the start or end of the sequence, and, unless $r = 1$, they indicate that the first or last symbol of the puzzle is a T. Two notable examples are $\langle 1 \rangle = S$ and $\langle 1, 1 \rangle = T$.

For two puzzles C_1 and C_2 of n and m cubes respectively, we define their *concatenation* $C_1 \circ C_2$ as the puzzle of $n + m$ cubes obtained by concatenating the underlying S-T sequences of C_1 and C_2 . In run-length encoding notation,

$$\langle d_1, \dots, d_r \rangle \circ \langle e_1, \dots, e_s \rangle = \langle d_1, \dots, d_{r-1}, d_r + e_1, e_2, \dots, e_s \rangle.$$

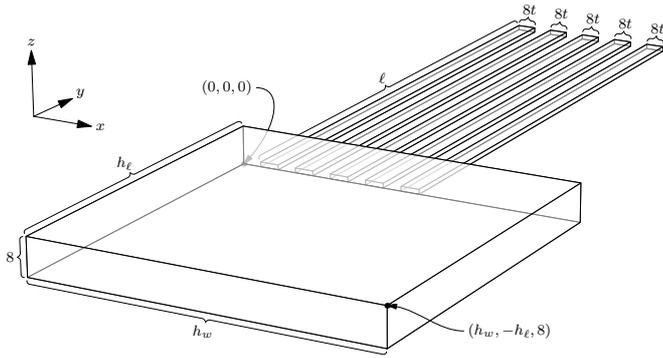


Fig. 4 A depiction of the different parts of the hub-and-slots shape. Not drawn to scale.

For example, if $C_1 = \langle 1, 3, 4, 1 \rangle = \text{TSTS}^2\text{T}$ and $C_2 = \langle 5, 6 \rangle = \text{S}^4\text{TS}^5$, then $C_1 \circ C_2 = \text{TSTS}^2\text{TS}^4\text{TS}^5 = \langle 1, 3, 4, 6, 6 \rangle$. A notation for *iterated concatenation* is also useful: For a puzzle C , the notation C^k means $C \circ C \circ \dots \circ C$, with k total copies of C .

Finally, we reduce from the 3-PARTITION problem, defined as follows. An instance of this problem is a (multi)set $V = \{v_1, \dots, v_{3n}\}$ of positive integers such that $\frac{1}{n} \sum_{i=1}^{3n} v_i = t$ is an integer. We require additionally that $t/4 < v_i < t/2$ for each i . A *3-partition* of V is a partition of V into n groups each with sum equal to the *target sum*, t ; each group necessarily has three elements. The instance V is a YES instance if and only if it has a 3-partition. The 3-PARTITION problem was shown to be strongly NP-hard by Garey and Johnson [4].

3. Overview

We first provide an informal description of the reduction that also serves as an outline for the remainder of the paper.

From a 3-PARTITION instance $V = \{v_1, \dots, v_{3n}\}$ with target sum t , we will construct a puzzle $R = R(V)$ that exactly fills a $u \times u \times u$ cube (where $u = \text{poly}(n, t)$, specified later) if and only if there exists a 3-partition for V .

The key idea for our reduction is as follows. Imagine a large box—the “hub”—with n separate, long and skinny “slots” sticking out of it, each with width $8t$, length l , and height 1. **Fig. 4** illustrates this “hub-and-slots” shape. Now suppose that we have $3n$ separate puzzles, or “pegs”: for each $1 \leq i \leq 3n$, form the puzzle $P_i = \langle \ell, 2, \ell, 2, \dots, \ell \rangle$, where there are $8v_i$ bars of length ℓ . If ℓ is long enough, then for any peg P_i to fit into the hub-and-slots, it must in fact zig-zag along one of the slots, occupying $8v_i$ of its width (possibly partially poking into the hub). It follows that all pegs P_1, \dots, P_{3n} can fit without overlap into the hub-and-slots if and only if V has a 3-partition.

Our reduction exactly sets up this situation. Specifically, from the 3-PARTITION instance V , our reduction constructs the puzzle

$$R = R(V) = R_1 \circ R_2 \circ R_3 \circ R_4, \quad (1)$$

where each R_i solves a specific, isolated task.

The first two portions, R_1 and R_2 , “carve out” the hub-and-slots shape from the $u \times u \times u$ cube. In other words, no

matter how $R_1 \circ R_2$ is positioned inside the $u \times u \times u$ cube, it must fill everything except the exact hub-and-slots shape needed. More specifically, puzzle R_1 (Section 4) carves out an $a \times b \times c$ box (with carefully chosen dimensions $a < b < c$) from the $u \times u \times u$ cube, and then R_2 (Section 5) carves out the hub-and-slots shape from this box.

Puzzle R_3 (Section 7) includes the pegs P_1, \dots, P_{3n} described above, so $R_1 \circ R_2 \circ R_3$ can fit into the $u \times u \times u$ cube only when there exists a 3-partition. Inside puzzle R_3 , the pegs P_i are separated by zig-zagging “filler material” (analyzed in Section 6) that is sufficiently flexible to allow each peg to be independently placed in the slots. This ensures that, when a 3-partition of V exists, puzzle R_3 can indeed fit into the hub-and-slots.

Finally, more zig-zagging filler material comprises puzzle R_4 (Section 8). Its purpose is to fill all space inside the hub that was not filled by R_3 . This ensures that, when a 3-partition exists, the puzzle $R_1 \circ R_2 \circ R_3 \circ R_4$ can fill all cells in the $u \times u \times u$ cube without gaps.

4. Cube to Box

We first formalize the notion of “carving out,” or *excising* a region from a larger region as described in the Overview (Section 3). This allows us to analyze different sections of R separately, for example, arguing that R_1 excises a box from the $u \times u \times u$ cube and therefore the R_2 portion of $R_1 \circ R_2$ must operate entirely inside this box. Since we wish to analyze R_1 and R_2 separately, some care must be taken to ensure that their endpoints join properly.

Definition 1. A *region* is a face-connected collection of cells in the unit grid. A *nontrivial mark* on a region S is a pair (c, f) , where c is a cell in S and f is a face of c . A puzzle configuration in S *starts/ends at mark* (c, f) if its first/last cell-and-face pair is (c, f) . For convenience, we also introduce a *trivial mark*, notated by (\emptyset, \emptyset) , and we declare that any puzzle configuration in S *vacuously starts/ends at the trivial mark*. A *marked region* is a region with a mark, which may be trivial or nontrivial.

The trivial mark is simply a notational convenience: it allows uniform treatment of regions with a chosen start or end location (nontrivially marked regions) and regions with no preferred start or end (trivially marked regions). Also, though it is not required, face f of a nontrivial mark will usually be a boundary face of S . Such marks may be specified from face f alone, because cell $c \in S$ is then uniquely determined.

Suppose we have two marked regions S and S' , with marks (c, f) and $(c', f') \neq (\emptyset, \emptyset)$ respectively, where f' is on the boundary of S' . We would like to say that a puzzle C “excises” S' from S if every configuration of C inside S starting at (c, f) must fill all of S except for a region congruent to S' and must end at face f' of this copy of S' . This is not quite the condition we need: for example, if puzzle C ends with a T cube, then even if the configuration leaves an S' -shaped hole, the last cube may be turned away from S' . We disallow this by requiring that the configuration be “extensible”

inside S , in such a way that this extension must enter cell c' via face f' .

Definition 2. Let S and S' be two (trivially or nontrivially) marked regions with marks (c, f) and (c', f') respectively, where f' is on the boundary of S' if $(c', f') \neq (\emptyset, \emptyset)$. A puzzle C *excises* S' from S if, for every configuration of C inside S starting at (c, f) and ending at some cell-and-face pair (c'', f'') in S , if the cell adjacent to f'' other than c'' is in S and not filled by the configuration, then the unfilled portion of S must be congruent to S' in such a way that, if $(c', f') \neq (\emptyset, \emptyset)$, then $f'' = f'$. Furthermore, at least one such configuration of C must exist.

At least in the case where S' is nontrivially marked, the following fact (which follows directly from the previous definition) shows that such an excision allows perfect separation of matters inside S' from those outside it:

Theorem 3. If C excises marked region S' with nontrivial mark (c', f') from marked region S (with mark (c, f)), and if C' is any other nonempty puzzle, then $C \circ C'$ can be configured inside S starting at (c, f) if and only if C' can be configured inside S' starting at (c', f') .

With these definitions in place, the following two lemmas show how to excise an arbitrary box from an appropriately chosen cube. These lemmas use puzzles called *tribars*, each of which consists of three consecutive bars of length greater than 2. A *horseshoe configuration* of a tribar is one where the three bars lie in a plane and the two parallel bars point in opposite directions, thus forming a “horseshoe” or “U” shape.

Lemma 4. Given integers $a < b < c$, let B be an $a \times b \times c$ box, marked at a unit square in the corner of an $a \times c$ face of the box. Let B' be an $a \times b' \times (c - 1)$ box satisfying $a < b' < (c - 1)$ and $2(b' + 2) > b$, marked at a unit square in the corner of an $a \times (c - 1)$ face of the box. Then there exists a puzzle C excising B' from B .

Proof: We construct C by concatenating a “filler” puzzle to a sequence of tribar puzzles. The *filler* puzzle is a puzzle $\langle c, 2, c, 2, \dots, c, 1 \rangle$. The relevant *tribar puzzle* in this proof has shape $\langle 1, c, b' + 2, c, 1 \rangle$. Let us first consider the behavior of the filler puzzle in B and then examine the behavior of the tribar puzzles.

A filler puzzle must lie inside B with its length- c bars along the long axis of B . The filler puzzle can fill a box of size $a \times (b - b' - 2) \times c$ with $a(b - b' - 2)$ length- c bars if the long bars are laid next to each other in $a \times c \times 1$ -size layers. The filler puzzle alone does not, however, excise an $a \times (b' + 2) \times c$ box from B , because the puzzle may adopt other configurations in B . Hence, we use the additional structure imposed by the tribar puzzles to excise B' from B .

A tribar $\langle 1, c, b' + 2, c, 1 \rangle$ must lie inside B with its length- c bars along the long axis of B . Because $b' + 2 > a$, the length- $(b' + 2)$ bar must lie along either the length- c or length- b axis of B , and because it lies in a different direction from the length- c bars, it must lie along the length- b axis of B . This tribar must then be in a horseshoe configuration.

In a sequence of tribars as above, the subsequent horseshoe configurations must lie on top of each other, since two horseshoe configurations cannot lie side-by-side in B because $2(b' + 2) > b$. Thus, from box B , a sequence of a tribars cuts out two or three rectangular regions. One region (inside the horseshoes) has dimensions $a \times b' \times (c - 1)$, and the final cube in the sequence of tribars must be a turn cube in the corner of an $a \times (c - 1)$ face of this region. The remaining two regions, meanwhile, must have total volume $a \times (b - b' - 2) \times c$.

Prepending a filler puzzle with $a(b - b' - 2)$ length- c bars to a sequence of a tribars as above produces a puzzle where, inside box B , the filler puzzle must fill an $a \times (b - b' - 2) \times c$ box. Hence, the puzzle C consisting of a filler puzzle of $a \times (b - b' - 2)$ length- c bars concatenated to a sequence of a tribar puzzles excises box B' from B . \square

Lemma 5. Given integers $a < b < c$ with $c - b \geq 2$, let B be an $a \times b \times c$ box marked at a unit square in the corner of an $a \times c$ face of the box. Then there exist $u < 4c$ and a puzzle C that excises B from an unmarked $u \times u \times u$ cube U .

Proof: Our first step is to excise a $(u - 1) \times (u - 3) \times a$ box from U . To excise this shape, we first use a sequence of tribars $H = \langle 1, u, u, u, 1 \rangle$. Consider the possible ways to arrange the puzzle $H^2 = H \circ H = \langle 1, u, u, u, 2, u, u, u, 1 \rangle$ inside the cube. Without loss of generality, the first two edges of H^2 must lie in a plane P parallel to the bottom face of the box, with the first bar pointing forward and the second pointing right.

For the sake of contradiction, suppose that the third bar of H^2 lies perpendicular to P . This implies that the end of the third bar is in either the right, forward, upper corner or the right, forward, lower corner of U , and hence, no matter which direction the subsequent length-2 bar points, the other end of the length-2 bar cannot be in a corner. The second H must therefore lie in a plane perpendicular to P , and all four such configurations intersect one of the bars already placed in U . Hence the first three bars of H^2 must all lie in P , forming a horseshoe configuration where the end of the third bar lies along the front right edge of the cube.

Now consider how the remainder of H^2 can be arranged. If the length-2 bar points left, then both of the possible arrangements of the subsequent H intersect one of the bars that has already been placed. The length-2 bar must therefore point up or down, and without loss of generality, let this bar point up. In this case the subsequent H forms a horseshoe that lies on top of the first horseshoe in either the same arrangement or a similar arrangement rotated by 90° .

Suppose now that we extend puzzle H^2 to H^u . By a similar argument, H^u must adopt an arrangement in the cube in which each H forms a horseshoe, and the set of H s together form a stack of u horseshoes in the cube, possibly with different rotations. We say that a stack of horseshoes is *orderly* if all horseshoes have the same rotation.

To complete this first excision and guarantee that the

stack of horseshoes formed from packing H^u into U are orderly, we concatenate onto H^u a “filler” puzzle followed by a sequence of tribar puzzles $I = \langle 1, u, u - 1, u, 1 \rangle$. Consider concatenating the **filler** puzzle $F = \langle 1, u, 1 \rangle^{(u-1)(u-2-a)}$ followed by the sequence I^a to H^u . Notice that the puzzle $H^u \circ \langle 1, u, 1 \rangle$ must be packed into U with the final length- u bar perpendicular to the length- u bars in H^u , and thus concatenating $\langle 1, u, 1 \rangle$ to H^u forces the horseshoes formed from packing H^u to be orderly. Hence, let us assume that the horseshoes in H^u are orderly and focus on the behavior of $F \circ I^a$ inside a $u \times (u - 1) \times (u - 2)$ box X marked at a corner of a $u \times (u - 1)$ face. This analysis is very similar (though not identical) to that in the proof of Lemma 4.

First, consider packing I^a into the volume of X . To pack I^a inside X , each length- u bar in I^a must lie parallel to the length- u edge of X . Because the length- $(u - 1)$ bars in I^a cannot also lie parallel to the length- u edges of X , each such bar must lie in the direction with extent $(u - 1)$. Consequently, each tribar I in X must be packed in a horseshoe configuration, and the resulting stack of horseshoes must be orderly.

Now, consider packing $F \circ I^a$ into X . To pack F into X , each length- u bar of F must lie in the direction with extent u . Furthermore, in order to pack I^a into X afterwards, F must be packed to fill a $(u - 2 - a) \times (u - 1) \times u$ volume that shares a $(u - 1) \times u$ face with X . Finally, the last cube in the puzzle I^a must lie adjacent to a corner of an $a \times (u - 1)$ face of the remaining volume. Let X_0 denote the $a \times (u - 3) \times (u - 1)$ empty space left after packing $F \circ I^a$ into X , and consider X_0 marked at a corner of an $a \times (u - 1)$ face.

To complete the proof, we excise the desired $a \times b \times c$ box B from X_0 using repeated applications of Lemma 4. For $i > 0$, let X_i denote the box of dimensions $a \times b_i \times c_i$ that results from applying Lemma 4 to X_{i-1} with $b_i = \max(\lceil b_{i-1}/2 \rceil, b)$. Notice that the markings on region X_i are compatible with the markings on X_{i-1} . At most $r \leq \lceil \lg \frac{u-3}{b} \rceil + 1$ applications of Lemma 4 are needed to obtain $b_r = b$ and thus to excise an $a \times b \times (u - 1 - r)$ box X_r from X_0 . Choosing u such that $c = u - 1 - r$ completes the proof. \square

The first part of the reduction is as follows:

Definition 6. From an instance $V = \{v_1, \dots, v_{3n}\}$ with target sum t , define $(a, b, c) = (8, 120tn + 1, 240tn + 9)$, and define $R_1(V)$ as the puzzle excising an $a \times b \times c$ box (marked at a corner of an $a \times c$ face) from a $u \times u \times u$ cube as in Lemma 5, where $u \leq 4 \cdot (240tn + 9)$ is as guaranteed in the lemma.

5. Hub and Slots

We now describe the precise measurements of the hub-and-slots shape, illustrated in Fig. 4, and show how to excise it from an appropriately-chosen box.

Definition 7. Fix a 3-PARTITION instance $V = \{v_1, \dots, v_{3n}\}$ with target sum t . The shape of the **hub** is

an $h_w \times h_\ell \times 8$ box, where $h_w = 120tn$ and $h_\ell = h_w + 4$. For convenience, we introduce a coordinate system so that the hub is situated with opposite corners at $(0, 0, 0)$ and $(h_w, -h_\ell, 8)$. There are n **slots**, each an $8t \times \ell \times 1$ box where $\ell = h_\ell + 4$, with the i th slot having opposite corners at $(16i + 8t(i - 1), 0, 0)$ and $(16i + 8ti, \ell, 1)$. In particular, the slots are spaced 16 units apart adjacent to the bottom of the front face of the hub, with width-16 padding on the left and width- $(h_w - 8tn - 16n)$ padding on the right. We refer to the union of these box shapes as the **hub-and-slots** shape.

For technical reasons, we also discuss an **augmented hub-and-slots** shape. This shape is obtained by adjoining to the hub-and-slots shape the box with opposite corners at $(-1, -h_\ell, 1)$ and $(0, 0, 8)$. Equivalently, this shape is the result of widening the hub by one unit in the negative x direction and then removing a $1 \times h_\ell \times 1$ box from the bottom-left edge.

Puzzle R_1 , as defined in Section 4, excises a box of dimensions $8 \times (h_w + 1) \times (h_\ell + \ell + 1)$ from a cube. We next show how to excise the augmented hub-and-slots shape from a box of these dimensions:

Lemma 8. Let B be the box with opposite corners at $(-1, -h_\ell, 0)$ and $(h_w, \ell + 1, 8)$ marked at the left face of cell $(-1, -h_\ell, 0)$. Let S' be the trivially-marked augmented hub-and-slots shape, using the same coordinates as in Definition 7. There exists a puzzle C excising S' from B . Furthermore, there exists a configuration of C in B ending at the front face of cube $(-1, 0, 7)$ (in the coordinate frame chosen for S').

Proof: We construct C by combining three smaller puzzles: $C = C_1 \circ C_2 \circ C_3$, where C_1 traces the outline of the slots, C_2 can be used to fill in the rest of the bottom two layers, and C_3 can be used to fill the remaining six layers.

The puzzle C_1 is given as follows:

$$C_1 = \langle 1, h_\ell + 1, 2 \rangle \circ \langle 15, \ell + 1, 8t + 2, \ell + 1, 1 \rangle^n \circ \langle h_w - 8tn - 16n - 1, 1 \rangle .$$

When configuring C_1 starting at the left face of cell $(-1, -h_\ell, 0)$, each of the bars longer than h_w (namely, those of length $h_\ell + 1$ and length $\ell + 1$) can only fit along the y direction, and so the other bars must lie in the x direction (as they are longer than the vertical dimension of 8). In fact, because $2\ell + 1$ is longer than the longest dimension of the enclosing box, the length- $(\ell + 1)$ bars must alternate between the positive and negative y direction. Furthermore, the bars between these must all face in the positive x direction; otherwise, an intersection is quickly forced. It follows that, when C_1 is configured starting at the left face of cell $(-1, -h_\ell, 0)$, it must trace the outline of the slots and the front edge of the hub, terminating in cell $(h_w - 1, 0, 0)$.

The next portion, C_2 , is defined in pieces. Define $H_0 = \langle 1, \ell, 2, \ell - 1, 2, \ell - 1, 2, \ell, 1 \rangle$, whose **preferred** (but not forced) configuration is shown in Fig. 5. Also define $H_1 = \langle 1, \ell, 2, \ell, 2, \dots, 2, \ell, 1 \rangle$, where there are $8t + 2$ bars of

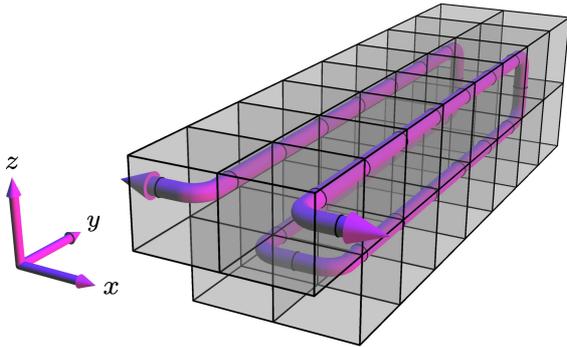


Fig. 5 An example of how the chain H_0 is arranged to fill part of the space between slots.

length ℓ . We define

$$C_2 = \langle \ell - 1, 2, \ell, 1 \rangle \circ H_0^{h_w/2 - 4tn - 8n - 1} \circ (H_1 \circ H_0^7)^n \circ H_0 .$$

Though it is not forced, puzzle C_2 can be configured (starting at the end of C_1 's configuration) to cover the first two layers of the enclosing box with the exception of the interiors of the slots: the H_0 instances are in their preferred configurations and fill the portions between the slots, while each H_1 covers one of the slots. Again, this configuration for C_2 is not forced, but it exists.

The final portion, C_3 , is quite simple: $C_3 = \langle 1, \ell, 2, \ell, 2, \dots, 2, \ell \rangle$, where there are $6(h_w + 1)$ bars of length ℓ . If C_2 is configured as described in the previous paragraph, then C_3 may be configured, starting at the end of C_2 's configuration, to fill the remaining six layers lying over the slots, one layer at a time. In this configuration, C_3 ends at the back face of cube $(0, 0, 7)$, proving that $C_1 \circ C_2 \circ C_3$ has a configuration whose complement is the hub-and-slots shape and whose endpoint is the required face.

It remains to show that $C_1 \circ C_2 \circ C_3$ must cut out the hub-and-slots shape. Consider any configuration of $C_1 \circ C_2 \circ C_3$ with the specified starting face. We already showed that C_1 must trace the outlines of the slots. In puzzle $C_2 \circ C_3$, there are **long bars** of length at least $\ell - 1$ alternating with length-2 bars. By a similar argument as used for C_1 , these long bars must alternate between the positive and negative y direction in the $y \geq 0$ portion of the enclosing box. Because of the design of $C_2 \circ C_3$, it can be observed that each of these long bars must touch the front-most face of the enclosing box. It follows that none of these long bars can lie inside a slot, because C_1 blocks the front box-face. So the slots must remain empty. Because the total number of cubes in $C_2 \circ C_3$ equals the total amount of space in the $y \geq 0$ portion of the box not covered by C_1 or the slots, it follows that $C_2 \circ C_3$ must exactly fill this region. So $C_1 \circ C_2 \circ C_3$ does indeed excise the hub-and-slots shape. \square

Definition 9. For a 3-PARTITION instance V , the second portion of the reduction, $R_2(V)$, is defined as the puzzle C excising the hub-and-slots shape from an $8 \times (h_w + 1) \times (h_\ell + \ell + 1)$ box as in Lemma 8.

6. Zig-Zag Universality

Define the **length- k zig-zag puzzle** as $Z(k) = T^k =$

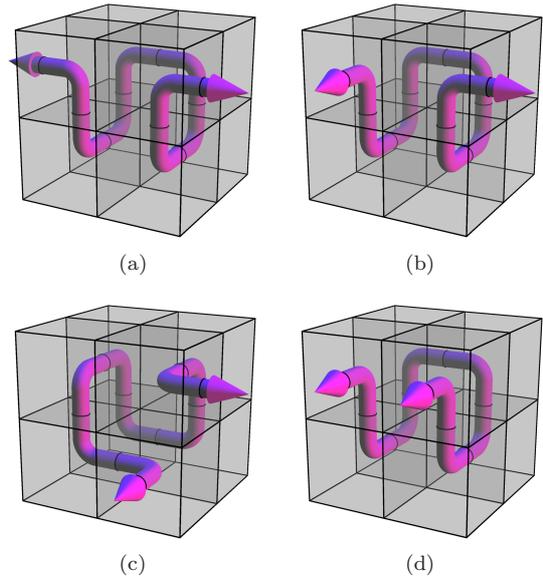


Fig. 6 The configurations of $Z(8)$ that can be used to connect adjacent cubes.

$\langle 1, 2, 2, \dots, 2, 1 \rangle$, where the run-length encoding has $k - 1$ twos. In preparation for the second half of the reduction, we prove a few universality results for classes of regions that can always be filled with zig-zags. In particular, zig-zags can fill arbitrary paths of $2 \times 2 \times 2$ cubes (i.e., paths of cells as defined in Section 2 after $2 \times 2 \times 2$ subdivision of each cell) and arbitrary paths of $4 \times 4 \times 4$ cubes, as well as arbitrary polycubes of $4 \times 4 \times 4$ cubes. As usual, we must be careful about starting and ending positions. All three of these positive results will be used in the following sections to cleanly navigate and fill the hub.

We say that cell (i, j, k) is **even** or **odd** depending on the parity of $i + j + k$; this defines a checkerboard-style labeling of the cells in the infinite grid.

Theorem 10. Suppose we are given a path A of r face-adjacent $2 \times 2 \times 2$ cubes of cells, two cells c_0, c_1 of opposite parity in the first and last cubes of A respectively, and boundary faces f_0, f_1 of c_0, c_1 on the boundaries of their respective $2 \times 2 \times 2$ cubes of A . Then the puzzle $Z(8r)$ can be configured inside A starting at (c_0, f_0) and ending at (c_1, f_1) .

Proof: First we verify the case $r = 1$. As c_0 and c_1 have opposite parity, these cells are either adjacent or diametrically opposite in cube A . In the former case, one of the three configurations of $Z(8)$ shown in Fig. 6 suffices, up to symmetry of the cube and/or reversal of direction. If c_0 and c_1 are opposite, then one of the paths in Fig. 7 works.

The general statement follows by induction on r . Suppose $r \geq 2$, and write A 's cubes in order as A_0, \dots, A_{r-1} . We may assume that c_0 is an even cell. Let c'_0 be one of the odd cells in A_0 adjacent to A_1 along the cell face f'_0 . By the base case there is a configuration of $Z(8)$ in A_0 starting at f_0 and ending at f'_0 , and by induction there is a configuration of $Z(8r - 8)$ inside $A \setminus A_0$ from f'_0 to f_1 . \square

Next we show an analogous statement for $4 \times 4 \times 4$ cubes.

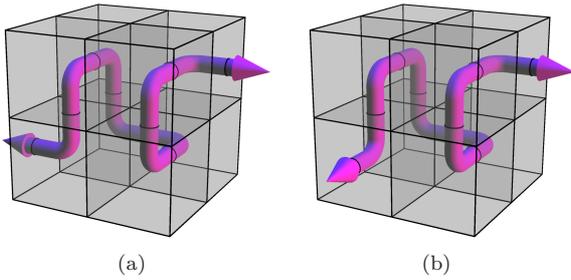


Fig. 7 The configurations of $Z(8)$ that can be used to connect two opposite cubes.

The proof will be almost identical, except that the $r = 1$ case requires more casework.

Theorem 11. Suppose we are given a path A of r face-adjacent $4 \times 4 \times 4$ cubes of cells, two cells c_0, c_1 of opposite parity in the first and last cubes of A respectively, and boundary faces f_0 of c_0 and f_1 of c_1 on the boundaries of their respective $4 \times 4 \times 4$ cubes of A . Then the puzzle $Z(64r)$ can be configured inside A starting at (c_0, f_0) and ending at (c_1, f_1) .

Proof: As above, it suffices to check only the case $r = 1$. A $4 \times 4 \times 4$ box decomposes as the union of eight $2 \times 2 \times 2$ boxes called its *octants*. Let V_0 and V_1 be the octants containing c_0 and c_1 respectively. We consider four cases depending on the relative position of V_0 and V_1 .

If V_0 and V_1 are adjacent, then the desired result follows from Theorem 10 using a path of the eight octants starting at V_0 and ending at V_1 . The same argument works if V_0 and V_1 are opposites. In the other two cases no such path of octants exists, so we must work a bit harder.

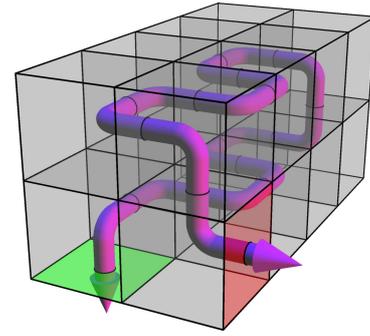
If V_0 and V_1 share an edge but not a face, as in Fig. 8, we may proceed as follows: Choose faces f'_0 and f'_1 as in Fig. 8(b); use Theorem 10 to join faces f_0 and f'_0 while filling the upper-right octants; connect this to H from Fig. 8(a), which connect faces f'_0 and f'_1 while filling the upper-left octants; and use Theorem 10 to connect f'_1 and f_1 while filling the bottom octants.

Finally, suppose $V_0 = V_1$. Up to symmetry, there are six cases to check (recall that c_0 and c_1 have opposite parity). These are shown in Fig. 9, and in each case we use Theorem 10 to join faces f'_0 and f'_1 while filling all octants except $V_0 = V_1$. \square

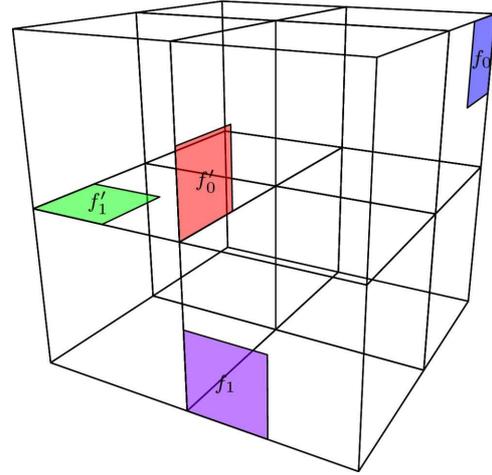
Finally, we show that zig-zags can cover not only paths of $4 \times 4 \times 4$ blocks, but also arbitrary polycubes formed from $4 \times 4 \times 4$ blocks, at the expense of slightly less control over endpoints:

Theorem 12. Let A be a connected polycube of r face-adjacent $4 \times 4 \times 4$ cubes of cells, and take any mark (c, f) on A . Then the puzzle $Z(64r)$ can be configured in A starting at (c, f) .

Proof: As shown in [2], if we subdivide each $4 \times 4 \times 4$ block of A into its eight octants and consider A as a polycube of $2 \times 2 \times 2$ cubes, then there is a hamiltonian cycle through all of these $2 \times 2 \times 2$ cubes. This implies that A may be represented as a path of $2 \times 2 \times 2$ cubes starting at the one



(a) The chain H . The red face is f'_0 , the green face is f'_1 .



(b) An example of how to select the faces f'_0 and f'_1 , given f_0 and f_1 .

Fig. 8 How to connect f_0 to f_1 when V_0 (upper right back) and V_1 (lower right front) share an edge but not a face.

containing mark (c, f) ; if $(c, f) = (\emptyset, \emptyset)$, we may pick this starting cube arbitrarily. By Theorem 10, this path may be filled by $Z(64r)$. \square

7. Fitting the Pegs

We now define the third part of the reduction, R_3 , showing roughly that it fits into the hub-and-slots shape if and only if the 3-PARTITION instance V has a 3-partition.

Definition 13. For each $1 \leq i \leq 3n$, define *peg* $P_i = \langle \ell, 2, \ell, 2, \dots, 2, \ell \rangle$, where there are $8v_i$ bars of length ℓ . Then define

$$R_3 = C_0 \circ Z_1 \circ P_1 \circ Z_2 \circ P_2 \circ Z_3 \circ \dots \circ Z_{3n} \circ P_{3n},$$

where

- $C_0 = \langle 1, 7, 2, 7, 2, \dots, 2, 7, 1 \rangle$, with h_ℓ bars of length 7,
- $Z_1 = Z(128t + 256 + 16h_\ell)$, and
- $Z_2 = \dots = Z_{3n} = Z(k)$, with $k = 64 \cdot 30tn$.

First we prove the easier direction:

Lemma 14. If R_3 can be configured in the augmented hub-and-slots shape (with unconstrained starting and ending positions), then V has a 3-partition.

Proof: In fact, if just the pegs P_1, \dots, P_{3n} can be configured in the hub-and-slots shape without overlap, then V has a 3-partition. Call the length- ℓ bars in the pegs P_i *long*

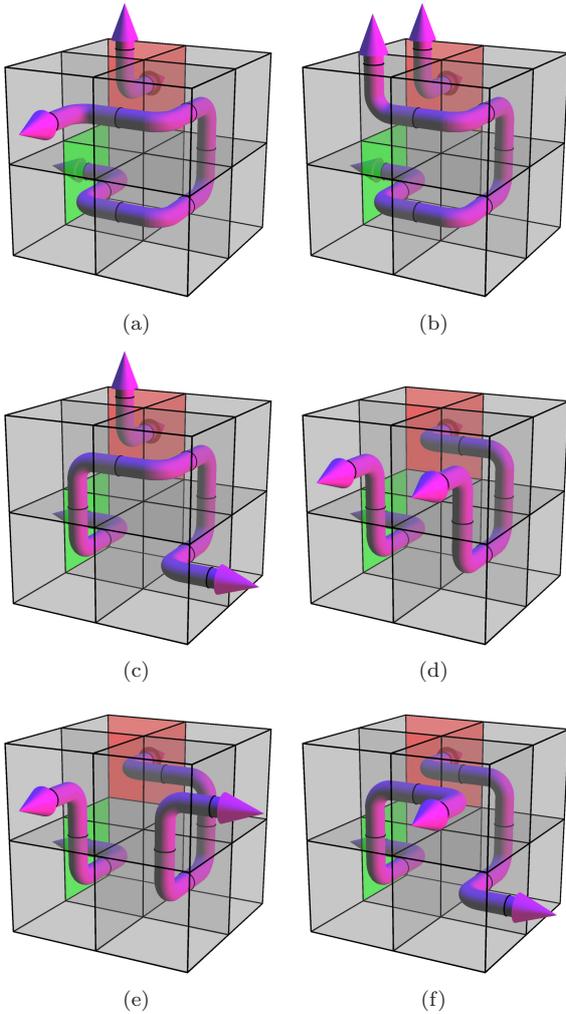


Fig. 9 Given any pair of faces f_0 and f_1 on the outside surface of the $4 \times 4 \times 4$ cube, these are the different ways to connect f_0 to f'_0 (the red face) and f_1 to f'_1 (the green face) within a single $2 \times 2 \times 2$ box.

bars; there are $8tn$ long bars among the pegs.

Because $\ell > h_\ell$ and $\ell > h_w + 1$, each long bar cannot fit fully in the hub and must therefore stick (at least partially) into a slot, meaning it must lie parallel to the y -axis and in the $z = 0$ plane. Furthermore, no two long bars can occupy the same x -coordinate (because $2\ell > h_\ell + \ell$), so the $8tn$ long bars must occupy exactly the $8tn$ x -coordinates of the slots. Finally, each peg P_i must have all of its long bars in a single slot, because the length-2 bars force the parallel long bars of a single peg to be adjacent. So the widths of the pegs in each slot add exactly to the width of the slot, and these widths (divided by 8) exactly give a 3-partition for $\{v_1, \dots, v_{3n}\}$. \square

The other direction is more difficult: we must show that, if there is a 3-partition, then R_3 can be configured in the hub-and-slots shape. As usual, we pay special attention to the endpoints.

Lemma 15. *If V has a 3-partition, then there is a configuration of R_3 in the augmented hub-and-slots shape starting at the front face of cell $(-1, 0, 7)$, such that the unfilled region is a face-connected polycube of $4 \times 4 \times 4$ cubes and*

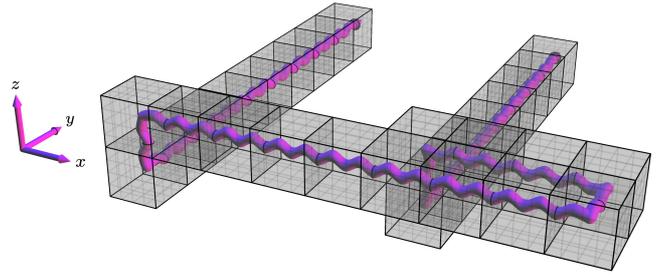


Fig. 10 An example of how Z_i might be configured. Each cube is a $4 \times 4 \times 4$ brick; the path indicates the order that Z_i passes through the bricks. Within each brick, the configuration of Z_i will be determined by Theorem 11; the oscillations serve as a visual reminder that the path of cubes winds through the 64 cells in each brick.

such that the configuration ends at a boundary face of this polycube.

Proof: The hub—specifically, the box with opposite corners at $(0, -h_\ell, 0)$ and $(h_w, 0, 8)$ —may be partitioned into $4 \times 4 \times 4$ boxes, which we refer to as **bricks** and identify in coordinates by their left, back, bottom corners.

Because there exists a 3-partition of V , the pegs P_1, \dots, P_{3n} may be positioned to exactly fill the slots. Do so in such a way that each peg starts and ends on the boundary of the hub, with the starting face to the right of the ending face. The brick adjacent to the starting face of P_i is the **starting brick** for P_i , and the **ending brick** is defined similarly. We may further assume that peg P_1 is chosen as the rightmost peg in the leftmost slot, so its starting brick is located at $(8t + 12, -4, 0)$. The $3n$ starting bricks and the $3n$ ending bricks are all distinct, and their left, back, bottom corners all have $y = -4$ and $z = 0$.

Configure C_0 to fill the “augmentation” of the augmented hub-and-slots: it starts at the front face of cell $(-1, 0, 7)$, has its length-7 bars in the z -direction and its length-2 bars in the negative y -direction, and ends at the right face of cell $(-1, -h_\ell, 7)$.

Now everything has been filled except for the hub, which is currently empty. We must show how to configure Z_1, \dots, Z_{3n} in the hub so that they connect where they must, don’t overlap each other, and leave a connected polyomino of bricks as their complement.

First we position Z_1 . Consider the path of bricks that starts at brick $(0, -h_\ell, 4)$ in the left, back, top corner of the hub, then moves right, down, and then forward to the starting brick of P_1 , namely $(8t + 12, -4, 0)$. This path has $2t + 4 + h_\ell/4$ bricks. By Theorem 11, we may configure Z_1 to exactly fill this path of bricks starting at the left face of cell $(0, -h_\ell, 7)$ (an odd cell) and ending at the front face of cell $(8t + 15, -1, 0)$ (an even cell). These faces are, respectively, where C_0 ends and where P_1 starts.

Finally, we describe how to configure each puzzle Z_i , for $2 \leq i \leq 3n$. Construct a path of bricks connecting the ending brick of P_{i-1} (say this brick has coordinates $(4q, -4, 0)$ with q odd) to the starting brick of P_i (with coordinates $(4r, -4, 0)$ with r even) as follows. If $q < r$, then:

- Start at $(4q, -4, 0)$,
- proceed back, then up, then right to $(s, -12i, 4)$ (parameter s is explained below),
- step forward one brick to $(s, -12i + 4, 4)$,
- then move left, then down, then forward to the starting brick of P_i , $(4r, -4, 0)$.

An example is shown in **Fig. 10**. If $q > r$, follow these instructions instead from $(4r, -4, 0)$ to $(4q, -4, 0)$ and then reverse its direction. Here, $s > \max(4q, 4r)$ is chosen so that the path covers a total of $30tn$ bricks: $s = 60tn + 2(q + r) - 12i - 2$, and we indeed have $s > 60tn - 36n \geq 8tn + 16n > \max(4q, 4r)$. (Note also that $4 \mid s$ because $q + r$ is odd.) Now, by Lemma 11, we may configure path Z_i to exactly fill this path of bricks, starting at the ending brick of P_{i-1} and ending at the starting brick of P_i . More specifically, we may configure Z_i in this path so that it starts at the front face of cell $(4q, -4, 0)$ —this cell is even and this face is the ending face of P_{i-1} —and ends at the front face of cell $(4r + 3, -4, 0)$ —an odd cell and the starting face of P_i .

Why do these configured puzzles Z_i not intersect each other? In the top layer of bricks, each Z_i is contained in a different set of rows: for $2 \leq i \leq 3n$, Z_i touches only rows $-12i$ and $-12i + 4$, and Z_1 touches only row $-h_\ell < -36n$. In the bottom layer of bricks, each Z_i for $1 \leq i \leq 3n$ is contained only in the columns corresponding to the starting and/or ending bricks it touches, and these columns are all distinct.

Let M be the complement of the Z_i configurations in the hub. Why is M connected? The top layer of M is connected because no individual configuration Z_i separates any bricks in the top layer, and no two configurations Z_i and Z_j are adjacent in the top layer. For each column in the bottom layer, the portion of the row contained in M is a connected sequence of bricks that is adjacent to at least one of the bricks in the top layer of M . Hence, M is indeed connected.

Finally, P_{3n} ends at a face adjacent to its ending brick, and this brick is contained in M . \square

8. Filling the Gaps

In the final portion of the reduction, we define $R_4(V)$ as a zig-zag with enough cubes to exactly fill the remaining portion of the original cube: $R_4(V) = Z(u^3 - \text{vol}(R_1(V) \circ R_2(V) \circ R_3(V)))$, where vol counts the number of cubes in the puzzle. With the results built up in the previous sections, the full result is now readily proved:

Theorem 16. *Take a 3-PARTITION instance V and build the resulting puzzle $R(V) = R_1(V) \circ R_2(V) \circ R_3(V) \circ R_4(V)$ of u^3 cubes as in the previous sections. Then puzzle $R(V)$ can be configured to exactly fill a $u \times u \times u$ box if and only if there exists a 3-partition of V .*

Proof: If $R_1 \circ R_2 \circ R_3$ can be configured in a $u \times u \times u$ box, then Lemmas 5, 8, and 14 guarantee that V has a 3-partition. Conversely, if V has a 3-partition, then Lemmas 5, 8, and 15 imply that $R_1 \circ R_2 \circ R_3$ may be configured so that the unfilled portion is a connected polycube of $4 \times 4 \times 4$ cubes,

ending at a face adjacent to this polycube. By configuring R_4 in this region by Theorem 12, we obtain a configuration of $R_1 \circ R_2 \circ R_3 \circ R_4$ filling the entire cube. \square

Corollary 17. *The problem of deciding if a given puzzle can exactly fill a cube is NP-complete.*

Proof: Such a configuration may be easily verified, so the problem is in NP. By Lemma 5, for an instance $V = \{v_1, \dots, v_{3n}\}$ with target sum t , the chosen cube side-length u is at most $4(h_\ell + \ell + 1) = \text{poly}(t, n)$, and the length of $R(V)$ is $u^3 = \text{poly}(t, n)$. The explicit map $V \mapsto R(V)$ may be computed in polynomial time, and Theorem 16 guarantees this is a valid Turing reduction. Because 3-PARTITION is strongly NP-complete, the result follows. \square

9. Open Questions

In this paper we have analyzed a natural generalization of the Snake Cube puzzles, but many related questions remain open:

- We only address whether a solved configuration of the puzzle exists. The actual puzzle, however, must be physically moved into its solved state. Can every solved configuration be reached by a continuous, non-self-intersecting motion from the initial (flat and monotone) configuration? If not, can this decision problem be solved efficiently?
- Is the analogous 2-dimensional problem also hard? Specifically, is it NP-complete to decide whether an S-T sequence of squares can pack into an $N \times N$ grid of squares, where S and T represent “straight” and “turn” squares as before? This is equivalent to the 3-dimensional problem of packing a puzzle into an $N \times N \times 1$ box. This also relates to the problem of finding simple, planar configurations of fixed-angle chains, a version of which was shown to be NP-hard in [3]. The methods from this paper, however, do not seem to directly apply to this setting.
- What if the target shape is allowed to have large “holes”? For example, is it still NP-hard to decide if a puzzle of $\alpha \cdot N^3$ cubes can be configured inside an $N \times N \times N$ cube, for all constants $0 < \alpha < 1$?

Acknowledgments These results arose from an open problem session organized for the *Geometric Folding Algorithms: Linkages, Origami, Polyhedra* course taught by Erik D. Demaine at MIT in Fall, 2010. We thank the other participants of this session for a productive and stimulating research atmosphere.

References

- [1] Creative Craffthouse. King snake.
- [2] Erik D. Demaine, Martin L. Demaine, Jeffrey F. Lindy, and Diane L. Souvaine. Hinged dissection of polypolyhedra. In *Proceedings of the 9th Workshop on Algorithms and Data Structures (WADS 2005)*, volume 3608 of *Lecture Notes in Computer Science*, pages 205–217, Waterloo, Ontario, Canada, August 15–17 2005.
- [3] Erik D. Demaine and Sarah Eisenstat. Flattening fixed-angle chains is strongly NP-hard. In *Proceedings of the 12th Algorithms and Data Structures Symposium (WADS 2011)*, pages

- 314–325, Brooklyn, New York, August 15-17 2011.
- [4] M. R. Garey and D. S. Johnson. Complexity results for multi-processor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975.
 - [5] Indiana University Lilly Library. The jerry slocum mechanical puzzle collection. <http://webapp1.dlib.indiana.edu/images/splash.htm?scope=lilly/slocum>. Acquisitions 024867, 024868, 024869, 024870, and 024871. <http://webapp1.dlib.indiana.edu/images/item.htm?id=http://purl.dlib.indiana.edu/iudl/lilly/slocum/LL-SLO-024867>, etc.
 - [6] Alison Elizabeth McDonough. Snake cube puzzles: Hamilton paths in grid graphs. Honors project, 2009.
 - [7] F. Ruskey and Joe Sawada. Bent hamilton cycles in d -dimensional grid graphs. *The Electric Journal of Combinatorics*, 10(1):R1, 2003. <http://www.combinatorics.org/ojs/index.php/eljc/article/view/v10i1r1>.
 - [8] Jaap Scherphuis. Jaap’s puzzle page: Snake cubes. <http://www.jaapsch.net/puzzles/snakecube.htm>, 2010.
 - [9] David Singmaster. Singmaster cube and puzzle collection. CD distributed at Gathering 4 Gardner 6, March 2004.



Zachary Abel is a Ph.D. candidate in the Department of Applied Mathematics at the Massachusetts Institute of Technology (MIT) working with Professor Erik D. Demaine in the Computer Science and Artificial Intelligence Lab (CSAIL). His primary research is in computational and discrete

geometry, including such topics as computational origami and geometric rigidity theory. Mr. Abel is currently funded under the National Science Foundation (NSF) Graduate Research Fellowship Program, and he received his Bachelors of Arts in Mathematics and Computer Science from Harvard in 2010.



Erik D. Demaine received a B.Sc. degree from Dalhousie University in 1995, and M.Math. and Ph.D. degrees from University of Waterloo in 1996 and 2001, respectively. Since 2001, he has been a professor in computer science at the Massachusetts Institute of Technology. His research interests

range throughout algorithms, from data structures for improving web searches to the geometry of understanding how proteins fold to the computational difficulty of playing games. In 2003, he received a MacArthur Fellowship as a “computational geometer tackling and solving difficult problems related to folding and bending—moving readily between the theoretical and the playful, with a keen eye to revealing the former in the latter”. He cowrote a book about the theory of folding, together with Joseph O’Rourke (*Geometric Folding Algorithms*, 2007), and a book about the computational complexity of games, together with Robert Hearn (*Games, Puzzles, and Computation*, 2009).



Martin L. Demaine is an artist and mathematician. He started the first private hot glass studio in Canada and has been called the father of Canadian glass. Since 2005, he has been the Angelika and Barton Weller Artist-in-Residence at the Massachusetts Institute of Technology. Both Martin

and Erik work together in paper, glass, and other material. They use their exploration in sculpture to help visualize and understand unsolved problems in mathematics, and their scientific abilities to inspire new art forms. Their artistic work includes curved origami sculptures in the permanent collections of the Museum of Modern Art (MoMA) in New York, and the Renwick Gallery in the Smithsonian. Their scientific work includes over 60 published joint papers, including several about combining mathematics and art.



Sarah Eisenstat is a Ph.D. student in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, working with Professor Erik D. Demaine. Her current research interests include computational geometry and algorithms. She received her Sc.B. and

Sc.M. degrees from Brown University in 2008 and 2009, respectively.



Jayson Lynch is an undergraduate in Physics and Computer Science at the Massachusetts Institute of Technology (MIT). He is a member of MIT’s Computer Science and Artificial Intelligence Laboratory (CSAIL) where he works with Professor Erik Demaine. His research interests include

computational origami, games and computation, algorithms, and the interplay between physics and computation.



Tao B. Schardl is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science (EECS) at the Massachusetts Institute of Technology (MIT). He is a member of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL), and works in the Supertech

group with Professor Charles Leiserson. His primary research is on developing parallel algorithms and parallel data structures. His previous research includes developing a work-efficient parallel breadth-first search algorithm, designing a theoretical framework for analyzing Cilk programs with nonconstant-time reducer hyperobjects, and developing mechanisms to support deterministic parallel random-number generation. Mr. Schardl received his Bachelor of Science in Computer Science and Electrical Engineering from MIT in 2009 and his Masters of Engineering in Computer Science and Electrical Engineering from MIT in 2010.