
Identifying Metastable States in Origami-Inspired Metamaterial Unit Cells



BACHELOR OF SCIENCE

submitted by Yun Li, *AUC*
supervised by Dr. Forrest Bradbury, *AUC*
and Dr. Ir. Bas Overvelde, *AMOLF*

AMSTERDAM UNIVERSITY COLLEGE
May 31, 2017

Layman's abstract

Properties of day-to-day materials are determined by their chemical composition: steel is stronger than plastic because the chemical bonds in the former is stronger than the latter. Metamaterials go one step beyond the limitations of chemistry. They are composed of unit cells larger than the scale of molecules or atoms, and the unit cells are arranged in different patterns so that metamaterials have properties rarely found in nature and usually beyond the properties of their base chemical compositions.

Mechanical metamaterials are a class of metamaterials that have novel mechanical features. This study focuses on multistability: the phenomenon that one structure can switch between different configurations, with each of the configuration stable in itself.

Recent work on origami-inspired metamaterials are reviewed in this work, followed by a mathematical model for unit cells that arise in a class of mechanical metamaterials. Three approaches are proposed to identify multistability in these unit cells, and their results are also presented. The findings are the first step to studying the multistability of bulk metamaterials, and are also useful for designing transformable devices.

Abstract

Mechanical metamaterials are composed of unit cells arranged in a designed pattern, so that their properties are not determined by the chemical compositions but rather its mechanical structure. It is a rapidly growing field of research, in which transformable metamaterials are particularly interesting due to their flexibility and functionality. However, for some origami-inspired modular metamaterials, it is difficult to identify metastable states to which they can transform due to the high degrees of freedom in such problems.

This project utilises a numerical model of a class of origami-inspired unit cells, and proposes three approaches to identify their metastable states by finding local energy minima in their energy function. The brute force approach and the stochastic approach based on Metropolis-Hastings algorithm are unable to explore the entire configuration space. In the final approach, we make use of the inherent symmetry in these unit cells, and reduced the amount of searches we need to perform, so that the problem becomes more manageable.

Tests were run on the truncated tetrahedron unit cell, and we found 10 metastable states, most of them previously unknown. The results can be generalised for designing space-filling tessellations that preserve multistability, which can further be applied to studying multistability in bulk metamaterials and designing distributed actuation systems on these metamaterial systems. They are also good candidates for transformable devices over a large range of length scales.

Contents

1	Introduction	3
2	Literature Review	4
2.1	Mechanical Metamaterials	4
2.2	Optimisation of High-Dimensional Functions	6
3	Mathematical Model and Brute Force approach	7
3.1	Model description	7
3.2	Brute force approach	8
3.2.1	Optimisation procedure for actuating hinges	9
3.2.2	A case study: extruded truncated tetrahedron	10
3.2.3	Feasibility of the brute force approach	11
4	Stochastic approach	11
4.1	Metropolis-Hastings algorithm	11
4.2	Optimisation procedure	12
4.3	Results from stochastic approach	13
5	Duplication Reduction	14
5.1	Conversion to graphs	14
5.2	Selection of nodes	15
5.2.1	Example hinge sets enumerated by hand	16
5.2.2	The “distance” matrix for determining unique hinge combinations	18
5.2.3	Submatrices of the “distance matrix”	21
5.2.4	Procedure of selection of unique hinge sets	21
5.3	Optimisation procedure	23
5.4	Results from duplication reduction	23
6	Discussions and Conclusion	25
Bibliography		28
A	Energy as a function of node coordinates	30
A.1	Vertices, edges, hinges, and faces	30
A.2	Energy and its gradient	32
B	Graphs: Directed vs. Undirected	33
C	Fabrication of prototypes	33
D	List of all (semi)-regular convex polyhedra	34
E	Space-filling tessellations of (semi-)regular convex polyhedra	36

List of Figures

1	Construction of a truncated-tetrahedron-based unit cell and its multistability.	3
2	The shape of the extruded unit cell is controlled by three angles: γ_1 , γ_2 , and γ_3 [11].	5
3	Deformations of a cube-based unit cell and their 3-D tessellations [11].	5
4	Miura-fold [15]	6
5	All movements allowed in the model.	8
6	Two types of hinges present in a unit cell.	9
7	Metastable states of truncated tetrahedron unit cell obtained through brute force. .	10
8	Random movements in stochastic methods may result in unphysical configurations. .	13
9	Two equivalent sets of hinges that appear in the brute force method.	14
10	Different polyhedra mapped onto directed graphs.	15
11	All possible ways to select 1, 2, or 3 hinges from tetrahedron.	16
12	All possible ways to select 1 or 2 hinges from truncated tetrahedron.	16
13	All possible ways to select 1, 2, 3, or 4 hinges from cube.	17
14	Choosing one hinge from the cube.	18
15	Four different cases of choosing two hinges from the cube.	19
16	Metastable states found by actuating hinge sets in the reduced list.	24
17	Metastable states found in figure 7 and 16 replicated with cardboard prototype. .	24
18	One metastable state with an extruded triangle popped through to the opposite side	25
19	Two different hinge sets in the reduced list converge to the same metastable state.	26
20	Two metastable configurations with the same set of hinges closed.	27
21	Hinge angle θ can be obtained from vectors a , b , and c	31
22	Small deformations associated with face bending can be described by φ	32
23	Cube with hinges numbered from 1 to 12.	33
24	Graphs constructed from cube.	33
25	Cardboard prototypes of unit cells.	34
26	List of uniform 3D tessellations [12].	36

List of Tables

1	Number of possible ways to select actuated hinges for different polyhedra	11
2	Hinge/Node selection for tetrahedron	23
3	Number of hinge sets after reduction.	23
4	Five Platonic solids	34
5	Thirteen Archimedean solids	35
6	Prisms and antiprisms	36

1 Introduction

Metamaterials are materials that are engineered to have properties beyond their constituent parts. Like a crystal where atoms or molecules are arranged in a certain pattern, a metamaterial also has its unit cells arranged in repeating patterns. Unlike a crystal, the unit cell in a metamaterial is artificially designed, and its size can range from several nanometres to several metres. The special properties of metamaterials are often a result of geometrical structures, rather than chemical compositions. They have great potential for applications in fields such as cloaking devices [14], impact absorbers [20], and reconfigurable devices and robotics [11, 12, 15] due to their diverse functionality.

Mechanical metamaterials are a group of metamaterials with special mechanical properties [11, 12, 14, 15, 16, 19, 20]. This study focuses on the multistability of unit cells that arises in some mechanical metamaterials. A multistable unit cell has two or more stable configurations, and can transition between them under certain circumstances. One source of inspiration for designing mechanical metamaterials is origami: by folding paper in different ways, we can get structures with different properties, such as shapes, volumes, or rigidities.

This project uses a numerical model to study the multistability of modular origami-like designs. The structures explored in this study are obtained by extruding faces of a (semi)-regular polyhedron along the normal direction, until the extrusion has the same length as the edge length of the template polyhedron [11, 12]. The template is then discarded, leaving only the hollow structure, as demonstrated by an example in the top panel of figure 1. The variables used in the simulation are the 3-D coordinates of the extruded structure. By studying the energy of each unit cell as a function of its configurations, we can identify unit cells that are multistable. The list of all (semi)-regular polyhedra used as templates in this study is included in appendix D.

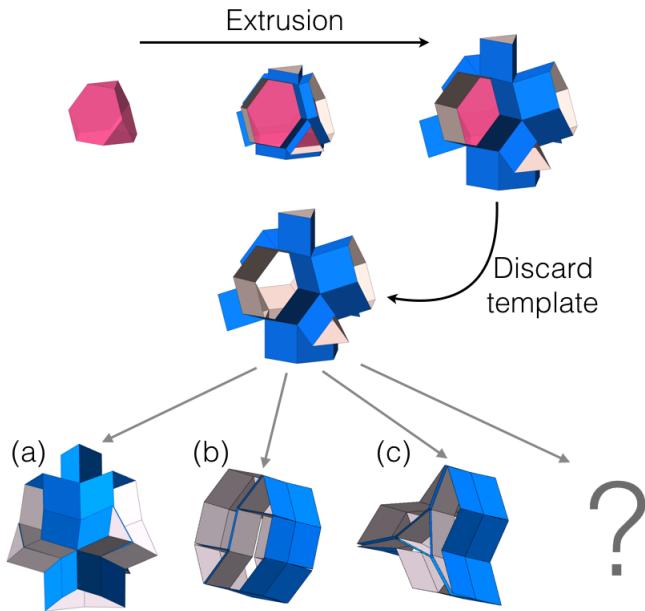


Figure 1: Construction of a truncated-tetrahedron-based unit cell and its multistability.
Top panel shows the extrusion process; Bottom panel shows three instances of metastable states of this unit cell already discovered at the beginning of this project, and the potential of discovering more metastable states.

Since a configuration with lower energy would be more stable, a local minimum in the energy function would correspond to a stable or metastable state¹, and more than one local minima would imply multistability. Therefore, we can treat the objective of finding metastable states as an optimisation problem where we search for local minima in the energy function. However, the number of variables inherent in each origami model is so high that the optimisation problem is difficult to solve, analytically or numerically².

This study explores three different approaches to tackle this problem. First, a brute force method is presented, and we demonstrate that the sheer amount of computation needed to generalise this method to more complicated polyhedra makes it practically impossible. This motivates the second stochastic approach, where we attempt to find configurations that are close to local energy minima by randomly sampling the configuration space. This method is also not effective due to the high dimensionality of the space. The last approach explores the symmetry in each origami unit cell, and uses this information to reduce the number of searches drastically.

We present a reliable method which is able to produce the metastable energy states given any polyhedra-based origami structure. This could pave the way for further studies on properties of this type of origami-inspired metamaterials, and bring them closer to applications in engineering and design. We also discuss the limitations of the presented methods, and provide directions for further improvements.

2 Literature Review

2.1 Mechanical Metamaterials

Metamaterials are a rapidly growing field of research, where scientists study and engineer new materials with novel optical, sonic, or mechanical properties not found in nature. Metamaterials usually consist of artificial unit cells arranged in repeating pattern that resembles a crystal structure, and it is their structure, rather than chemical composition, that gives rise to their novel properties. In the study of mechanical metamaterials, special mechanical properties are harnessed such as negative Poisson's ratio, negative stiffness, and multistability [11, 12, 14, 15, 16, 19, 20].

Recently, snapology—a branch of origami—has inspired the design of a class of modular mechanical metamaterials. Heinz Strobl initially invented this method to construct polyhedra using ribbons of paper [4], and Overvelde et al. adapted this method to fabricate transformable metamaterials. They analysed a cube-based unit cell, and showed that under the constraint of rigid faces, there are three degrees of freedom (dof) left. The 3 dof correspond to three angles $\gamma_1, \gamma_2, \gamma_3$ that lie in the template cube (see figure 2), and they can deform continuously in the range of $[0, \pi]$. Thus, this unit cell is not multistable, but is reconfigurable because it can be completely folded with external actuation [11], as shown in figure 3. They also showed that it is possible to control the shape, volume and stiffness of material consisting of this type of unit cells through embedded pneumatic actuation distributed on each unit cell.

¹A metastable state exists at an energy level higher than that of a more stable configuration, and certain perturbations can induce a transition to the more stable state.

²In the simple case of the extruded truncated tetrahedron in figure 1, for example, the total number of vertices is 48, which means the total number of variables is $48 \times 3 = 144$, with three coordinates for each vertex. The high dimensionality makes it difficult to find the path to optimal configurations. In this sense this is comparable to the problem of protein folding, which is considered to be hard to solve [2, 3].

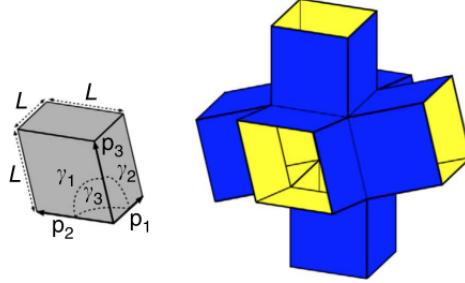


Figure 2: The shape of the extruded unit cell is controlled by three angles: γ_1 , γ_2 , and γ_3 [11].

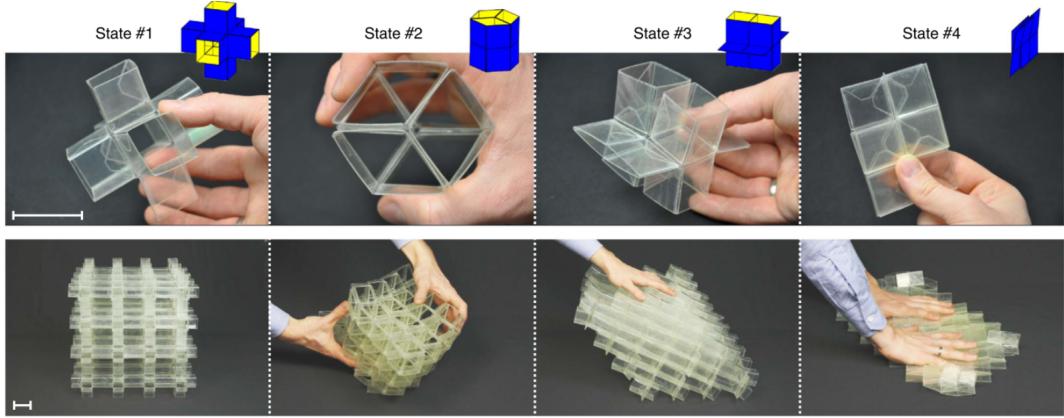


Figure 3: Deformations of a cube-based unit cell and their 3-D tessellations [11].
(scale bars: 3 cm)

The unit cells were fabricated using two outer layers of polyethylene terephthalate (thickness $t = 0.25$ mm and 0.05 mm) and one layer of double-sided tape ($t = 0.05$ mm) in the middle.

Supplementary videos can be found at

<https://www.nature.com/articles/ncomms10929>

Following that, Overvelde et al. used the same design strategy derived from snapology, and proposed a robust method to construct space-filling tessellations³ of polyhedra-based unit cells [12]. They showed that by selectively extruding faces on polyhedra, the dof in the tessellation can be tuned, thereby enhancing or decreasing the reconfigurability.

Apart from reconfigurability, multistability is another interesting property that can be found in some of the similar origami-inspired metamaterials. It is observed that the truncated tetrahedron unit cell can “pop through” to other metastable states with external actuation, and stay in that configuration if no other perturbations are present, as shown in figure 1 (a)–(c). This is similar to the bistability in the Miura-fold observed by Silverberg et al. [15]. The Miura-fold is another type of Origami consisting of four adjacent equal-area facets joined together by alternating mountain and valley creases, with the four creases intersecting at one vertex (see figure 4 a, b). In 2D tessellations of Miura-fold, a pop-through defect can be introduced by popping one vertex, so that one fold is suppressed and adjacent faces are bent through an angle ϕ , as shown in figure 4 (c)–(e).

³A space-filling tessellation is a tiling in three dimensional space using polyhedra, with no overlaps and no gaps. Here we only consider periodic tessellation with (semi)-regular polyhedra, and a list of all such tessellations can be found in appendix E.

By introducing defects at different locations, they were able to control the compressive modulus and thus change the properties of the system.

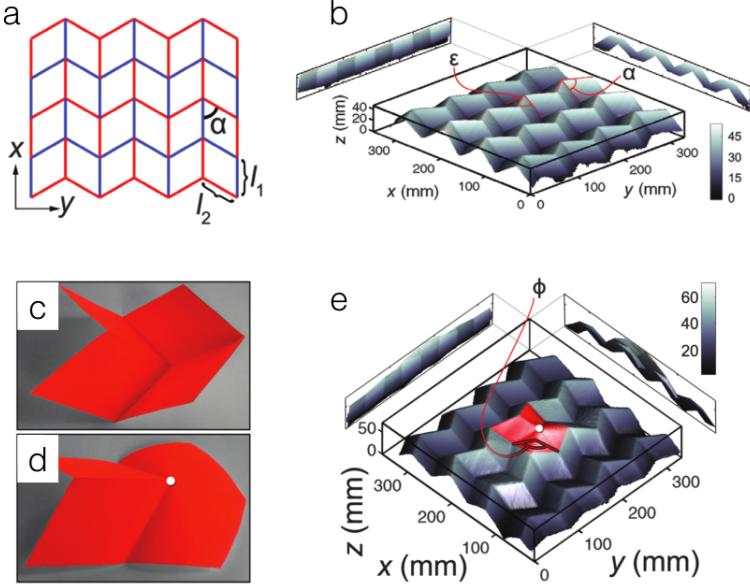


Figure 4: Miura-fold [15]
 (a): Schematic of Miura-fold and (b): a 3-D construction.
 (c)–(e) shows an introduced defect which is meta-stable.

Owing to their reconfigurability and multistability, origami-inspired metamaterials have potential applications in many fields, such as self-folding robots, foldable solar panels, reconfigurable devices such as exoskeletons, turnable photonic systems, and other morphable systems [11, 12, 14, 15, 16, 19, 20]. Also, their scale-free character allows the material designs be directly translated to a large range of length scales.

In addition to experimental studies, numerical models are also widely used for studying origami-inspired metamaterials. Origami structures can be modelled as rigid planar faces connected by torsional hinges with a large range of motion. These models provide reasonably good descriptions of folded paper sheets [19], as well as laser-cut thin plastic sheets connected by adhesive tapes [11]. Further, Silverberg et al. discovered that allowing the faces to bend brings about hidden degrees of freedom that allows the system to deform into states otherwise not accessible with rigid faces [16]. Therefore, all these possible deformations need to be considered when modelling these origami-inspired metamaterials.

2.2 Optimisation of High-Dimensional Functions

The study of mathematical optimisation is a vast endeavour. Modern computers make iterative techniques such as the steepest descent method, Newton’s method, and the interior point methods

more accommodating in terms of the cost for time and memory space. However, for high dimensional problems or problems with a non-convex objective function, these methods soon become ineffective or inefficient. In this case, stochastic methods or heuristics/metaheuristics⁴ are often favoured because of their ease to use and efficiency in finding the global minimum⁵.

Osman defines a metaheuristic as “an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search spaces using learning strategies to structure information in order to find efficiently near-optimal solutions” [10]. In other words, it borrows ideas from other fields of science, such as evolutionary biology, artificial intelligence, and statistical mechanics, and use them as a guideline to search for optimal solutions. Examples of metaheuristics include tabu search, simulated annealing, Metropolis-Hastings, neural networks, multi-start methods, as well as hybrids of these methods [8, 10, 17, 18].

Metropolis et. al proposed a Monte-Carlo method to simulate materials consisting of interacting individual molecules. It is now referred to as the Metropolis-Hastings algorithm which is explained in detail in section 4.1 below, and it captures the evolution of these molecules towards thermal equilibrium at a fixed temperature T [9]. For a more general system, thermal equilibrium can be interpreted as a local minima in the system’s energy at temperature T . In this method, there is a possibility to accept worse solutions on the way towards optima, which thus allows more extensive exploration and search in the configuration space. It is now also widely used for drawing random samples from an unknown (usually high-dimensional) distribution function [18].

Simulated annealing (SA) is an adaptation of the Metropolis-Hastings algorithm, and is also stochastic in nature. Similar to the Metropolis-Hastings algorithm, it describes a process where a system consisting of molecules gradually achieves thermal equilibrium. However, a key difference is that the temperature is slowly lowered in the process of SA, so that the system is more likely to end up in the global optimum position [10]. It is aptly named after its metallurgical namesake, annealing, a process where heated metal is cooled down slowly to achieve better ductility.

It is worth noting that despite their potentially powerful performance, not all metaheuristic methods guarantee final convergence due to their stochastic nature [10, 18]. However, in deterministic methods, convergence is more straightforward to prove or disprove.

3 Mathematical Model and Brute Force approach

3.1 Model description

The implementation of the numerical model for origami-like unit cells is discussed in this section. It is adopted from previous works by Overvelde et al. [11, 12], and is scripted in MATLAB.

As noted above, it is simple and experimentally valid to model facets connected by creases as a set of torsional springs [11, 19]. To account for the hidden degrees of freedom involved in face-bending [16], as well as possible stretching in the edges, we also treat them as springs and include their potential energies in the total energy, with spring constants k_{hinge} , k_{face} , and k_{edge} respectively.

⁴This is not an exclusive categorisation.

⁵Broadly speaking, stochastic methods use random values to probe the distribution of outputs, and give estimations based on these outputs. Deterministic methods (which include the above mentioned steepest descent method, Newton’s method, and interior point method), on the other hand, produce outputs fully determined by the input parameter, and no randomness is involved in the process.

We can also define the deformations in each hinge $\Delta\theta_i$ as the difference between the hinge angle with the neutral angle. Since the faces are allowed to bend, we capture this deformation with an out-of-plane distance $\Delta\varphi_i$ for each face⁶. Lastly, the deformation of each edge Δe_i is the lengths it contracts or expands. These movements are summarised in figure 5 below.

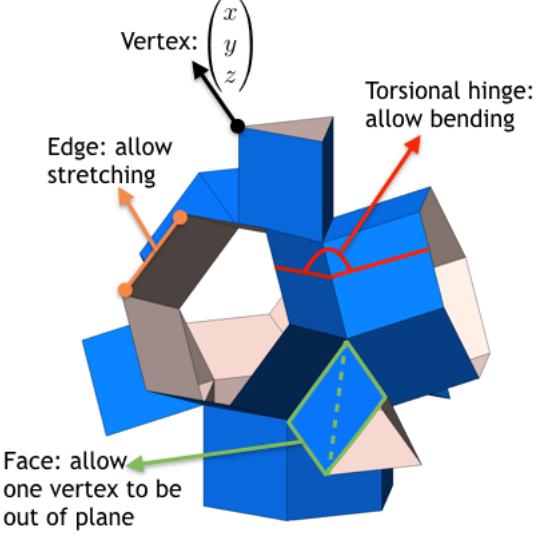


Figure 5: All movements allowed in the model.
Note that the angle θ of the torsional hinge lies in the range of $[0, 2\pi]$.

Then, the energies associated with these movements are:

$$E_{\text{hinge}} = \sum_{i=1}^H \frac{1}{2} k_{\text{hinge}} (\Delta\theta_i)^2 \quad (1)$$

$$E_{\text{face}} = \sum_{i=1}^F \frac{1}{2} k_{\text{face}} (\Delta\varphi_i)^2 \quad (2)$$

$$E_{\text{edge}} = \sum_{i=1}^E \frac{1}{2} k_{\text{edge}} (\Delta e_i)^2 \quad (3)$$

where H , F , and E are the total number of hinges, faces, and edges present in the unit cell respectively. The total energy for each unit cell is

$$E_{\text{tot}} = E_{\text{edge}} + E_{\text{face}} + E_{\text{hinge}} \quad (4)$$

Since all deformations can be expressed in terms of the coordinates of vertices of the unit cell, the total energy E_{tot} is also a function of the coordinate of vertices. The details are derived in Appendix A.

3.2 Brute force approach

Experimentally, the multistability of these origami structures was first discovered by closing some of the hinges, and observing that the structure “snaps” into a deformed metastable state. This

⁶For details, see appendix A.

can be translated into constrained optimisation in terms of numerical simulations, where a closed hinge is represented by an angle of zero degrees. The most straightforward idea, then, is to assign a number to each hinge in the structure, and make a list of all the possible combinations of them. Each combination is a unique “hinge set”, and by closing all hinges in such a hinge set simultaneously, we can enumerate all possible stable configurations.

In this study, the assumption is made that all meta-stable states will have at least one hinge closed, and that no other configurations can be stable with all angles in the range of $(0, 2\pi)$, except the original configuration. This allows us to explore the configuration space by actively closing combinations of hinges.

Before we set out to investigate how feasible this seemingly naïve approach is, we state a second assumption: for each unit cell, its deformed metastable states can be characterised by the presence of completely closed hinges. In particular, it is sufficient to use only the set of hinges that lie on the skeleton of the inner polyhedra⁷. Therefore, the polyhedron hinges are the only ones considered henceforth.

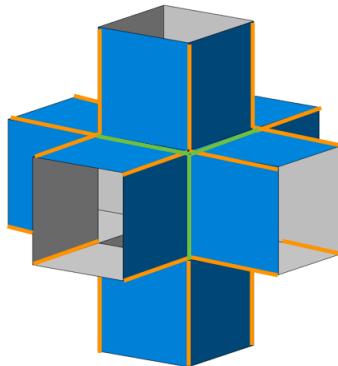


Figure 6: Two types of hinges present in a unit cell.
Green: polyhedron hinges; orange: extruded hinges.

This assumption can be made because the angles of the two types of hinges are not independent from each other. Once the polyhedra hinges are constrained, the optimal configuration of the extruded hinges are also determined.

3.2.1 Optimisation procedure for actuating hinges

The following procedure outlines how to identify all possible metastable states by using deterministic optimisation algorithms including gradient descent and interior point method:

1. An extruded polyhedron object is created and stored in memory.
2. A set of hinges is selected to be actuated at the same time.
3. Then, constraints are imposed on the selected hinges, i.e., the selected set of hinges are closed, and optimisation⁸ is started until the structure converges or the maximum number

⁷We shall call these hinges on the inner polyhedra the “polyhedron hinges” (green in figure 6), and those on the extruded parts the “extruded hinges” (orange in figure 6).

⁸The MATLAB built-in optimisation solver fmincon with the interior-point algorithm is used.

of iterations allowed is reached. In the latter case, we can conclude that actuating this set of hinges does not result in a physically meaningful result.

4. If convergence does occur in step 3, we will lose the constraints and use a gradient descent method to further force the system into a local minimum⁹. This eliminates the possibility that the system simply converges to an unstable state in step 2 due to the imposed constraints.

By repeating step 2–4 for all possible sets of hinges that can be selected from each unit cell, all metastable states should be enumerated.

3.2.2 A case study: extruded truncated tetrahedron

The extruded truncated tetrahedron is one example which is known to have multiple metastable states, shown in figure 1. It is also reasonably simple so that we can enumerate all possible configurations using the procedure outlined above. The results shown below in figure 7 are metastable states found through simulation. By comparing further results from other methods with the results presented here, we can conclude whether they are at least as complete as this one.

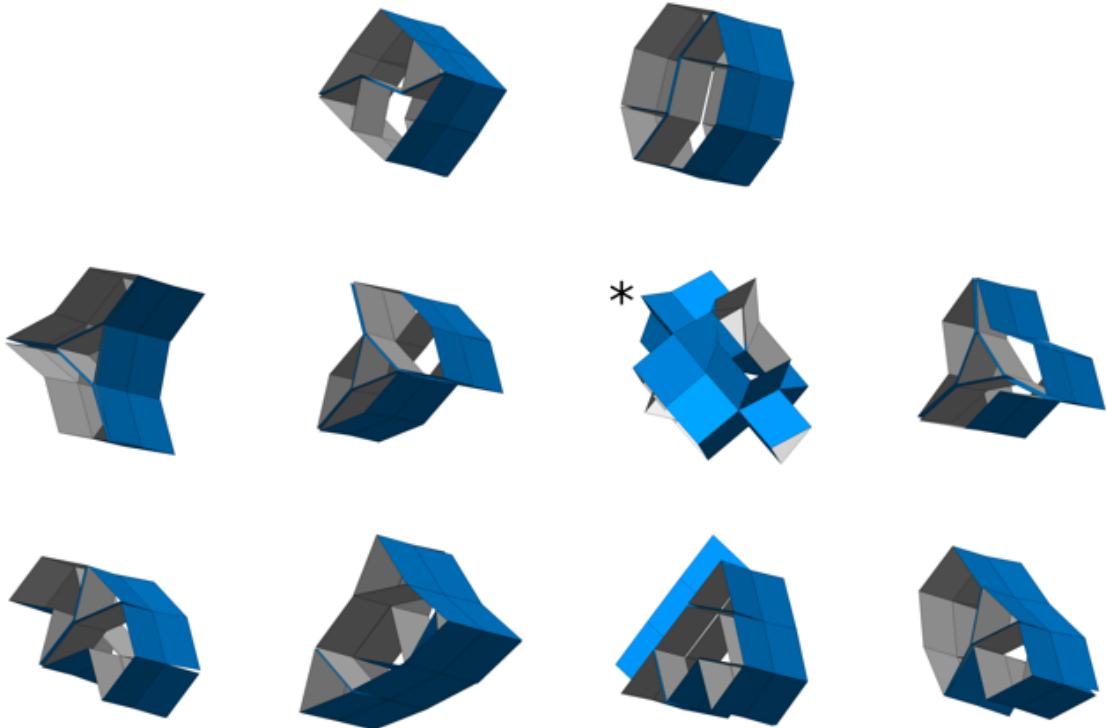


Figure 7: Metastable states of truncated tetrahedron unit cell obtained through brute force.
Duplicate metastable states are not shown.

⁹We opt for gradient descent rather than `fmincon` in this step in order to have better control over step sizes when close to the minima.

3.2.3 Feasibility of the brute force approach

Given H polyhedron hinges in a structure, the number of ways to select different sets is $C_0^H + C_1^H + C_2^H + \dots + C_{H-1}^H + C_H^H = 2^H$. The exact numbers are presented in table 1. As we can see, the number of possible ways to actuate a set of hinges increases exponentially with the total number of hinges. For a not-too-complex polyhedron such as cuboctahedron, exhausting all possibilities would take years to finish on a common desktop computer¹⁰.

Number of hinges actuated	Number of ways to actuate				
	tetrahedron 	cube 	truncated tetrahedron 	cuboctahedron 	...
0	$C_0^6 = 1$	$C_0^{12} = 1$	$C_0^{18} = 1$	$C_0^{24} = 1$	
1	$C_1^6 = 6$	$C_1^{12} = 12$	$C_1^{18} = 18$	$C_1^{24} = 24$	
2	$C_2^6 = 15$	$C_2^{12} = 66$	$C_2^{18} = 153$	$C_2^{24} = 276$	
3	$C_3^6 = 20$	$C_3^{12} = 220$	$C_3^{18} = 816$	$C_3^{24} = 2024$	
4	$C_4^6 = 15$	$C_4^{12} = 495$	$C_4^{18} = 3060$	$C_4^{24} = 10626$	
5	$C_5^6 = 6$	$C_5^{12} = 792$	$C_5^{18} = 8568$	$C_5^{24} = 42504$	
6	$C_6^6 = 1$	$C_6^{12} = 924$	$C_6^{18} = 18564$	$C_6^{24} = 134596$	
7	-	$C_7^{12} = 792$	$C_7^{18} = 31824$	$C_7^{24} = 346104$	
8	-	$C_8^{12} = 495$	$C_8^{18} = 43758$	$C_8^{24} = 735471$	
⋮	⋮	⋮	⋮	⋮	⋮
N	C_N^6	C_N^{12}	C_N^{18}	C_N^{24}	
total	$2^6 = 64$	$2^{12} = 4096$	$2^{18} = 262144$	$2^{24} = 16777216$...

Table 1: Number of possible ways to select actuated hinges for different polyhedra

4 Stochastic approach

Clearly, the attempt to enumerate everything using brute force will make it impossible for the author to graduate within normal temporal constraints. One of the reasons why the brute force method does not work is that the constraints result in many unphysical states. Therefore, as an improvement, we devised a procedure that combines stochastic optimisation, which amply explores the more feasible area of the entire configuration space by sifting through random samples, and deterministic optimisation, which then lets the system relax into a lower-energy state.

4.1 Metropolis-Hastings algorithm

We first introduce the Metropolis-Hastings algorithm below. The state of the system we study is described by a vector \mathbf{x} .

¹⁰It takes about 10 seconds to complete one round of optimisation for one hinge set. Completing all sets would take $10 \times 16777216 = 1.7 \times 10^8$ s = 5.3 yr.

1. Define temperature T of the system, and let $\beta = \frac{1}{T}$. A higher temperature allows the system to have more random motions so that it is more likely to enter states with higher energies. The probability of the system to be in a certain energy state $E(\mathbf{x}) = E_0$ follows the Boltzmann distribution, i.e., $P(E(\mathbf{x}) = E_0) \propto e^{-E_0\beta}$.
2. Starting from the initial configuration $\mathbf{x} = \mathbf{x}_0$ with $E(\mathbf{x}_0) = 0$, repeat the following procedure N times:
 - (a) Select a candidate solution by adding a small random perturbation¹¹ to each component x_i of \mathbf{x} :

$$(\mathbf{x}_{\text{can}})_i = x_i + \text{rand}(0, \delta)$$

where δ defines the maximum length of each random step.
 - (b) Compute the energy of the new candidate solution $E(\mathbf{x}_{\text{can}})$
 - (c) Compute the acceptance rate α , which is the ratio of the probability of the system in a state of $E(\mathbf{x}_{\text{can}})$ to the probability of the system in a state of $E(\mathbf{x})$:

$$\alpha = \frac{P(E(\mathbf{x}_{\text{can}}))}{P(E(\mathbf{x}))} = \frac{e^{-\beta E(\mathbf{x}_{\text{can}})}}{e^{-\beta E(\mathbf{x})}} = e^{-\beta(E(\mathbf{x}_{\text{can}}) - E(\mathbf{x}))}$$

- (d) Draw a new random number between 0 and 1:

$$r = \text{rand}(0, 1)$$

If $r < \alpha$, let $\mathbf{x} = \mathbf{x}_{\text{can}}$. Otherwise, let $\mathbf{x} = \mathbf{x}$.

In other words, the candidate solution is accepted with probability $\min\{1, \alpha\}$. In this way, when the candidate solution has a lower energy, we will always accept it, but when it has a higher energy, there is still a possibility to accept the worse solution, so the solver does not get stuck at that point.

4.2 Optimisation procedure

Our first goal is to select a good distribution of points in the configuration space that explore the space as much as possible and at the same time correspond to physically meaningful configurations. Given these points, we can then apply a deterministic optimisation method (such as gradient descent) to make each of them converge to its closest energy minimum. In this way, we will hopefully obtain most, if not all, of the metastable states without going through unphysical transformations. The procedure is outlined below.

1. An extruded polyhedron object is created, and represented by \mathbf{x} , a vector containing the coordinates of all vertices. The initial configuration is $\mathbf{x} = \mathbf{x}_0$.
2. The Metropolis-Hastings algorithm is applied for M times, alternating between a high and low temperature. We set N_1 to be the number of iterations in each leg. Therefore, the total number of Metropolis-Hastings iterations will be $M \cdot N_1$.
A higher temperature allows the system more random motion, so that it tends to be more “explorative” and cross energy barriers to reach metastable states; A lower temperature makes the system less likely to accept higher-energy solutions, and this more “conservative” tendency drives it to correct some unphysical features, such as over-stretched edges (see figure 8).

¹¹This random perturbation can be made to follow a pre-defined distribution, such as the uniform distribution, the normal distribution, the beta distribution, etc.

3. Out of the $M \cdot N_1$ iterations from step 2, select n of those with the largest angle deformation, and repeat the following for each such solution:¹²
 - (a) let \mathbf{x}_0 be one of the n solutions, and use it as the new starting point.
 - (b) Let $\nabla E(\mathbf{x})$ be the gradient of the energy function. Then, the direction opposite to $\nabla E(\mathbf{x})$, i.e., $-\nabla E(\mathbf{x})$, will always be the direction of fastest descent in energy.
 - (c) Repeat the following procedure N_2 times, or until otherwise specified:
 - i. Let the next iteration \mathbf{x}_{k+1} be

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \cdot \nabla E(\mathbf{x}_k)$$

where h is the step size.

- ii. Calculate the energy difference d between two consecutive iterations

$$d = |E(\mathbf{x}_{k+1}) - E(\mathbf{x}_k)|$$

If $d < \epsilon$, exit the loop. Here ϵ is a pre-defined value for tolerance.

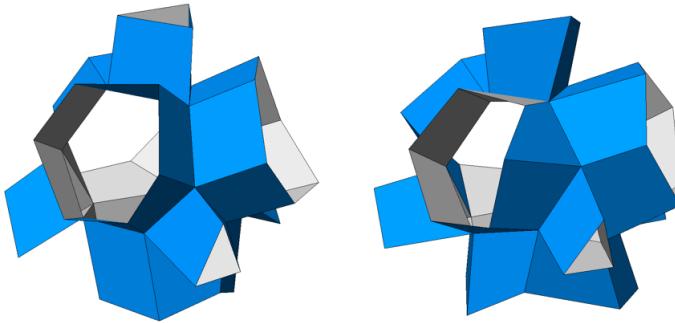


Figure 8: Random movements in stochastic methods may result in unphysical configurations.

4.3 Results from stochastic approach

We ran the above procedure 1500 times on the extruded truncated tetrahedron, and observed no instance where the system converged to a metastable state. The parameters used were $k_{\text{hinge}} = 1$, $k_{\text{face}} = 100$, $k_{\text{edge}} = 100$, $\beta_0 = 0.1$, $\beta_1 = 0.5\delta = 0.05$, $N_1 = N_2 = 1000$ (arbitrary units).

With limited computing power, we were unable to explore the entire configuration space of truncated tetrahedron unit cell, and didn't encounter any meaningful results. What we often observed is that the unit cell seems to be almost crossing over an energy barrier and moving towards a metastable state, but soon due to the random nature of the process, it moves towards another direction, rendering it unable to cross the energy barrier between two stable states.

¹²The rest of the solutions are discarded because up until now, we are only trying to select good points in the configuration space that are close to some energy minima, while at the same time do not represent unphysical structures. Many of the iterations are very similar to each other because of the inherent symmetry of the unit cell; by selecting the ones with largest angle deformations, we are more likely to get rid of the similar results.

The balance here is hard to achieve: in order to cross the energy barrier, we need the unit cell to enter a somewhat unphysical configuration. This can be done by increasing the temperature T of the system, or the maximum step size δ . But this also makes the final result less likely to be a physical one. We are thus faced with either having jiggling movements around the original configuration, or drastic changes in the unit cell that often end up being unphysical.

5 Duplication Reduction

The stochastic approach is an attempt to improve the brute force method by reducing the occurrences of unphysical states. Although not entirely successful, it naturally points us to another direction of improvement: reducing repetitions.

Since all polyhedra under consideration are highly symmetrical, many hinge sets from the brute force approach are equivalent with each other, either through rotation or mirror symmetry, such as the example shown in figure 9.

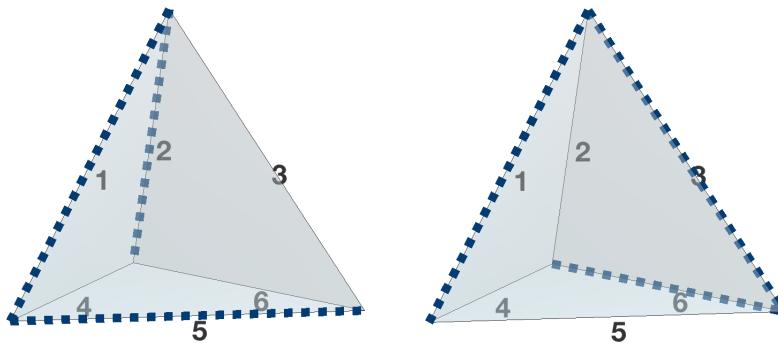


Figure 9: Two equivalent sets of hinges that appear in the brute force method.

Left: hinges selected are 1, 2, 5. Right: hinges selected are 1, 3, 6.

They are related by rotational symmetry.

The final approach proposed here makes use of the intrinsic symmetry in each polyhedron, thereby reducing the size of the problem. This can be done by mapping polyhedra to more tangible mathematical objects, such as a directed graph¹³. Further, by using the graph as an aid, we devise an algorithm that measures the “distance” between any pair of hinges in the polyhedra, which then helps us to select non-equivalent sets of hinges.

5.1 Conversion to graphs

We begin by mapping each hinge in the polyhedron to a node in the graph, numbered from 1 to H . An arc is constructed from node i to node j if their corresponding hinges in the polyhedron I and J are incident on one vertex, and I can be rotated clockwise to J without crossing another hinge when viewed from outside of the polyhedron.

¹³See appendix B about why an undirected graph is not suitable.

In the case where two types of hinges exist on a polyhedron¹⁴, (for example, in truncated tetrahedron, some hinges are between two hexagons, and some are between a triangle and a hexagon), we assign them different “flavours”, represented as different numerical attributes of the graph nodes. A truncated tetrahedron has two flavours, so we assign 1 to the triangle-hexagon nodes, and 2 to the hexagon-hexagon nodes (the order is arbitrary). Examples of the directed graphs for tetrahedron and truncated tetrahedron are shown in figure 10.

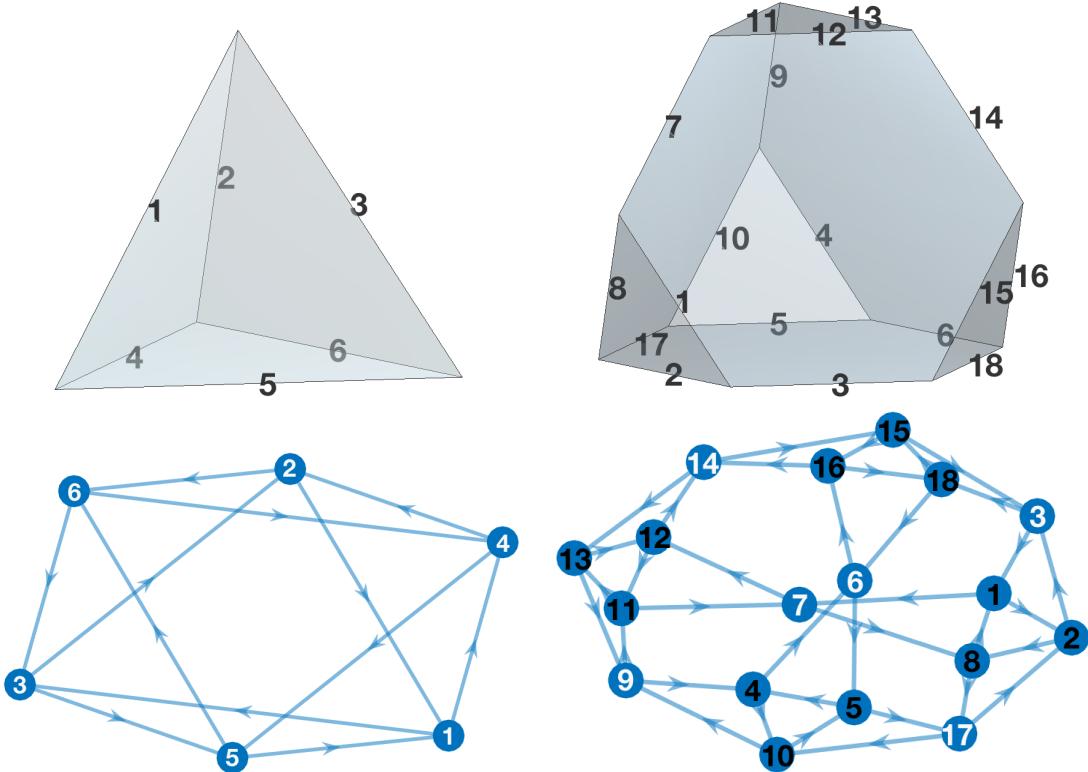


Figure 10: Different polyhedra mapped onto directed graphs.

Left: tetrahedron; Right: truncated tetrahedron. Directions are as indicated by the arrows. Different “flavours” of the truncated tetrahedron graph nodes are marked in different colors:
black ones correspond to hinges between two hexagons, and
white ones between one hexagon and one triangle.

5.2 Selection of nodes

Based on the optimisation procedure presented in section 3.2.1, the only modification we need is to reduce the duplicate selections as much as possible. This is achieved through selecting subgraphs from the graphs created as above. Each hinge set with l hinges will correspond to a subgraph with l nodes. Thus, l is defined as the length of such a set.

¹⁴Not to be confused with the “polyhedron hinge” and the “extruded hinge” types. Here the different flavours arise from the geometrical properties of the polyhedra themselves, and have nothing to do with the extruded part.

5.2.1 Example hinge sets enumerated by hand

Before we shift our focus to graphs entirely, we first put forward a series of sketches that show the complete set of hinges for different geometry at different values of l . Some of them are shown in figure 11, 13, 12. This is used to confirm that no solutions are left out in the new algorithms developed in section 5.2.4.

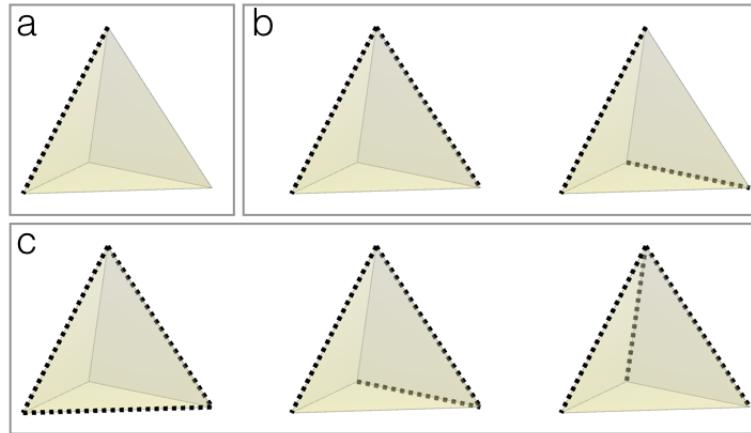


Figure 11: All possible ways to select 1, 2, or 3 hinges from tetrahedron.
(a): $l = 1$; (b): $l = 2$; (c): $l = 3$. Selected hinges are highlighted in dashed lines.

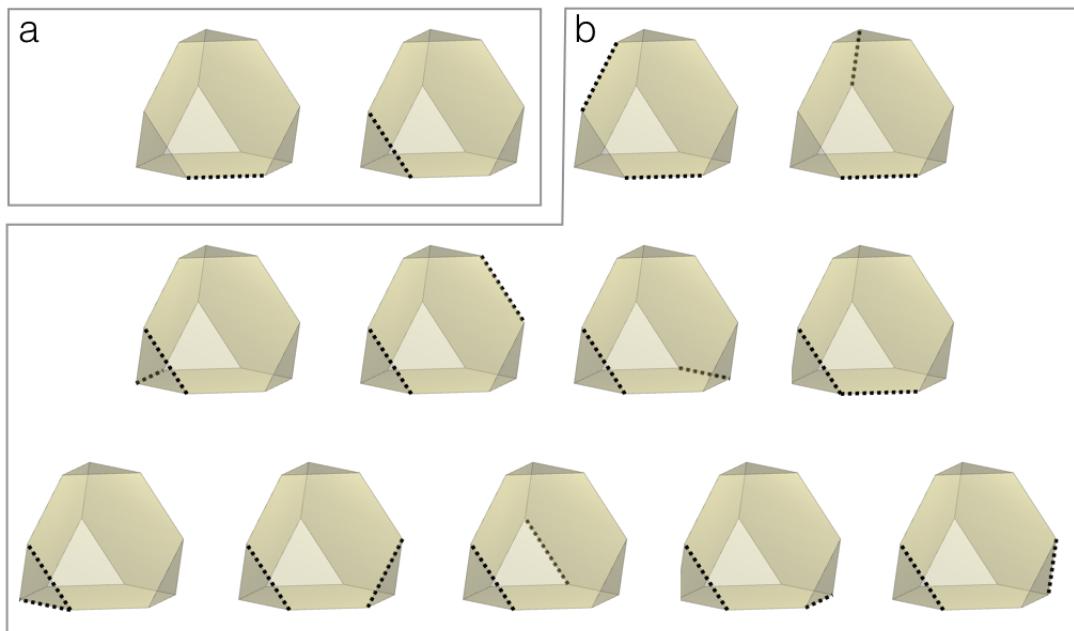


Figure 12: All possible ways to select 1 or 2 hinges from truncated tetrahedron.
(a): $l = 1$; (b): $l = 2$.

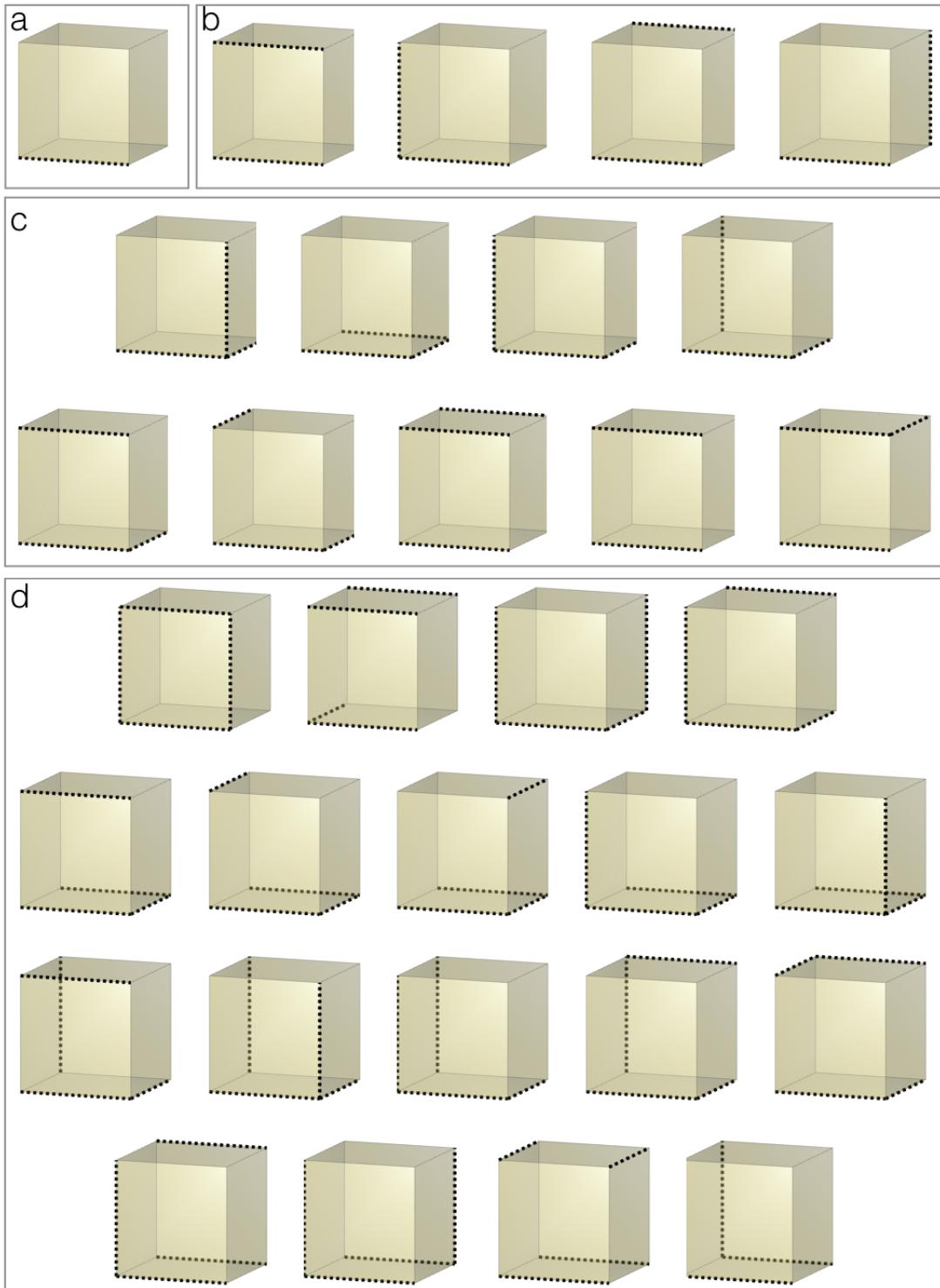


Figure 13: All possible ways to select 1, 2, 3, or 4 hinges from cube.
 (a): $l = 1$; (b): $l = 2$; (c): $l = 3$; (d): $l = 4$.

While making these sketches, one strategy is to build larger hinge sets from smaller ones. Let us use the cube as an example to illustrate.

First, all hinges on the cube are of the same flavour, so when choosing sets with $l = 1$, there is

only one possibility, and choosing any hinge will give an equivalent result. We can choose the hinge labeled “1”, as shown in figure 14.

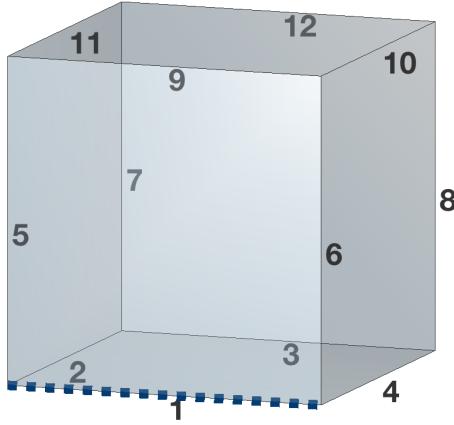


Figure 14: Choosing one hinge from the cube.

The chosen hinge is highlighted in dashed line. Note that this is the same as figure 13 (a).

When choosing hinge sets with $l = 2$, we only need to extend the previous result of $l = 1$. Since there are only 4 ways of adding another hinge to create a selection which is unique after symmetry transformations, we obtain 4 results for $l = 2$, as shown in figure 15. This is again the same as figure 13 (b).

Similarly, this procedure can be applied for $l = 3, 4, \dots$.

For a human brain, the complexity rises very quickly as we consider larger l , or more complicated polyhedra. Further, it is still a case-specific approach and cannot be readily generalised. Our hope lies in distilling a selection criterion from the corresponding graph properties that can be understood by a computer. Afterwards, by using the same strategy of building larger selected hinge sets on smaller ones, we may generate this for any convex polyhedron.

5.2.2 The “distance” matrix for determining unique hinge combinations

Selection criterion for identifying unique hinge sets requires us to find a mathematical representation for the geometry between different nodes. A first guess is that such a criterion might come from the adjacency matrix, which delineates how the nodes in a graph are connected. For example, the adjacency matrix for the tetrahedron graph shown in figure 5.1 is

$$A_{\text{tetrahedron}} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

where entry $A_{i,j}$ represents the existence of the arc from node i to node j .

Since the goal is to select subsets of the nodes, it is worthwhile to investigate what their corresponding submatrices represent. Can we tell how the nodes in a submatrix are connected (i.e.,

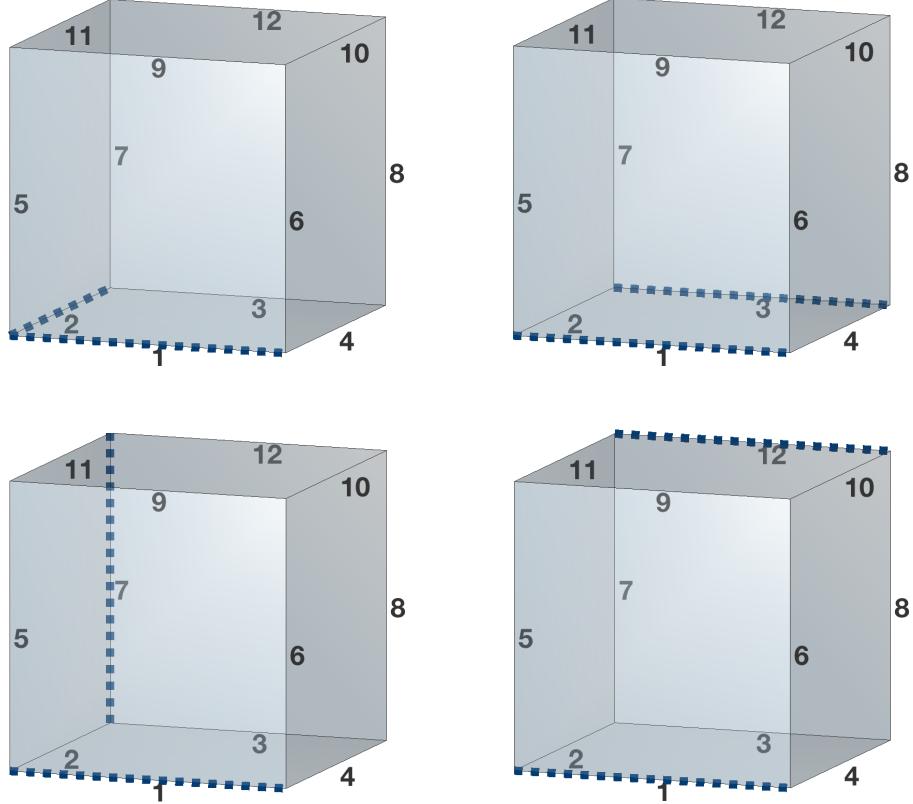


Figure 15: Four different cases of choosing two hinges from the cube.
Note that these are the same selections as in figure 13 (b).

where the hinges are located with respect to each other)? The answer is most likely no, because adjacency matrices only tell us which nodes are immediate neighbours. For example, in figure 15, the four selections correspond to the following four submatrices:

$$A = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

where the full adjacency matrix for the cubic graph is

$$A_{\text{cube}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

When selecting hinges from polyhedra, we also need to know the relative position of hinges that are farther apart from each other, but this information is not explicitly available from the adjacency matrix.

Between any pair of nodes, we define a measure of “distance” which is a unique quantity that differentiates all structurally different pairs. From the directed graph we’ve constructed, there are multiple pathways¹⁵ between each pair of nodes, and each of these paths could be labeled by the flavours of the nodes that lie on the path. For example, in the left panel of figure 10, to travel between node 1 and 2, the possible simple paths are:

```

1 → 4 → 2
1 → 3 → 2
1 → 4 → 5 → 6 → 3 → 2
1 → 3 → 5 → 6 → 4 → 2
2 → 1
2 → 6 → 4 → 5 → 1
2 → 6 → 3 → 5 → 1

```

Since tetrahedron has only one type of hinges, we can let the nodes all have the same flavour, say 1. Then, the above sequences can be written as sequences of node flavours:

1 → 4 → 2	111
1 → 3 → 2	111
1 → 4 → 5 → 6 → 3 → 2	111111
1 → 3 → 5 → 6 → 4 → 2	111111
2 → 1	11
2 → 6 → 4 → 5 → 1	11111
2 → 6 → 3 → 5 → 1	11111

Treating these labeled paths as numbers, we can define the “shortest path” between node 1 and 2 to be 11. If we apply the same procedure to all pairs of nodes, we obtain a “distance matrix”

$$D_{\text{tetrahedron}} = \begin{pmatrix} 1 & 11 & 11 & 11 & 11 & 1111 \\ 11 & 1 & 11 & 11 & 1111 & 11 \\ 11 & 11 & 1 & 1111 & 11 & 11 \\ 11 & 11 & 1111 & 1 & 11 & 11 \\ 11 & 1111 & 11 & 11 & 1 & 11 \\ 1111 & 11 & 11 & 11 & 11 & 1 \end{pmatrix}$$

Similarly, we can obtain the same distance matrix for a truncated tetrahedron, where node flavour 1 represents triangle-hexagon hinges, and 2 hexagon-hexagon hinges.

¹⁵Here we only consider simple paths, which are paths that do not cross over each other. Without this restriction, there would be infinitely many paths between any two nodes.

$$D_{\text{truc.tetrahedron}} = \begin{pmatrix} 1 & 11 & 12 & 1112111 & \dots \\ 11 & 1 & 12 & 11121 & \dots \\ 12 & 12 & 2 & 11212 & \dots \\ 1112111 & 11121 & 11212 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

5.2.3 Submatrices of the “distance matrix”

Moreover, submatrices of the distance matrix also contain information about how each node is positioned relative to one another. Therefore, we have further reduced the problem to selecting distinct submatrices from the distance matrix.

But how can one tell if two submatrices represent equivalent selections? In the example of figure

9, the submatrices are $A = \begin{pmatrix} 1 & 11 & 11 \\ 11 & 1 & 1111 \\ 11 & 1111 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 11 & 1111 \\ 11 & 1 & 11 \\ 111 & 11 & 1 \end{pmatrix}$, respectively.

Since they represent the same selection only with nodes shifted to different locations, A and B should be related by a permutation matrix. And indeed, we have

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 11 & 11 \\ 11 & 1 & 1111 \\ 11 & 1111 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 11 & 1111 \\ 11 & 1 & 11 \\ 111 & 11 & 1 \end{pmatrix}$$

And the permutation matrix is $P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ such that $PAP^{-1} = B$.

This indicates that when faced with two selections, one method of determining their equivalence is by looking at their corresponding submatrices. If the two are similar and the transformation for the similarity is a permutation matrix, then we can confirm that they represent geometrically equivalent situations, so that we only need to keep one set of selections, thereby reducing the total number of situations we need to consider. Further, similar matrices have the same eigenvalues, and this can be used to simplify the selection procedure.

5.2.4 Procedure of selection of unique hinge sets

Equipped with the above observations, we can start to develop an algorithm to select all the hinge sets that are not equivalent to each other.

Given the graph as constructed in section 5.1, we give each node a label from 1 to H , where H is the total number of hinges in the geometry (or equivalently the total number of nodes in the graph). The distance matrix D is then computed. Also, let f be the number of flavours a given geometry. We define l to be the number of nodes being selected for a set s , and S_l is a set whose elements are hinge sets with l elements.

We break down the algorithm into different cases:

- $l = 0$ or $l = H$:
Trivial.
- $l = 1$:
If $f = 1$, all nodes are of the same flavour, then simply select the node labeled with the smallest number, and return as the result
Otherwise, $f \geq 2$, which means there are more than one flavours present. In this case, select the nodes labeled with the smallest number for each flavour, and return as the result.
- $2 \leq l \leq \frac{H}{2}$:
Assume all hinge sets with length $l - 1$ are given, and let \mathcal{S}_{l-1} be the set of all such hinge sets.
To generate \mathcal{S}_l , i.e., all hinge sets with l elements, first initialise an empty variable \mathcal{C}_{eig} as a cache to store eigenvalues of submatrices.
Then, repeat the following for each hinge set $s \in \mathcal{S}_{l-1}$:

1. Let n_1, n_2, \dots, n_{l-1} be all the nodes in s .
Take the $(n_1)^{\text{th}}, (n_2)^{\text{th}}, \dots, (n_{l-1})^{\text{th}}$ column of the distance matrix D .
2. Among these columns, extract the rows with unique values. To avoid double counting, we also exclude the first n_{l-1} rows. For instance, once $\{1, 2\}$ is counted, we would not go backwards and count $\{2, 1\}$ again.
3. For each of the remaining rows, repeat the following:
 - (a) Find the index of the current row, let it be m . Then $s_{\text{can}} = \{n_1, n_2, \dots, n_{l-1}, m\}$ form a new candidate hinge set.
 - (b) Obtain the submatrix consisting of nodes in s_{can} , and compute its set of eigenvalues Λ .
If the eigenvalues are unique, the submatrix is not similar to what had been found previously. Thus the submatrix represents a unique hinge set. Therefore, if $\Lambda \notin \mathcal{C}_{\text{eig}}$, add Λ to \mathcal{C}_{eig} , and add s_{can} to \mathcal{S}_l .
Otherwise, $\Lambda \in \mathcal{C}_{\text{eig}}$. In this case, we skip to the next iteration.

Return \mathcal{S}_l as the result.

- $\frac{H}{2} \leq l \leq H - 1$:¹⁶
Selecting s out of H elements is the same as choosing those that are not chosen when selecting $H - s$ elements. Hence, to generate \mathcal{S}_l , we only need to repeat the following for each $s \in \mathcal{S}_{H-l}$:

1. Take the compliment set $\bar{s} = U \setminus s$, where $U = 1, 2, \dots, H$ is the set of all nodes.
2. Add \bar{s} to \mathcal{S}_l .

Return \mathcal{S}_l as the result.

For example, in the case of tetrahedron, the selections made are:

¹⁶Note that $l = \frac{H}{2}$ is considered again to account for the “negative part” of itself, since they are generally different hinge sets.

l	Set(s) of nodes/hinges selected
0	-
1	{1}
2	{1, 2}, {1, 6}
3	{1, 2, 3}, {1, 2, 5}, {3, 4, 6}, {4, 5, 6}
4	{2, 3, 4, 5}, {3, 4, 5, 6}
5	{2, 3, 4, 5, 6}
6	{1, 2, 3, 4, 5, 6}

Table 2: Hinge/Node selection for tetrahedron
Here l is the number of hinges/nodes selected.

5.3 Optimisation procedure

Once the non-repetitive selections are obtained from the algorithm laid out above, we simply need to apply the procedure as described in section 3.2.1 in order to find whether each given set of hinge closures results in a metastable state.

5.4 Results from duplication reduction

Table 3 below shows the reduced number of hinge sets for 4 geometries. As we can see, compared to the brute force approach, there is a 1–2 orders of magnitude reduction in the number of cases we need to verify with this method.

Number of hinges actuated	Number of ways to actuate				
	tetrahedron	cube	truncated tetrahedron	cuboctahedron	...
0	1 (1)	1 (1)	1 (1)	1 (1)	
1	1 (6)	1 (12)	2 (18)	1 (24)	
2	2 (15)	4 (66)	11 (153)	6 (276)	
3	4 (20)	9 (220)	44 (816)	24 (2024)	
4	2 (15)	18 (495)	146 (3060)	125 (10626)	
5	1 (6)	24 (792)	381 (8568)	539 (42504)	
6	1 (1)	60 (924)	802 (18564)	2070 (134596)	
7	-	24 (792)	1329 (31824)	5928 (346104)	
8	-	18 (495)	1794 (43758)	13790 (735471)	
⋮	⋮	⋮	⋮	⋮	⋮
total	12 (64)	174 (4096)	12950 (262144)	390636 (16777216)	...

Table 3: Number of hinge sets after reduction.
Numbers from brute force approach are listed in brackets for comparison.

Also, we tested the found hinge sets on truncated tetrahedron, and the results are shown in figure 16. The found metastable states are in good agreement with what we see in the brute force

approach (figure 7) discussed in section 3.2.2.

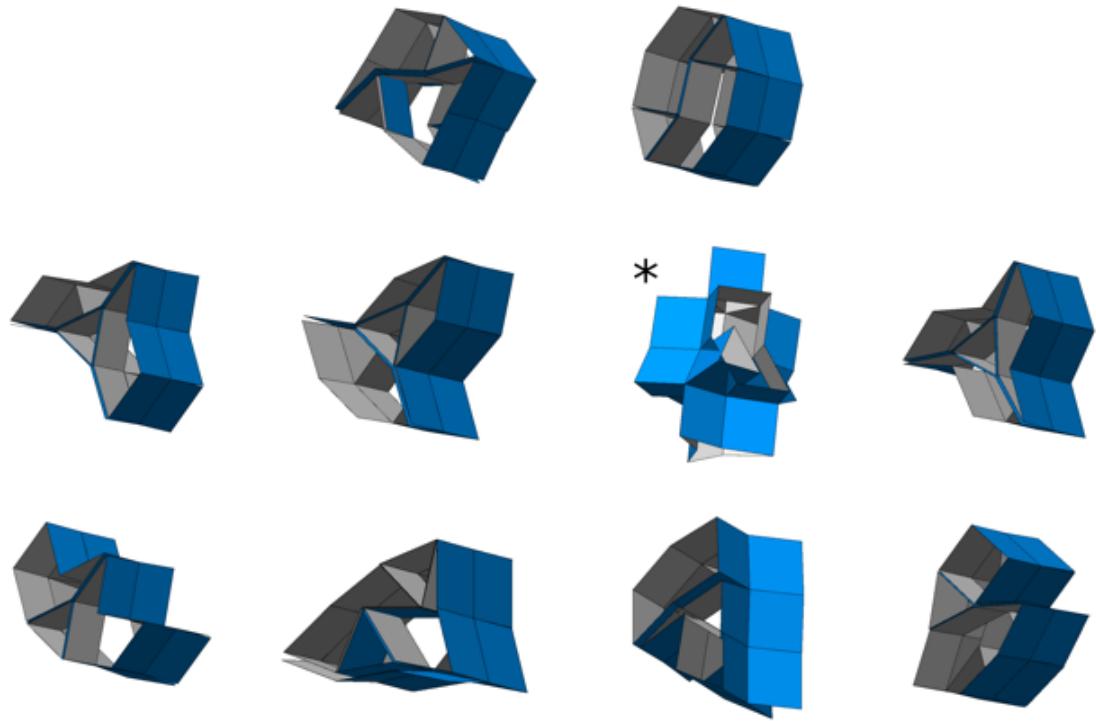


Figure 16: Metastable states found by actuating hinge sets in the reduced list.

These metastable states can also be replicated in physical models made with cardboard, as shown in figure 17. For details of fabrication, see appendix C.

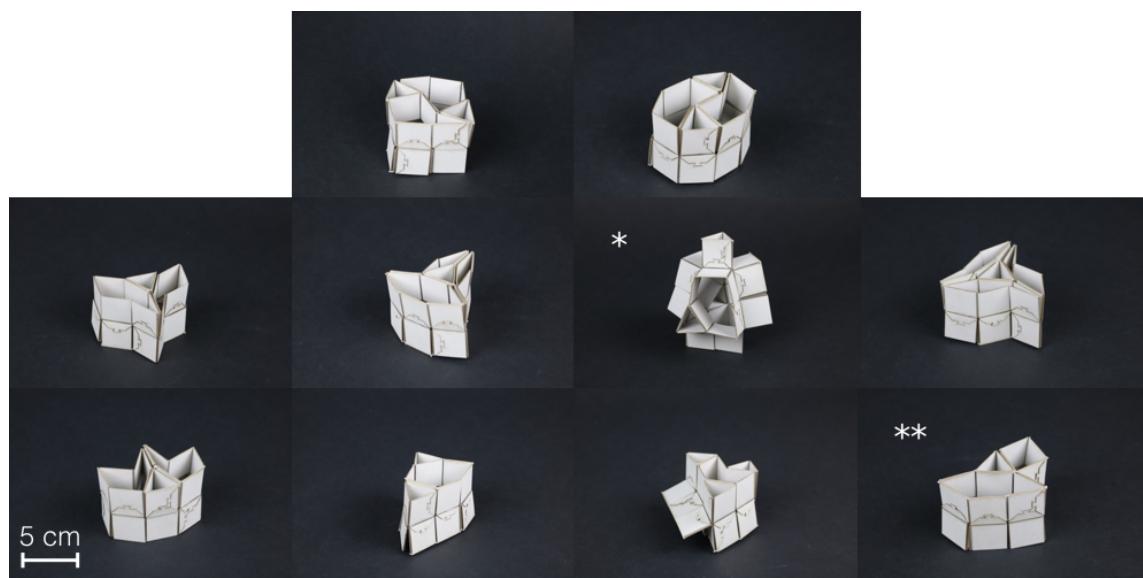


Figure 17: Metastable states found in figure 7 and 16 replicated with cardboard prototype.

We also found another metastable state that is not reproducible with cardboard models, where an extruded triangle completely pops through to the opposite side, as shown in figure 18. This is because the cardboard model has stiffer faces than what the simulation accounts for. Overvelde et al. also showed that in similar cardboard prototypes, face bending is more energetically costly than hinge bending [12]. In addition, the simulated structures are allowed to cross higher energy barriers without breaking apart, as they would in real life. Theoretically, the same situation could happen to all four triangles components. However, this also implies a even higher energy barrier to cross, and as a result we did not encounter any other similar cases.

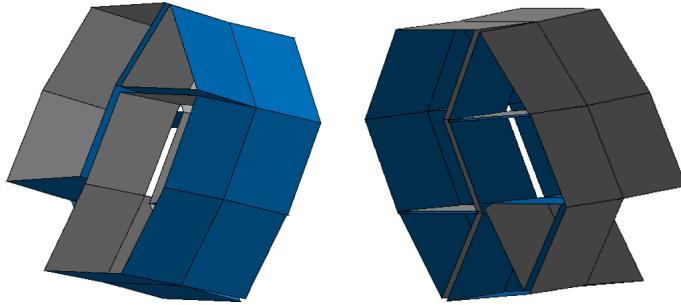


Figure 18: One metastable state with an extruded triangle popped through to the opposite side
This is otherwise the same configuration as the top right one in figure 16.

6 Discussions and Conclusion

This study presents three different approaches aimed at identifying metastable states in a class of origami-inspired metamaterial unit cells. The stochastic approach is unable to explore the entire configuration space. The brute force approach in theory yields the most complete results; however, it also produces many duplicates because of the symmetry in the unit cells. Based on that observation we introduced new algorithms to eliminate the duplicates, producing results that are in good agreement with those from the brute force approach.

Further reduction can still be made to reduce computation time. Although the hinge sets have been reduced to a minimum, among these sets many still give the same end results (see figure 19). It would be worthwhile to develop an algorithm to go through all the converged results of different actuation patterns, and cull out the ones that are different from each other. This could be done by marking closed hinges in different metastable states in their corresponding graphs, and comparing these graphs to determine whether they are equivalent to each other.

This study does not give any examples of geometries with more vertices than the cuboctahedron, because the added complexity is still difficult to handle. The reduced number of hinge sets for cuboctahedron (390636 to be precise, see table 3) still amounts to around 2 months of CPU time. It might be useful to keep a list of converged metastable states while running optimisation on each hinge set, so that we can stop the current process once we detect that it will result in one of the existing metastable states, and move on to the next hinge set. Of course, upgraded computing power will also solve this problem.

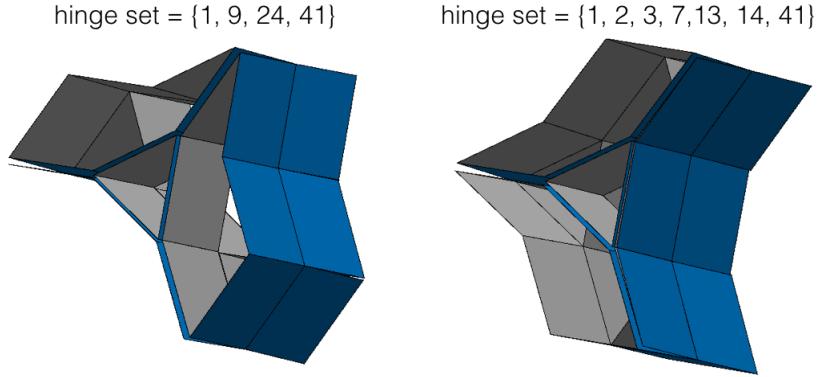


Figure 19: Two different hinge sets in the reduced list converge to the same metastable state.

Artifacts of optimisation algorithms also plays a role in which metastable states can be uncovered. In the gradient descent method implemented for this study, a penalty function is used to prevent hinge angles from going beyond the range of $[0, \pi]$. When calculating the total energy at each step, an extra amount E_{pen} is added, which equals $\sum_{i=1}^H \left(\max \left\{ 0, \frac{1}{2}(\theta_i - 0.95\pi)(\theta_i + 0.95\pi) \right\} \right)$. However, the parameter in the penalty function is not perfectly tuned for every unit cell, and as a result we still find some cases with self-contact, as can be seen in figure 7 (*) and 16 (*). The MATLAB built-in optimisation solver `fmincon` excels at handling constraints in this case, but it does not allow a user-defined (maximum) step size. This may cause the solver to overshoot and miss the energy minima when it is nearby, as we discovered in the beginning of this study. A reliable optimisation algorithm that respects all the constraints is needed to deal with these artifacts.

The majority of the metastable state for the truncated tetrahedron unit cell we found have unbent faces, even though crossing the energy barrier requires bending in the faces. One exception is figure 17 (*). Its simulation counterpart (figure 16 (*)) without correct boundary conditions has the triangular components crossing over other faces, as discussed above. This implies that in this case, the energy minimum lies on the boundary of the configuration space, and that face bending is required for the unit cell to stay in this configuration. This is similar to multistability in the Miura-fold, where popping a vertex always results in bent faces [15].

The assumptions that all metastable states have completely closed polyhedron hinges and can be characterised by them also needs further scrutiny. We found one counterexample to this assumption, where the state in figure 17 (**) can yet again be popped into a different metastable state, as shown in figure 20. This cannot be described in the current model because the two configurations depicted below have the same set of closed polyhedron hinges. This should be taken into account in further investigations of similar structures.

The ratio of spring stiffness $\frac{k_{\text{face}}}{k_{\text{hinge}}}$ determines whether the unit cell is more likely to deform through hinges or faces. This ratio also determines the “route” in the configuration space in which energy minima are found. The situation in figure 18 is such an example. Therefore, the parameters in this simulation need to be tuned in order to study different physical systems. A unit cell with stiffer hinges, for example, may have less metastable states found in this study. A finite thickness in the faces prevents the hinges from closing completely, and this also interferes with the search for metastable states.

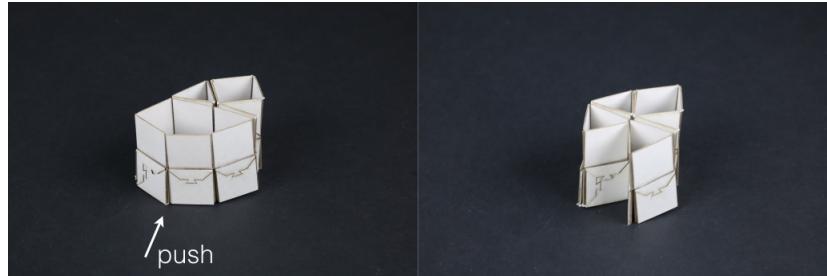


Figure 20: Two metastable configurations with the same set of hinges closed.

Not all metastable states found in the unit cell can be transferred to **space-filling tilings**, because many of the metastable configurations do not preserve the symmetries of the original geometry. Further research is needed to study how these unit cells can be assembled together so that the tessellation is still multistable. The current study can be extended to include groups of geometries together that form space-filling tessellations (see appendix E) with periodic boundary conditions. Distributed actuation as proposed in [11] can be used to actuate the meta-stable metamaterials.

We hope that this study provides the necessary information and inspirations for new design strategies for mechanical metamaterials. Multistabilities in the unit cells we studied could also be adapted for designing reconfigurable devices or modular robots. We would also like to see how these structures keep up their multisatibility at various length scales: at several meters where gravity plays a more significant role, as well as at nano- and micro-scale where material/surface properties will no longer be negligible.

Bibliography

- [1] CONWAY, J. H., BURGIEL, H., AND GOODMAN-STRAUSS, C. *The Symmetries of Things*. CRC Press, Florida, 2008.
- [2] CRESCENZI, P., GOLDMAN, D., PAPADIMITRIOU, C., PICCOLBONI, A., AND YANNAKAKIS, M. On the Complexity of Protein Folding. *Journal of Computational Biology* 5, 3 (1998), 423–465.
- [3] FRAENKEL, A. S. Complexity of protein folding. *Bulletin of Mathematical Biology* 55, 6 (1993), 1199–1210.
- [4] GOLDMAN, F. Using the snapology technique to teach convex polyhedra. In *Fifth International Meeting of Origami Science, Mathematics, and Education*. CRC Press, Florida, 2011, pp. 99–110.
- [5] HAARIO, H., SAKSMAN, E., AND TAMMINEN, J. An Adaptive Metropolis Algorithm. *Source: Bernoulli* 7, 2 (2001), 223–242.
- [6] HOLDEN, A. *Shapes, space, and symmetry*. Columbia University Press, New York, 1971.
- [7] JENSEN, PAUL A AND BARD, J. F. Algorithms for Constrained Optimization. In *Operations research models and methods*. John Wiley & Sons Incorporated, 2003, ch. Appendix A.
- [8] MARTÍ, R. Multi-start methods. In *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds. Kluwer Academic Publishers, 2003, ch. 12, pp. 355–368.
- [9] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. Equation of state calculations by fast computing machines. *Journal Chemical Physics* 21, 6 (1953), 1087–1092.
- [10] OSMAN, I. H., AND KELLY, J. P., Eds. *Meta-Heuristics: Theory and applications*. Kluwer Academic Publishers, Massachusetts, 1996.
- [11] OVERVELDE, J. T., DE JONG, T. A., SHEVCHENKO, Y., BECERRA, S. A., WHITESIDES, G. M., WEAVER, J. C., HOBERMAN, C., AND BERTOLDI, K. A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees of freedom. *Nature Communications* 7 (2016), 10929.
- [12] OVERVELDE, J. T. B., WEAVER, J. C., HOBERMAN, C., AND BERTOLDI, K. Rational design of reconfigurable prismatic architected materials. *Nature* 541, 7637 (2017), 347–352.
- [13] PAPALAMBROS, P. Y., AND WILDE, D. J. *Principles of optimal design: modeling and computation*. Cambridge University Press, Cambridge, 2000.
- [14] REIS, P. M., JAEGER, H. M., AND VAN HECKE, M. Designer Matter: A perspective. *Extreme Mechanics Letters* 5 (2015), 25–29.
- [15] SILVERBERG, J. L., EVANS, A. A., MCLEOD, L., HAYWARD, R. C., HULL, T., SANTANGELO, C. D., AND COHEN, I. Using origami design principles to fold reprogrammable mechanical metamaterials. *Science* 345, 6197 (2014), 647–650.
- [16] SILVERBERG, J. L., NA, J.-H., EVANS, A. A., LIU, B., HULL, T. C., SANTANGELO, C. D., LANG, R. J., HAYWARD, R. C., AND COHEN, I. Origami structures with a critical transition to bistability arising from hidden degrees of freedom. *Nature materials* 14, 4 (2015), 389–393.

- [17] SNYMAN, J. A., AND FATTI P., L. A Multi-Start Global Minimization Algorithm with Dynamic Search Trajectories. *Journal of Optimization Theory and Applications* 54, 1 (1987), 121–141.
- [18] TALBI, E.-G. *Metaheuristics: From design to implementation*. John Wiley & Sons, Inc, New Jersey, 2009.
- [19] WAITUKAITIS, S., MENAUT, R., CHEN, B. G. G., AND VAN HECKE, M. Origami multi-stability: From single vertices to metasheets. *Physical Review Letters* 114, 5 (2015), 2–6.
- [20] YASUDA, H., AND YANG, J. Reentrant origami-based metamaterials with negative Poisson’s ratio and bistability. *Physical Review Letters* 114, 18 (2015), 1–5.

Appendix

A Energy as a function of node coordinates

A.1 Vertices, edges, hinges, and faces

Let $\mathbf{x}_\nu = \begin{pmatrix} x_\nu \\ y_\nu \\ z_\nu \end{pmatrix}$ be the coordinate of vertex ν , where $\nu \in \{1, 2, \dots, V\}$, and V is the total number of vertices. We can then store the coordinates of all vertices in one $1 \times 3V$ vector

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{3V-2} \\ u_{3V-1} \\ u_{3V} \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_V \\ y_V \\ z_V \end{pmatrix}$$

Let E be the total number of **edges**, then the length of the k^{th} edge is $e_k = \|\mathbf{x}_{k(1)} - \mathbf{x}_{k(2)}\|$, where $\mathbf{x}_{k(1)}$ and $\mathbf{x}_{k(2)}$ are the two end nodes of this edge.

Then, we have

$$\begin{aligned} e_k &= \left\| \begin{pmatrix} x_{k(1)} \\ y_{k(1)} \\ z_{k(1)} \end{pmatrix} - \begin{pmatrix} x_{k(2)} \\ y_{k(2)} \\ z_{k(2)} \end{pmatrix} \right\| = \left\| \begin{pmatrix} u_{3k(1)-2} \\ u_{3k(1)-1} \\ u_{3k(1)} \end{pmatrix} - \begin{pmatrix} u_{3k(2)-2} \\ u_{3k(2)-1} \\ u_{3k(2)} \end{pmatrix} \right\| \\ &= \sqrt{(u_{3k(1)-2} - u_{3k(2)-2})^2 + (u_{3k(1)-1} - u_{3k(2)-1})^2 + (u_{3k(1)} - u_{3k(2)})^2} \quad (5) \end{aligned}$$

And the vector containing the lengths of all edges is $\mathbf{e} = (e_1 \ e_2 \ e_3 \ \dots \ e_E)^T$, which is a function of \mathbf{u} .

Let H be the total number of **hinges**. The k^{th} hinge angle θ_k can be calculated via the normal vector of its two adjacent faces. Take four vertices from the two adjacent faces $\mathbf{x}_{k(i)}, \mathbf{x}_{k(ii)}, \mathbf{x}_{k(iii)}, \mathbf{x}_{k(iv)}$, and calculate the vectors indicated in 21:

$$\begin{aligned} \mathbf{a} &= \mathbf{x}_{k(ii)} - \mathbf{x}_{k(i)} \\ \mathbf{b} &= \mathbf{x}_{k(iii)} - \mathbf{x}_{k(ii)} \\ \mathbf{c} &= \mathbf{x}_{k(iv)} - \mathbf{x}_{k(i)} \\ \hat{\mathbf{n}}_1 &= \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|} \\ \hat{\mathbf{n}}_2 &= \frac{\mathbf{a} \times \mathbf{c}}{\|\mathbf{a} \times \mathbf{c}\|} \end{aligned}$$

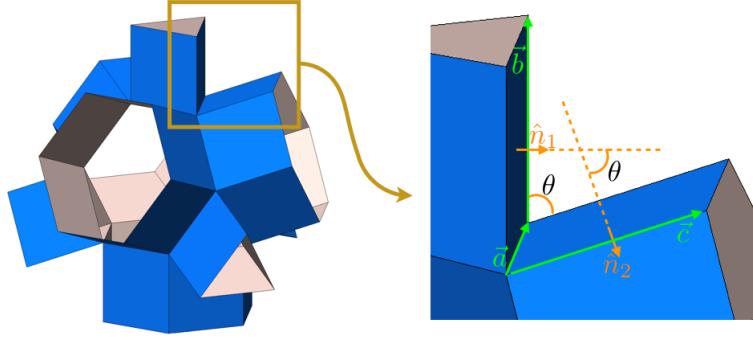


Figure 21: Hinge angle θ can be obtained from vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} .

where $\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2$ are the normal vectors of the two adjacent faces. Let $\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$ be the normal vector along the axis, and we have

$$\begin{aligned}\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2 &= \|\hat{\mathbf{n}}_1\| \cdot \|\hat{\mathbf{n}}_2\| \cdot \cos \theta_k = \cos \theta_k \\ \hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2 &= \|\hat{\mathbf{n}}_1\| \cdot \|\hat{\mathbf{n}}_2\| \cdot \sin \theta_k \cdot \hat{\mathbf{a}} = \hat{\mathbf{a}} \sin \theta_k\end{aligned}$$

Therefore,

$$\begin{aligned}\cos \theta_k &= \hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2 \\ \sin \theta_k &= (\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2) \cdot \hat{\mathbf{a}} \\ \tan \theta_k &= \frac{(\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2) \cdot \hat{\mathbf{a}}}{\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2} \\ \theta_k &= \arctan \left(\frac{(\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2) \cdot \hat{\mathbf{a}}}{\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2} \right)\end{aligned}\tag{6}$$

Note that the vector containing all angles $\Theta = (\theta_1 \ \theta_2 \ \theta_3 \ \dots \ \theta_H)^T$ is also a function of the vertices vector \mathbf{u} .

Finally, let F be the total number of faces. Note that in the extruded unit cells, all faces are squares. On each face, a plane can be determined by three vertices. In order to account for the possibility of face bending, we allow the fourth vertex to bend out of the plane, as shown in figure 22. (Note that this is true for small deformations in the face.)

Let φ_k be deformation in the k^{th} face. To calculate φ_k , let $\mathbf{a} = \mathbf{x}_B - \mathbf{x}_A$ and $\mathbf{b} = \mathbf{x}_C - \mathbf{x}_A$. Then, the normal vector of the plane determined by A, B, C is $\hat{\mathbf{n}} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|}$. We can then obtain the value

$$\varphi_k = (\mathbf{x}_{D'} - \mathbf{x}_D) \cdot \hat{\mathbf{n}}\tag{7}$$

We can define an $F \times 1$ vector $\boldsymbol{\varphi} = (\varphi_1 \ \varphi_2 \ \varphi_3 \ \dots \ \varphi_F)^T$, which contains the value of deformations in all the faces. As before, this is also a function of \mathbf{u} .

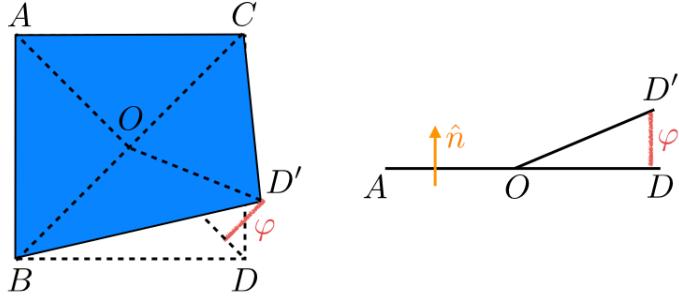


Figure 22: Small deformations associated with face bending can be described by φ .

A.2 Energy and its gradient

Since the total energy includes edge stretching, hinge bending, and face bending, we will break it down into three parts accordingly.

First, we define $d\mathbf{u}$ to be the change in vertices' coordinates between two configurations.

For all E edges,

$$\begin{aligned} E_{\text{edge}} &= \sum_{i=1}^E \frac{1}{2} k_{\text{edge}} (\Delta e_i)^2 \\ &= \frac{1}{2} (\mathbf{d}\mathbf{e})^T \cdot \mathbf{K}_{\text{edge}} \cdot \mathbf{d}\mathbf{e} \end{aligned}$$

where \mathbf{K}_{edge} is a diagonal matrix where all entries on the main diagonal is equal to the spring constant k_{edge} , and $\mathbf{d}\mathbf{e}$ is a vector describing the change in length of all edges, with

$$\mathbf{d}\mathbf{e} = \begin{pmatrix} de_1 \\ de_2 \\ de_3 \\ \dots \\ de_E \end{pmatrix} = \mathbf{J}_{\text{edge}} \cdot d\mathbf{u}$$

\mathbf{J}_{edge} is the Jacobian of edge lengths, which can be calculated element-by-element from equation 5. Hence,

$$E_{\text{edge}} = \frac{1}{2} (d\mathbf{u})^T \cdot \mathbf{J}_{\text{edge}}^T \cdot \mathbf{k}_{\text{edge}} \cdot \mathbf{J}_{\text{edge}} \cdot d\mathbf{u}$$

Similar operations can be applied to hinge deformations and face deformations as well, and we obtain

$$\begin{aligned} E_{\text{hinge}} &= \frac{1}{2} (d\theta)^T \cdot \mathbf{K}_{\text{hinge}} \cdot d\theta \\ &= \frac{1}{2} (d\mathbf{u})^T \cdot \mathbf{J}_{\text{hinge}}^T \cdot \mathbf{K}_{\text{hinge}} \cdot \mathbf{J}_{\text{hinge}} \cdot d\mathbf{u} \\ E_{\text{face}} &= \frac{1}{2} (d\varphi)^T \cdot \mathbf{K}_{\text{face}} \cdot d\varphi \\ &= \frac{1}{2} (d\mathbf{u})^T \cdot \mathbf{J}_{\text{face}}^T \cdot \mathbf{K}_{\text{face}} \cdot \mathbf{J}_{\text{face}} \cdot d\mathbf{u} \end{aligned}$$

The gradient of the energy ∇E can be derived in a similar fashion.

B Graphs: Directed vs. Undirected

The reason we are using directed instead of undirected graphs is that the latter cannot capture all information needed when determining the relative positions between two hinges in a polyhedron. Consider, for example, constructing an undirected graph from a cube (figure 23 (a)), as shown in figure 24 (a):

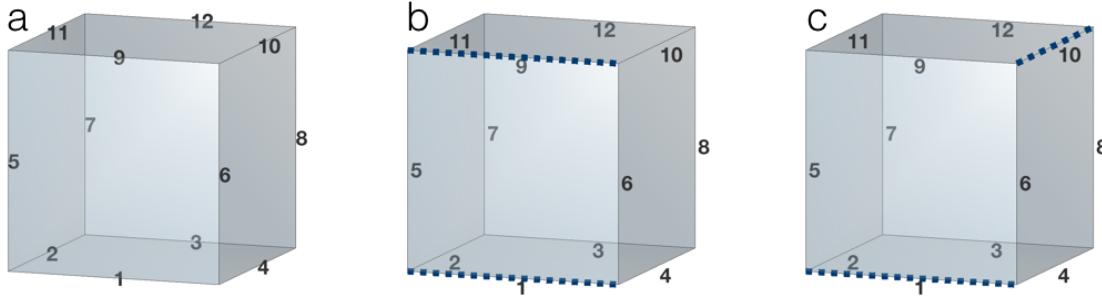


Figure 23: Cube with hinges numbered from 1 to 12.
 (a): No hinges selected; (b): Selecting hinges {1, 9}; (c): Selecting hinges {1, 10}.

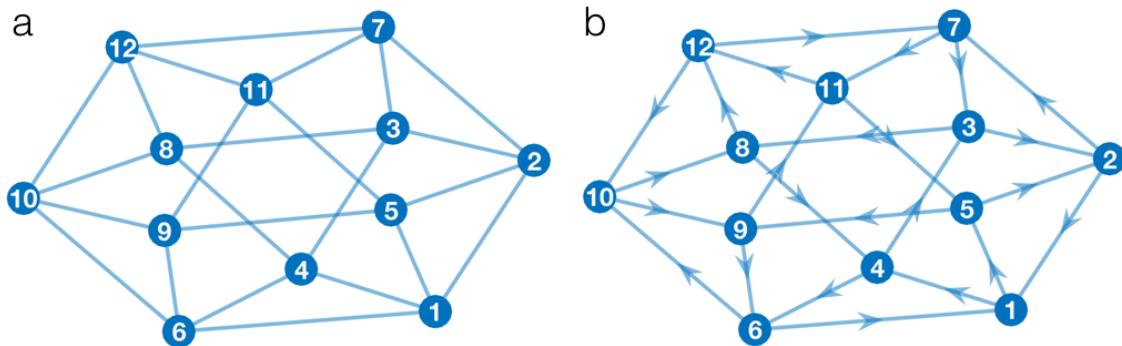


Figure 24: Graphs constructed from cube.
 (a): undirected; (b): directed

Here, selecting hinges {1, 9} and {1, 10} are equivalent situations in the graph: the shortest distance between the two nodes are the same in both cases. However, in the real geometry, these two selections are different, as shown in figure 23 (b) and (c).

If we use directed graph (see figure 24 (b)), on the other hand, cases like this could be easily differentiated. In this example, to reach node 9 from node 1 we need to cross at least one other node; to reach node 10 from node 1 we need to cross at least two other nodes.

C Fabrication of prototypes

The cardboard prototypes (shown in figure 25) were fabricated from two layers of cardboard (thickness 0.4 mm) connected by one layer of double-sided tape (thickness 0.1mm). For details of fabrication, see supplementary materials of [12].

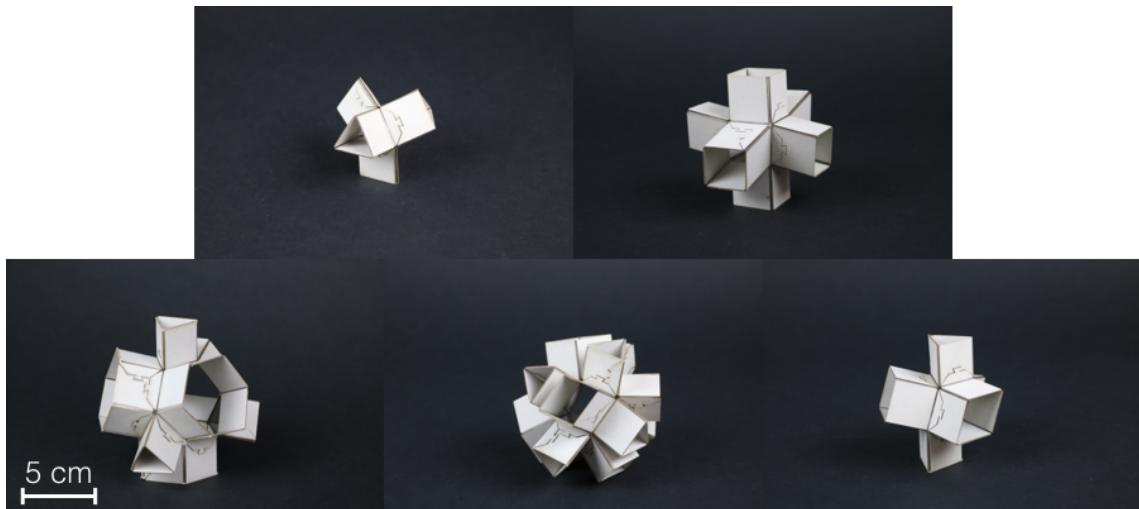


Figure 25: Cardboard prototypes of unit cells.

Top row: (left to right): tetrahedron, cube.
 Bottom row: truncated tetrahedron, cuboctahedron, triangular prism.

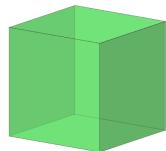
D List of all (semi)-regular convex polyhedra

Regular convex polyhedra, or Platonic solids, consist of congruent regular polygons with the same number of faces meeting at each vertex. There are only five polyhedra in the category:

tetrahedron



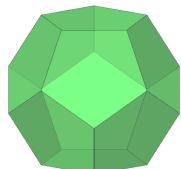
cube



octahedron



dodecahedron



icosahedron

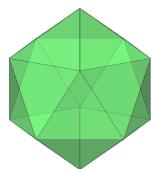


Table 4: Five Platonic solids

Semi-regular convex polyhedra consist of more than one types of regular polygons with equal edge lengths. They include 13 Archimedean solids, and an infinite series of convex prisms and convex antiprisms:

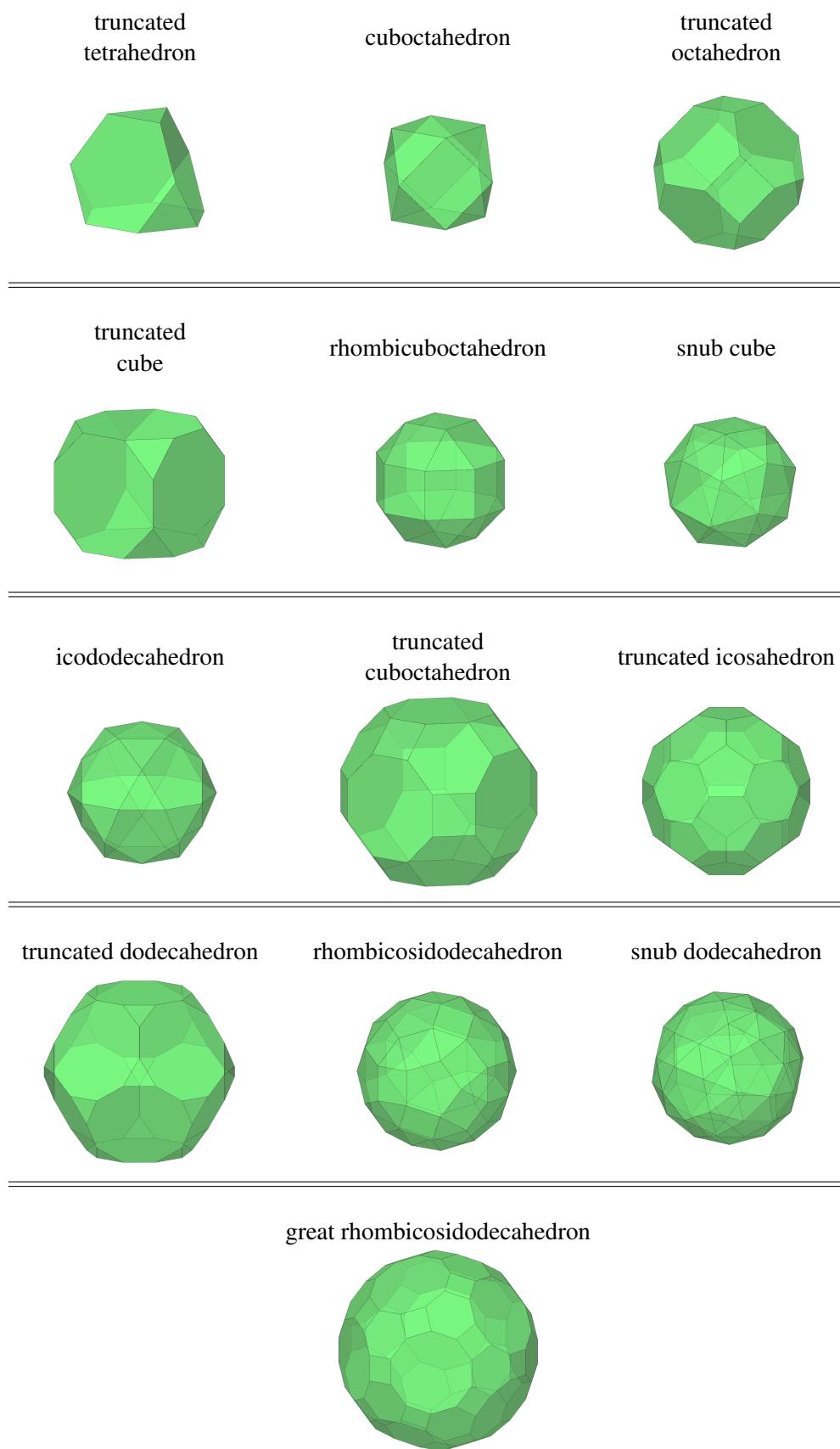


Table 5: Thirteen Archimedean solids

triangular prism ... hexagonal prism ...



triangular antiprism
(octahedron) ... hexagonal antiprism ...



Table 6: Prisms and antiprisms

E Space-filling tessellations of (semi-)regular convex polyhedra

There are 28 groups of convex (semi-regular) polyhedra that can form space-filling tessellations together, as listed below:

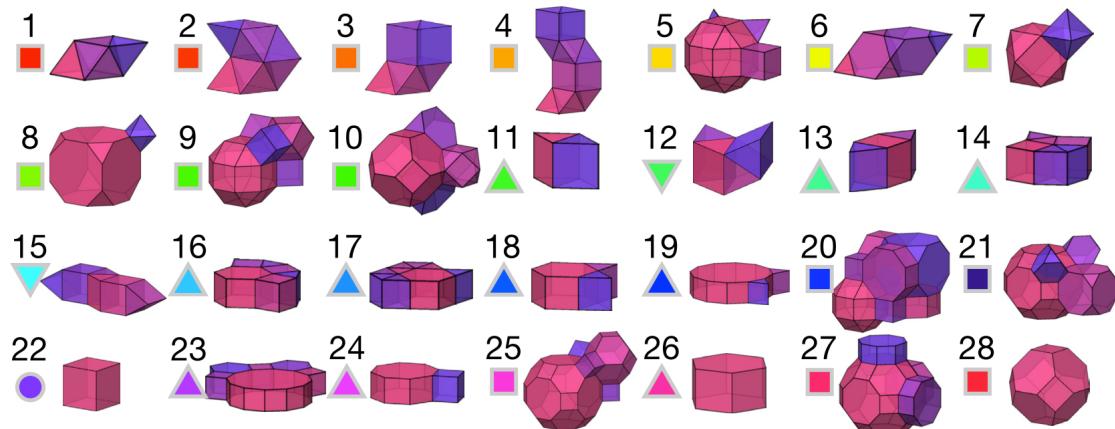


Figure 26: List of uniform 3D tessellations [12].