A unistable polyhedron with 14 faces

Article *in* International Journal of Computational Geometry & Applications · March 2014

DOI: 10.1142/S0218195914500022

1 author:

Alexander Reshetov
NVIDIA
19 PUBLICATIONS **410** CITATIONS

SEE PROFILE

International Journal of Computational Geometry & Applications
© World Scientific Publishing Company

A UNISTABLE POLYHEDRON WITH 14 FACES

ALEXANDER RESHETOV*
NVidia Research, 2701 San Tomas Expy, Santa Clara, CA 95050, USA
areshetov@nvidia.com

Received Sep 27, 2013
Revised May 17, 2014
Communicated by (Name)

ABSTRACT

Unistable polyhedra are in equilibrium on only one of their faces. The smallest known homogeneous unistable polyhedron to date has 18 faces. Using a new optimization algorithm, we have found a unistable polyhedron with only 14 faces, which we believe to be a lower bound. Despite the simplicity of the formulation, computers were never successfully used for solving this problem due to the seemingly insurmountable dimensionality of the underlying mathematical apparatus. We introduce new optimization approaches designed to overcome the problem's intractability and discuss its significance to other application areas. We also mathematically prove the unistable property of the found bodies using rational arithmetic. Surprisingly enough, all of our computer-generated unistable polyhedra look similar to the human eye, providing important insights into the nature of the problem.

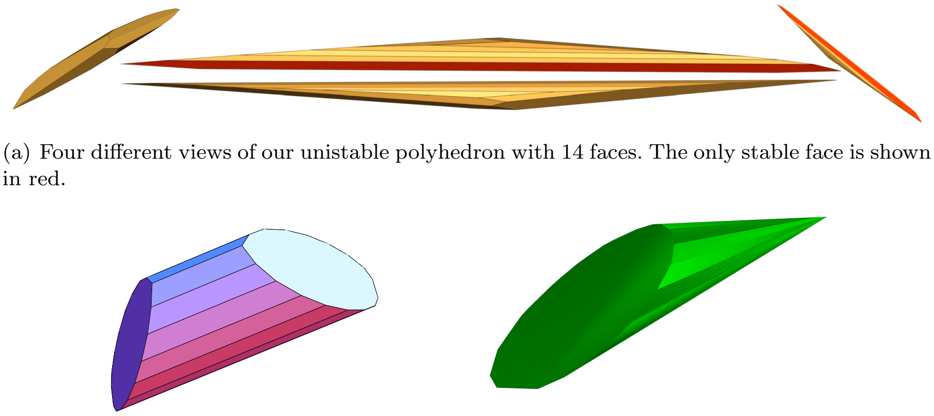
Keywords: miscellaneous polyhedra, stability, center of mass, bound constrained Lipschitz optimization, unsolved problems

1. Introduction

We consider all homogeneous convex polyhedra (3D polytopes). If such a body is in equilibrium on only one of its faces, it is said to be *unistable*[1] (with respect to the uniform gravity field orthogonal to the rest plane). In discrete geometry it is also common to refer to a unistable polyhedron as *monostatic polyhedron.* To date, all known unistable polyhedra are fruits of human ingenuity. Conway and Guy[2] constructed a unistable convex homogeneous polyhedron with 19 faces. It was long believed to be the best possible result, until improved by Bezdek[3], who discovered an 18-faceted polyhedron. Both these polyhedra are shown in Figure 1(b).

Every homogeneous tetrahedron is stable on at least two faces[4]; therefore the unexplored interval is from 5 to 17 faces. One possibility to harness computational power for probing this interval is to

*part of this work was done in Intel Labs



(a) Four different views of our unistable polyhedron with 14 faces. The only stable face is shown in red.

(b) Prior art unistable polyhedra. **Left**: Guy's polyhedron (19 faces). **Right**: Bezdek's polyhedron (18 faces).

Fig. 1. Unistable polyhedra.

a. describe a mechanism for creating an arbitrary convex polyhedron from a set of given variables,

b. provide a cost function which will have a minimum when the constructed polyhedron becomes unistable, and (1)

c. devise an efficient optimization routine.

Conceptually, all these steps can be carried out using exact computations, either with integer or rational numbers. However, computations with fixed precision float numbers are typically more efficient. If inexact calculations are used to find a candidate unistable body, it has to be converted afterwards to an exact form and its properties have to be provably reasserted to get a definite mathematical proof.

The success of this approach depends on the complexity of the used cost function. A posteriori, we plotted the typical behavior of the cost function in the local neighborhood of our 14-faced unistable polyhedron (Figure 2, right). The probability of randomly finding such body is about one in $(1/10^{-4})^{3\times 14}/14! \approx 10^{157}$, accentuating the need for an efficient optimization.

As shown by Várkonyi and Domokos[5], the number of stable equilibria can easily be increased by carving away small portions of the polyhedron. The reverse is false: i.e., for a typical body it is very difficult to decrease the number of equilibria via small perturbations. This pretty much excludes local optimization techniques, making the problem even more convoluted.

Minich[6] attempts to reduce the problem space by considering only polyhedra that are similar to Guy's object. This still results in a high-dimensional problem, so clever approaches were used to further exclude large portions of the search space. These efforts were not successful.

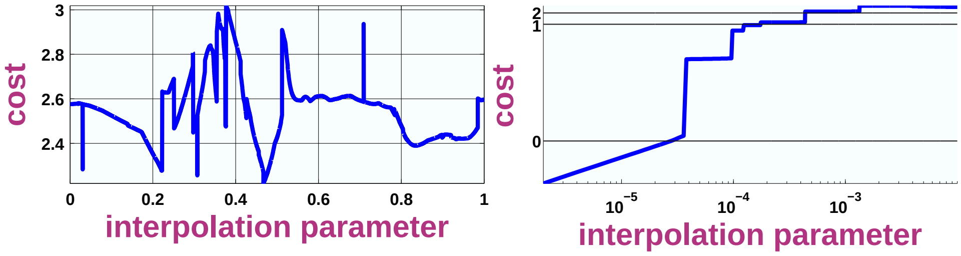


Fig. 2. Typical behavior of the cost function. Values less than 0 correspond to unistable polyhedron. **Left**: interpolating between 2 random parameter sets in 60-dimensional space. **Right**: local neighborhood (cut along a straight line in a parametric space) of the found unistable polyhedron with 14 faces in log-log scale. For each chart, 5000 sample points were used.

We believe the only way to obtain leverage against the problem is to design our own cost function. Without loss of generality, such a function maps a point in n-dimensional parametric space, which uniquely specifies a particular polyhedron, into a single value that characterizes how close we are to the solution (unistable body). The most important property of the cost function is a correlation between its values at different points. If the correlation is significant, we could "predict" the behavior of the cost function by analyzing sparsely located points (with respect to Euclidean distance in n-dimensional parametric space). We will provide the formal definition of this property in section 2.2.

It is worth mentioning that there are homogeneous 3D bodies which have exactly one stable and one unstable *point* of equilibrium. Such bodies were first described by Várkonyi and Domokos[5], confirming the hypothesis proposed by Vladimir Arnold in 1995. These mathematicians deserve credit for making stability problems popular again.

Unistable bodies exist in all dimensions greater than 2. And, in fact, there are unistable simplices in 10-dimensional space, as shown by Dawson[7].

2. Mathematical Apparatus

In this section, we will describe our implementation of steps (1), i.e., steps required to create a plurality of polyhedra and find ones with the desired properties.

2.1. *Polyhedron Generation*

As discussed by Henk and Ziegler[8] and Schneider[9], three basic ways of generating a random convex polytope are

(1) construct a convex hull of finitely many random points,
(2) compute an intersection of closed half-spaces,
(3) project a high-dimensional polytope into a subspace.

The third method is somewhat esoteric. Using convex hulls is an attractive proposition since integral properties of any polyhedron (volume, centroid, etc) can be algebraically computed from coordinates of its vertices as described by Mirtich[10] and Wang[11]. However, we rejected this approach since it almost surely creates only triangular faces. Note that all prior art unistable polyhedra (see Figure 1(b)) contain faces with high valence. Another problem with this approach is that it potentially results in many dormant points, that lie inside the convex hull of other points. Accordingly, the final shape will not depend on these variables, confounding the optimization process.

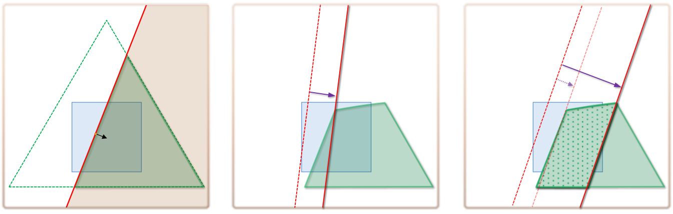


Fig. 3. Building polyhedron through intersection of half-spaces. **Left**: half-space is defined by a point in the unit cube. **Middle**: re-adjusting non-intersected planes. **Right**: avoiding very small faces (the resulting body is shaded).

Correspondingly, the second approach (half-spaces) creates *vertices* with valence 3 and could also have dormant planes that do not contribute to the final body. Moreover, unlike the convex hull approach, the intersection of arbitrary half-spaces can be open. We fight all these problems with a custom process illustrated in Figure 3 and outlined below.

For each three numbers $\mathbf{n} = [n_x, n_y, n_z]$ in the $[-0.5, 0.5]^3$ cube, we define a halfspace $H(\mathbf{n})$ such as

- its plane is orthogonal to vector $\mathbf{n}$
- the halfspace plane contains point $[n_x, n_y, n_z]$
- origin $[0, 0, 0]$ is inside $H(\mathbf{n})$

We split a set of given variables into triplets $\mathbf{n_j}$ and construct a corresponding polyhedron as follows:

(1) We initialize the constructed body $\Omega = H(\mathbf{t_1}) \cap H(\mathbf{t_2}) \cap H(\mathbf{t_3}) \cap H(\mathbf{t_4})$ where $\mathbf{t_i}$ are normals defining a regular tetrahedron. Accordingly, we will always have a valid closed polyhedron at all steps of the construction. The intent is to have the tetrahedron big enough so its faces will not contribute to the final result, but it is not required.
(2) We consider all given triplets $\mathbf{n_j}$ one by one. If the intersection $\Omega \cap H(\mathbf{n_j}) \neq \emptyset$, we use it at the next iteration (left of Figure 3).
(3) If the plane does not intersect the body, we move it so it will go through the second closest vertex (middle).

The rationale for step 3 is to allow most of the half-spaces to contribute to the final body. Simply ignoring non-intersecting plane(s) might also work, but we never tested it.

Vertex coordinates are not used explicitly in the polyhedron definition but rather derived from planes (a vertex is considered as an intersection of 3 planes).

This algorithm will work if $\mathbf{n_j}$ are represented by rational numbers and will create an exact convex polyhedron. If $\mathbf{n_j}$ are float numbers, we will have to deal with numerical inaccuracies, given that these steps can create very small faces (or even the plane coinciding with an existing face). If edges of the newly created faces are shorter than the given threshold, we will consider the next closest vertex and so on (Figure 3, right). In extreme cases, nothing will work and we will just reject the current plane, leaving the three corresponding variables dormant.

The number of the used half-spaces roughly corresponds to the number of faces in the built polyhedron. It can be increased by the number of the contributing faces from the original tetrahedron and decreased by the number of the rejected planes. It is also quite possible that the currently considered half-space completely cuts off already built faces, leaving the corresponding variables dormant. This is undesirable by itself, but it also helps us to create polyhedra with a smaller number of faces than the number of the original half-spaces (remember that our goal is to reduce the lower bound for the number of faces in a unistable polyhedron).

We found that starting with 20 planes is adequate for finding record-breaking polyhedra (note that this decision stipulates a 60-dimensional search space). After each optimization phase, we eliminate all dormant variables. If the number of half-spaces falls below the pre-defined threshold (13), we add an additional half-space, assuming that unistable polyhedra with a very small number of faces do not exist. Note that the number of faces in the created body could be less than the number of the used half-spaces (some half-spaces might not contribute to the final body), so such optimization does not preclude, in theory, search for a unistable polyhedra with smaller than 13 faces.

For expediency, we use float numbers during the initial search for candidate bodies. We have discovered that singe precision float numbers are not adequate, due to the extremely small steps required when the cost function approaches 0. We use double precision float numbers instead.

Once candidate unistable polyhedron is found, we convert float numbers, which define the body, to their rational approximation (at maximum accurancy), rebuild the polyhedron, and verify the unistable property.

Such rational approximation could conceivably create a body that is topologically different from one defined by float numbers, especially if valency of some vertices in the original body was more than 3. Accordingly, the verification process using rational approximation is an integral part of the search for unistable polyhedra. It so happens that in all found unistable bodies, valency of all vertices was always 3, despite the fact that step 3 above might encourage creation of vertices with higher valency. We speculate that such bodies were purged during the optimization process, as higher valency vertices tend to belong to bodies with higher number of faces.

2.2. *Cost Function*

It is rather straightforward to define a plain vanilla cost function that will be 0 at unistable polyhedra and positive for other bodies. It is more challenging to find one that will guide us through the optimization process, with smaller values corresponding to bodies that are somehow closer to the goal.

The number of stable faces by itself cannot be indicative of any progress toward a unistable polyhedron, as it is very easy to create a polyhedron with just two stable faces (just consider a variety of assorted cut pyramids).

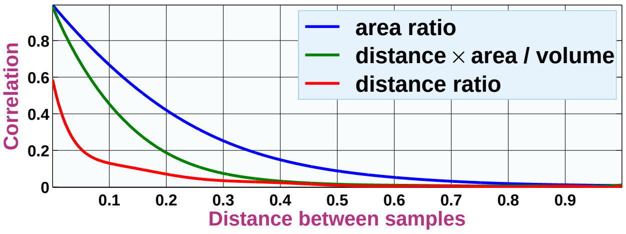
Since a polyhedron is defined by a sample point in 60-dimensional parametric space (as described in the previous section), it is expedient to identify *closeness* with Euclidean distance in this space (with the distance between maximally distant samples in the unit hypercube equal to $\sqrt{60}$). Note that the samples that are close geometrically could still produce very different polyhedra due to the threshold-based nature of step 3 from the previous section.

In most cases, candidate cost functions quickly cause creation of degenerate bodies (very flat or very long). In order to avoid this, a cost function should yield a unitless numerical value, so it can use lengths, areas, and volumes only as a ratio. This is a necessary but not sufficient condition.

It would be nice to be able to tell if one cost function is better than another without running multi-day simulations. We cannot afford to densely sample all neighborhoods of the parametric space that defines polyhedra. Thus we will place more value on those cost functions that have a better correlation between sparsely placed samples.

To measure the expected correlation as a function of distance between two samples, we evaluate cost function for multiple random pairs of sample points in the parametric space at a specific distance from each other and then compute the mathematical correlation between the resulting series. Note that this will give us a higher-order function (functor). For any given cost function it will create *a function* that maps distance to the expected correlation. We will then inspect the shape of this function and favor functions that have longer tails (see Figure 4 for a few examples).

Fig. 4. Average correlation between two samples at the specified distance for different cost functions; the maximum possible distance between samples is $\sqrt{60}$.



To formally introduce possible choices for cost functions, we partition the faces of a given polyhedron into groups containing exactly one stable face. A face's stability can be verified by projecting the body's centroid (the center of mass) onto the face plane: the face is stable only if the projection lies inside the face (for convex bodies). We partition the faces by

- initially including all stable faces into their own groups and then
- iterating over remaining unstable faces and assigning them to the group for the stable face onto which the polyhedron naturally rolls when set on a given face on a flat surface (ignoring possible momentum overruns and considering only geometric properties).

By design, this process creates groups that contain one and only one stable face each.

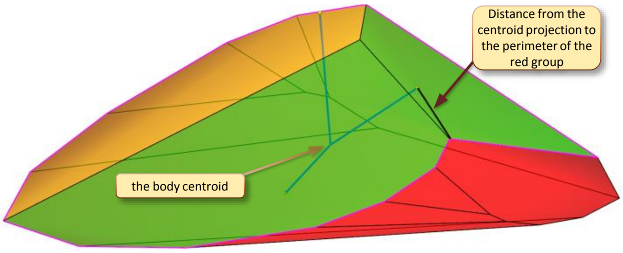

Distance from the centroid projection to the perimeter of the red group
the body centroid

Fig. 5. Cost function computation (using 13-faced polyhedron as an example). Three different facet groups are shown in brown, green, and red color. The red group is a good one (has the largest total area).

During the search for unistable bodies, we can say that progress is made when the largest group (by total area) grows, and the rest shrink to the point of vanishing. For lack of more discernible terminology, we will call the largest group (and its faces) *good* and all others *bad*. In Figure 5, there are three facet groups, i.e., there are three stable faces. We want to make brown and green groups smaller and the red one bigger.

One possible choice for the cost function is the ratio of the total area of the bad faces to the good ones. Unfortunately, it does not tell us very reliably how close we are to the goal. If we perturb the faces of the prior art unistable polyhedra (Figure 1(b)) just a little bit, we may get a multistable polyhedron in which bad faces will be large (with respect to the perturbation). What is happening actually is that previously good faces are suddenly becoming bad ones.

This can be dealt with by projecting the polyhedron's centroid onto stable faces in bad group(s), and considering the distance between this point and the perimeter separating bad and good faces. In Figure 5, the perimeter is shown as magenta line and the distances — as black lines (only one is noticeable since the other one — for the brown group — is very small). We compute this distance in Euclidean space, i.e. if the bad stable face is not adjacent to a good face, we compute the length of the vector that goes through the solid. This approach is chosen just to simplify computations, we did not test any other options. To get a unitless value, the minimum distance has to be divided by another length, for example, the minimum distance from the centroid projection onto the good stable face to its edges.

If this distance-based cost function is used as-is, it tends to create very flat polyhedra with just a few faces. Additionally, cost values for closely placed samples are poorly correlated with each other, making it difficult to "predict" the cost values.

Figure 4 shows the correlation functions for different cost functions. The distance-based cost function (red color) has a modest correlation and is a poor choice. While the correlation for the area-based cost function (blue) is high, it cannot guide us to the goal as this cost function is always discontinuous at unistable bodies. The green line shows the correlation for the cost function we have eventually used, combining the good properties of the distance-based and the area-based cost functions, and avoiding their shortcomings.

It is computed as follows. For all bad faces, we find the minimum distance to the perimeter of the good group from the centroid projections and multiply it by the face area. The sum of all such values is then divided by the polyhedron volume. This function has a few properties that make it very attractive. It does not differentiate between bad faces: we do not care if a bad face is stable or not, we just want it to go away. Similarly, good faces contribute to the cost function only indirectly (through volume), as we do not want to numerically quantify the goodness property until we actually find a unistable body (then we will want bigger safety margins — see section 2.4). Note also that all face planes are meaningfully contributing to the cost by influencing the body volume, which is important for numerical optimization.

Such cost function has 0 value for all unistable bodies and positive for other ones. It might be convenient to further differentiate between unistable bodies pursuing certain desirable properties. For example, we could search for bodies that will be easier to manufacture. This can be facilitated by assigning negative values to cost function for unistable bodies and allowing the optimization process find the minimum corresponding to the most desired properties. This approach is further explained in section 2.4.

2.3. *Cost Function Minimization*

We need an optimization process that can take advantage of the correlation between sparsely placed samples and guide us to 0 (which corresponds to a unistable polyhedron), since we cannot afford exploring all neighborhoods of the unit cube $[-0.5, 0.5]^{3\times 20}$ in 60-dimensional space. Figure 2 (left) shows typical behavior of our cost function, computing it along a random line in this space. Obviously, derivative-based optimization techniques will not be able to find the global minimum of such a function.

We initially assumed that quadratic (or higher order) approximation to the objective function would allow us to codify the correlation into a simpler model, ensuring a fast rate of convergence, similar to the NEWUOA method[12]. However, the extrema of the cost function are localized in very narrow regions, and the NEWUOA procedure treated them as outliers. Similarly, neither the simplex method[13] nor simulated annealing[14] was able to penetrate these regions. Furthermore, the standard Nelder-Mead simplex algorithm becomes inefficient in high dimensions[15] and simulated annealing does not exhibit a high rate of convergence even in the best scenarios.

Our chosen technique uses Lipschitzian optimization in the DIRECT method, which stands for "DIviding RECTangles"[16]. Ironically, our cost function does not have Lipschitz continuity. What makes the DIRECT algorithm suitable is that it does not use the explicit Lipschitz constant, instead it is adaptively dividing the initial hypercube and continuing with the potentially optimal rectangles. Conceptually, this algorithm carries out simultaneous searches using all possible constants, and therefore operates at both the global and local level. The fact that such a constant does not exist at all becomes immaterial, since our function exhibits strong correlation between sparsely placed points (rectangle vertices) and thus can guide the optimization process.

Still, the DIRECT method was not quite able to go directly to the solution, instead getting stuck at local minima. To unstick it, we alternate between different search spaces, some of them having more and others fewer independent variables than the original 60-dimensional space describing polyhedron planes. For each alternative space, the starting point was one that would recreate the current best polyhedron verbatim, while still allowing us to explore alternative representations. These spaces are defined as follows:

(1) We could actually treat *vertices* of the given polyhedron as variables, increasing the number of the independent variables (in comparison with face planes). In general, face vertices become un-planar when changed by the optimization routine. For each function evaluation, we refit all vertices of a particular face into a plane and use these modified planes as half-spaces to create a new convex polyhedron and evaluate the cost function of the modified polyhedron. Not only does this search space have more degrees of freedom, it can also dramatically change the structure of the polyhedron.

(2) A linear transformation (scaling + shear + rotation), defined by $3 + 3 + 3 = 9$ parameters. This transformation converts planes into planes, thus not changing the structure of the optimized polyhedron, but allowing us to non-uniformly stretch and skew it.

(3) A simplified form of the linear transformation, allowing only scaling along three principle component axes of the perimeter vertices. The rationale for this is that these axes represent a natural framework for the analyzed body as far as we measure everything with respect to the perimeter between good and bad faces.

(4) If the number of faces in the current polyhedron is small, we could also consider adding an extra face, optimizing its placement through the search in the corresponding $[-0.5, 0.5]^3$ space.

If optimization in one of these spaces was successful, we reverted to the face-based optimization using the best found polyhedron as a new starting point for the DIRECT method (i.e., the center of the new hypercube). If not — we tried to increase the number of the function evaluations. If this did not help, we then aborted the current pursuit, continuing with a random initial sample.

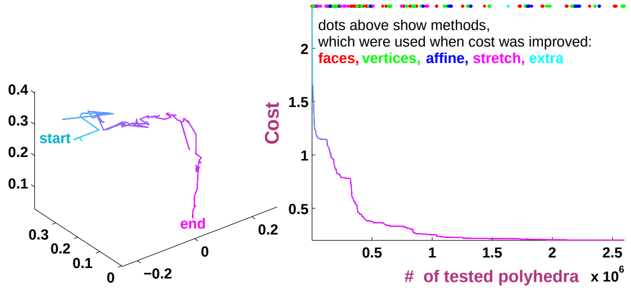


Fig. 6. **Left**: typical evolution of one of the planes defining the polyhedron; each 3D point represents three plane parameters at a given iteration (out of 60 overall parameters). **Right**: reduction in the cost value with respect to the number of tested polyhedra.

Figure 6 shows a typical evolution of one of the planes, defining a particular polyhedron: on the left as a 3D point, and on the right as the corresponding change in the cost function. Top of the chart on the right also shows optimization techniques (as described in this section), at which improvements in the cost function were achieved.

We do not plot the complete simulation, as it typically requires about 10 billion function evaluations and the cost function chart would be L-shaped (quick initial progress and then very slow convergence to 0).

We did not try genetic algorithms in our optimization, but observed that our approach to the simulation — when small changes in parameters could have dramatic consequences — exhibits many features of a genetic optimization[17].

2.4. *Post-processing*

Once a unistable polyhedron is found, we could try improving it. The obvious choice is to increase the safety margin, defined as the minimum distance between the centroid projections onto all faces and the corresponding face edges, divided by $volume^{1/3}$. This could be achieved by continuing the DIRECT optimization. Previously, we have defined the cost function to be 0 for unistable bodies (and positive for other ones). It could be redefined for unistable bodies to be equal to the negative value of the safety margin. Accordingly, the optimization process will try to minimize this value, i.e. maximize the safety margin, once a unistable body is found.

Figure 7 shows such projections for a 15-faced polyhedron *after* the optimization process. We did not show a 14-faced polyhedron, as all projections for such a body appear to land on edges due to the very small margins. Table 1 provides the best found margins for different polyhedra. When optimization process first produces

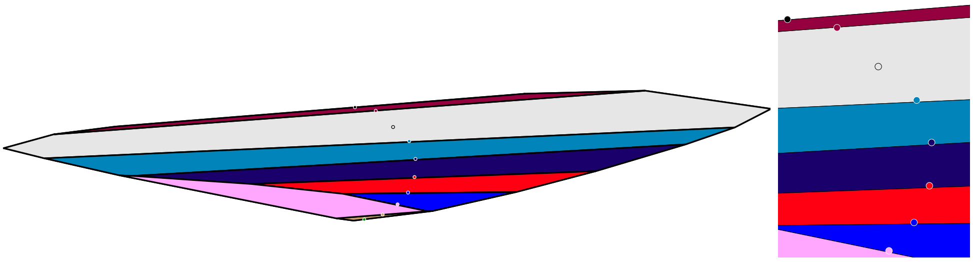


Fig. 7. Centroid projections on face planes of 15-faced polyhedron. The only stable face (i.e. one that contains the centroid projection) is a gray one on the top. Projections have the same color as the corresponding face (zoomed up section on the right).

a unistable body, safety margins are typically less than 10^{-6}. The safety margin improvements are needed to allow 3D printing of real unistable bodies (section 4).

faces	14	15	16	17
safety	0.0012	0.0042	0.0067	0.0094

Table 1. Best-found safety margins (relative distances to edges from centroid projections) for unistable polyhedra with different number of faces.

Another possibility for post-processing is to make one of the faces as small as possible without destroying unistability. We then can exclude this face and re-optimize the new body in the hope of reducing the lower bound. It turned out that this is the most efficient way of achieving this goal. We obtained a 15-faced body through direct search only a few times. The face reduction allowed us to create many more such bodies, in one case sequentially reducing the number of faces from 19 to 15.

2.5. *Unistable Dekatesserahedron*

The face reduction process (see the previous section) was not successful for 15 → 14 improvement without revisiting the basic simulation principles.

First, the distance×area cost function (section 2.2), severely penalizes bodies with two or more bad groups, quickly causing the creation of a polyhedron with one good and one bad group. This is great for expediently purging unpromising configurations, but once we are very close to the minimum, it locks the independent variables in a very narrow range, not allowing bigger (but potentially more advantageous) changes. In effect, by analyzing Figure 7, the centroid projections become almost planar and lie on a circle. Accordingly, any variable changes that violate this will be rejected (because they will drastically increase the cost). There is still room to maneuver for 15-to-17 faceted bodies, but those with 14 faces are significantly more constrained (see Figure 2).

Thus, while optimizing 14-faced bodies we switched to the distance-based cost, using the square of the distances to the perimeter (Figure 5). Beginning the search process with this cost function yields flattened bodies, but it is useful late in the search process, when we are near a solution. It allows the search variables to vary in a greater interval without the significant increase in the cost.

Another way to fight the apparent dimensionality reduction near the minimum is to learn it. We did this by finding a new minimum along the line connecting the last two found minimums, let's say p_0 and p_1, $f(p_1) < f(p_0)$. We employed a rather trivial strategy, evaluating the cost at $p_2 = p_1 + 0.1\ (p_1 - p_0)\ t$ for $t = 1 \ldots n$, continuing the process while the function value is still improving. In effect, we tried to explore the steep canyon punctuated by the last two minima.

The difficulty is in integrating this with the DIRECT method[16], in which the function is evaluated at multiple regular hyper-rectangle vertices, reusing the older values while sequentially splitting the rectangles. We just used the found value $f(p_2)$ when DIRECT was expecting $f(p_1)$. Arguably, this is not a very smart approach, as it replaces a regular target function with a non-deterministic procedure. The only consolation is that it actually worked, due to the overall robustness of the DIRECT method.

3. Mathematical Proof

In half-space representation of polyhedra, vertices are not included in the definition by itself, and only computed as an intersection of the corresponding planes. Accordingly, we could always consider a given representation as one with infinite precision, which just happens to be described by decimal numbers.

Each such float number can be converted to a ratio of two integers, and we can carry out all computations using rational arithmetic. This is essential in proving the unistable property, as safety margins of the found bodies can be very small, sometimes below float epsilon.

The only catch is that each operation with rational numbers essentially doubles the number of digits in the enumerator/denominator. We wrote a *Mathematica* program, which computes the centroid projections onto faces of a given body and verifies that only one such projection is inside the face. We had expected that it would crash (most likely scenario), or would be very slow. Instead, it finished in 0.6 seconds, conclusively proving the unistable property. The computed centroid representation had about 25 thousand digits.

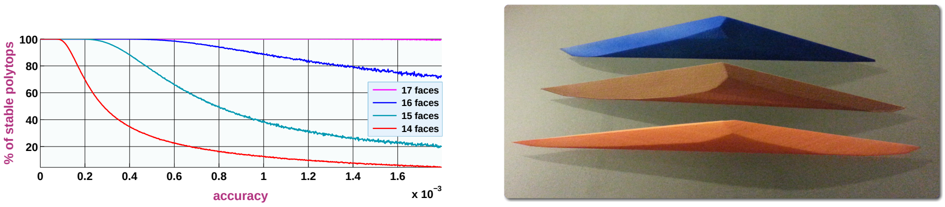
Since the unistable property is invariant with respect to rotation, translation, and uniform scaling, it is possible to find the transformation after which all plane variables become integers and the overall representation will have a minimum length. One such representation for a 14-faced unistable polyhedron is given in Appendix A as a *Mathematica* notebook. To verify unistability, it computes signed distances to the nearest edge from projections of the body's centroid onto all faces. It is interesting to note that the shortest representation is actually achieved for a 15-faced body and contains just 64 digits.

Through the ages, it was assumed that any mathematical proof could, in principle, be checked by a competent mathematician to confirm its validity[18]. In our case, the math is elementary and we could easily generate a manuscript containing all calculations in rational arithmetic. It is significantly more difficult though to find a mathematician who would really go over it. So, our belief in the existence of 14-faced unistable polyhedra is solely based on the correctness of the computer program *Mathematica*. It is not that big a stretch though, given that thousands of people are flying on airplanes every day.

This belief is also corroborated by creating a real world unistable polyhedra through 3D printing.

4. 3D Printing

3D printers have a limited precision, 0.1 *mm* at most. For about 10 *cm* dimensions, it corresponds to a relative accuracy of 10^{-4}. Given low safety margins for the found polyhedra (Table 1), an interesting question is whether it is actually possible to print and verify such models.



(a) The probability to get a unistable polyhedron for different 3D printing accuracy settings.

(b) Three 3D-printed models of unistable polyhedra on a glass surface with 15 (orange), 16 (brown), and 17 (blue) faces.

Fig. 8. Making unistable polyhedra.

In Figure 8(a), we tried to predict the outcome of such an experiment by perturbing planes of the found polyhedra and checking the unistable property for different precision settings. It can be seen that a 17-faced polyhedron can be reliably printed with the current technology.

In addition to limited precision, very thin details could break during printing or handling of a model. For resin material, which generally has higher precision than plastic, this thickness restriction is about 2 *mm*. All our models have rather acute areas near polar regions. An obvious way to fight this is to make a model bigger. Another possibility is to penalize elongated bodies during the post-processing optimization (section 2.4), while simultaneously preserving safety margins at an acceptable level.

We have printed 15, 16, and 17-faced bodies using unpolished plastic material (Figure 8(b)). All these bodies are unistable.

5. Discussion

When we started this work, we were hoping that we would be able to re-create models similar to ones shown in Figure 1(b) and (if we were very lucky) reduce the number of faces. Instead, quite unexpectedly, we discovered a new type of unistable polyhedra. It brings into focus a series of rather interesting problems.

Why were prior art bodies never discovered during the simulation? Our contention is that such bodies are too artificial and not very well correlated with their immediate neighborhood in the parametric space. On the contrary, the general outline of bodies, which we discovered, emerges rather quickly during our simulations, followed by steady improvement over multiple iterations. This is how natural selection works, and our approach resembles this, including sharp cost improvements due to beneficial mutations. We refer readers to Várkonyi and Domokos [19], in which the importance of stability for natural and human-made creations is discussed in greater detail.

What is the real complexity of the problem? We do not know. We were able to find a solution after about 10^{10} function evaluations (corresponding to 2-3 days work on an eight core machine), which is significantly less than $2^{3 \times 20} \approx 10^{18}$, the number of sample points when each dimension is sampled only twice. This seems to indicate that unistable bodies are distributed more or less densely in the parametric space. It is shown by Das and Goodrich [20] that many optimization problems involving convex polyhedra are NP-hard or NP-complete. Apparently, searching for unistable bodies is not one of them.

Does a 13- (or fewer) faceted unistable polyhedron exist? Bender [21] gives an asymptotic expression for the number of combinatorially distinct convex polyhedra as a function of the number of faces, see Table 2. Our polyhedra are *not* combinatorially equivalent even within a class with the same number of faces, but all our unistable polyhedra look similar when watched with a human eye. This may be a consequence of the chosen cost function, or it may represent a class of objects wherein unistability is most robust.

13	14	15	16	17	18
2.3304×10^7	3.5830×10^8	6.9132×10^9	1.0705×10^{11}	2.1122×10^{12}	3.9914×10^{13}

Table 2. Estimation of number of combinatorically distinct polyhedra with a given number of faces.

We surmise that 14 faces is a limit for this class, but there could be other possibilities. Since the number of distinct polyhedra with the number of faces less or equal to 13 is so small, it might even be possible to employ a different search strategy by describing a plurality of bodies in each group, using a significantly smaller number of parameters.

A considerably more difficult problem is to disprove the existence of a unistable property for smaller number of faces. Inspired by Figure 7, we speculate that from any monostatic polyhedron, another monostatic polyhedron can be derived, in which the projections of the centroid on the faces' planes lie in a plane on a circle. If this is correct, we might try disproving existence of such simplified monostatic polyhedra and then transfer it to a general case.

Another interesting problem is to find unistable polyhedra in which the stable face is the smallest. Conway pointed out that in 3D space by increasing the number of faces one can modify any unistable polyhedron so that it is stable on its smallest area face[22]. The challenge is to find such body with minimum number of faces. We surmise that the approaches described in this paper are directly suitable for tackling this problem.

6. Conclusions

We researched equilibrium properties of a certain class of polyhedra. Stability is the most ubiquitous property of bodies in the real world, while polygonal mesh models are the preferred geometric representation in computer graphics. The discovery of unistable polyhedra with a minimum number of faces was made possible through the interplay of the following ideas:

- We contend that in complex optimization problems correlation between function values at sparsely located samples is one of the most important properties, allowing the construct of an agile optimization process.
- Alternative descriptions of the problem can be used to intelligently move the solution out of local minima.
- Non-continuity of the goal function can be a blessing in disguise, allowing sharp cost improvements, similar to genetic optimization.
- The DIRECT optimization technique survives non-deterministic goal functions, allowing the use of local optimization methods underneath it.

We hope that some of these ideas could be beneficial in other application areas as well.

Acknowledgements

We would like to thank the reviewers for their detailed and helpful comments and suggestions. We gratefully acknowledge invaluable discussions with and assistance from Chris Burns.

References

1. I. Pak, Lectures on discrete and polyhedral geometry (2009).
URL `http://www.math.ucla.edu/~pak/`

2. R. Guy, Solution to problem 66-12, stability of polyhedra, SIAM Review 11 (1) (1969) 78–82.
3. A. Bezdek, On stability of polyhedra, in: Workshop on Discrete Geometry, Sep 13-16, 2011, Fields Institute, Canada, 2011, pp. 2490–2491.
4. R. J. M. Dawson, Monostatic simplexes, American Mathematical Monthly 92 (8) (1985) 541–546.
5. P. L. Várkonyi, G. Domokos, Mono-monostatic Bodies: The Answer to Arnold's Question, The Mathematical Intelligencer 28 (4).
6. C. Minich, Search for small monostatic polyhedron, in: WSCG Communication Papers Proceedings 2012, 2012, pp. 309–316.
7. R. Dawson, W. Finbow, Monostatic simplexes III, Geometriae Dedicata 84 (1-3) (2001) 101–113. doi:10.1023/A:1010339220243.
URL http://dx.doi.org/10.1023/A%3A1010339220243
8. Henk, Richter-Gebert, Ziegler, Basic properties of convex polytopes, in: J. E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press, 1997, 2004, Vol. 2, 2004.
9. R. Schneider, Recent results on random polytopes, in: Boll. Un. Mat. Ital., Vol. 1, 2008, pp. 17–39.
10. B. Mirtich, Fast and accurate computation of polyhedral mass properties, Journal of Graphics Tools 1 (2), iSSN 1086-7651.
11. Z. J. Wang, Improved Formulation for Geometric Properties of Arbitrary Polyhedra, AIAA Journal 37 (1999) 1326–1327. doi:10.2514/2.604.
12. A. Zaslavski, M. Powell, The NEWUOA software for unconstrained optimization without derivatives, in: P. Pardalos, G. Pillo, M. Roma (Eds.), Large-Scale Nonlinear Optimization, Vol. 83 of Nonconvex Optimization and Its Applications, Springer US, Boston, 2006, Ch. 16, pp. 255–297. doi:10.1007/0-387-30065-1_16.
URL http://dx.doi.org/10.1007/0-387-30065-1_16
13. J. A. Nelder, R. Mead, A Simplex Method for Function Minimization, Comput J 7 (4) (1965) 308–313. doi:10.1093/comjnl/7.4.308.
URL http://dx.doi.org/10.1093/comjnl/7.4.308
14. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
15. F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, Comp. Opt. and Appl 51 (1) (2012) 259–277.
URL http://dx.doi.org/10.1007/s10589-010-9329-3
16. D. R. Jones, C. D. Perttunen, B. E. Stuckman, Lipschitzian optimization without the Lipschitz constant, J. Optim. Theory Appl. 79 (1) (1993) 157–181. doi:10.1007/BF00941892.
URL http://dx.doi.org/10.1007/BF00941892
17. Wikipedia, Mathematical Optimization: P Versus NP Problem, Pareto Efficiency, Optimization, Operations Research, Genetic Algorithm, Least Squares, General Books LLC, 2011.
URL http://books.google.com/books?id=WBMgSgAACAAJ
18. S. G. Krantz, The history and concept of mathematical proof (2007).
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.8702; http://www.math.wustl.edu/~sk/eolss.pdf
19. P. L. Varkonyi, G. Domokos, Static equilibria of rigid bodies: Dice, pebbles, and the poincare-hopf theorem.
URL http://dx.doi.org/10.1007/s00332-005-0691-8
20. G. Das, M. T. Goodrich, On the complexity of optimization problems for 3-

dimensional convex polyhedra and decision trees, Comput. Geom. Theory Appl 8 (1995) 8–123.

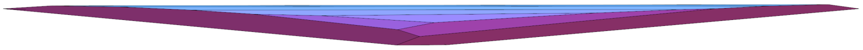
21. E. A. Bender, The number of three-dimensional convex polyhedra, Amer. Math. Monthly 94 (1987) 7–21.
22. R. Dawson, W. Finbow, What shape is a loaded die?, The Mathematical Intelligencer 21 (2) (1999) 32–37. doi:10.1007/BF03024844.
URL `http://dx.doi.org/10.1007/BF03024844`

Appendix A. Prove of unistable property for the found polyhedron with 14 faces (Mathematica notebook)

```
(* vertex is an intersection of 3 planes
   defined by their v[[i]] indices into list of normals n
 *)
Vertex[i_Integer, v_List, n_List] :=
Module[{x, y, z, p = {x,y,z}},
 p /. First[Solve[Thread[Dot[p - #, #]&/@ n[[v[[i]]]] == 0], p]]
];

(* geometric center of f[[i]] face *)
VertexCenter[i_Integer, f_List, v_List, n_List] :=
Module[{c = {0,0,0}, j},
 For[j = 1, j <= Length[f[[i]]], j++,
  c += Vertex[f[[i,j]], v, n];
 ];
 (* return *) c/Length[f[[i]]]
];

Centroid[f_List, v_List, n_List] :=
Module[{c = {0,0,0}, m = 0, v0, v1, v2, a2, i, j},
 For[i = 1, i <= Length[f], i++,
  v0 = Vertex[f[[i,1]], v, n];
  v1 = Vertex[f[[i,2]], v, n];
  For[j = 3, j <= Length[f[[i]]], j++,
   v2 = Vertex[f[[i,j]], v, n];
   a2 = Abs[v2.(v0 \[Cross] v1)];
   c += (v0 + v1 + v2) * a2;
   m += a2;
   v1 = v2;
  ];
 ];
 (* return *) c/(4*m)
];

(* return the projected *point* in the plane *)
ProjectToPlane[p_List, n_List] := p + ((n-p).n/n.n) * n;

(* return the *vector* from the line to the point *)
ProjectToLine[p_List, l0_List, l1_List] :=
 (p-l0) - (Dot[l1-l0, p-l0]/Dot[l1-l0, l1-l0]) * (l1-l0);
```

```
(* minimal signed distance^2 from edges of the polygon f[[i]]
   to the projection of point c to f[[i]] plane.
   If and only if the distance is positive,
   the projection is inside the polygon.
 *)
Distance[i_Integer, c_List, f_List, v_List, n_List] :=
Module[{
  pc     = ProjectToPlane[c, n[[i]]],
  fc     = VertexCenter[i, f, v, n],
  len    = Length[f[[i]]],
  d2min  = Infinity,
  d2, e2fc, e2pc, v0, v1, j},

 (* see if the projection pc can be separated from
    the center fc by an edge *)
 v0 = Vertex[f[[i,len]], v, n]; (* last one *)
 For[j = 1, j <= len, j++,      (* iterate  *)
  v1     = Vertex[f[[i,j]], v, n];
  e2fc   = ProjectToLine[fc, v0, v1];
  e2pc   = ProjectToLine[pc, v0, v1];
  d2     = e2pc.e2pc * Sign[e2pc.e2fc];
  d2min  = If[d2min < d2, d2min, d2];
  v0     = v1;
 ];
 (* return *) d2min
];

(* verify that
   1. lists f,v,n define a convex polyhedron
   2. it is a unistable one
 *)
Valid[f_List, v_List, n_List] := Module[{points,px,pn,c,dp},
 (* check if there are invalid {0,0,0} normals *)
 If[Fold[Or, False, Thread[Plus@@#&/@Abs[n] == 0]],
  Return[False]];

 (* can all vertices be resolved? *)
 If[Length[Position[Det[n[[v[[#]]]]]&/@
  Range[Length[v]], 0]] != 0, Return[False]];

 (* for each face, its vertices (defined by intersection of
    3 planes) have to have one and only one
    common plane == face plane
  *)
 px = Intersection @@ v[[f[[#]]]]&  /@ Range[Length[f]];
 If[Fold[And, True, Thread[Length /@ px == 1]] == False,
  Return[False]];

 (* are all vertices on one side of all face planes? *)
```

```
  points = Vertex[#, v, n] & /@ Range[Length[v]];
 pn = Outer[(#1 - #2).#1 &, n, points, 1];
 If[Fold[And, True, Thread[Flatten[pn] >= 0]] == False,
  Return[False]];

 (* compute margins *)
 c  = Centroid[f, v, n];
 dp = Distance[#, c, f, v, n] & /@ Range[Length[f]];
 (* verify that there is only one stable face
    (for which the distance > 0)
  *)
 Length[Select[dp, (# > 0) &]] == 1
];

(* describe 14-faced polyhedron and validate its properties:
   faces f are given by lists of their vertices;
   vertices v are implicitly specified by triplets of
   incidental face normals (i.e. as intersection of 3 planes);
   each normal n defines a face plane orthogonal to vector n[[i]]
   and passing through point n[[i]].
 *)
f = {{16,10,14,1,7,17},{13,1,14},{23,5,3,19,18,9,8,24},
     {12,7,1,13},{21,19,3,2,6,22},{6,2,4},{4,2,3,5},{22,6,4,5,23},
     {17,7,12,11,9,18},{24,8,15,20},{15,8,9,11,10,16},
     {10,11,12,13,14},{20,15,16,17,18,19,21},{20,21,22,23,24}};

v = {{1,2,4},{5,6,7},{3,5,7},{6,7,8},{3,7,8},{5,6,8},{1,4,9},
     {3,10,11},{3,9,11},{1,11,12},{9,11,12},{4,9,12},{2,4,12},
     {1,2,12},{10,11,13},{1,11,13},{1,9,13},{3,9,13},{3,5,13},
     {10,13,14},{5,13,14},{5,8,14},{3,8,14},{3,10,14}};

n = {{19,70,-410},{0,289,-186},{-26,-389,-2},{-6,207,-331},
     {13,-329,101},{1,-167,233},{-5,-263,181},{0,-73,253},
     {-27,-85,-443},{0,107,201},{0,181,103},{0,287,-27},
     {32,-394,-134},{0,18,244}};

(* prove that given faces/vertices/normals
   form a unistable polyhedron
 *)
Valid[f,v,n]

True

(* visualize *)
points = Vertex[#, v, n] & /@ Range[Length[v]];
Graphics3D[GraphicsComplex[points, Polygon[f]]]
```



```
(* The expression for 'points' above yields an explicit
   representation for mesh vertices. Togeather with
   the face indices defined by 'f', it can be used to
   export the model into 3D formats like Wavefront obj.
   Note that normals directly computed from 'f' and 'points'
   will have an opposite direction to normals
   defined by 'n', i.e. for each face fi

   fi = 1;
   Normalize[Cross[points[[f[[fi,2]]]] - points[[f[[fi,1]]]],
                   points[[f[[fi,3]]]] - points[[f[[fi,2]]]]]] ==
  -Normalize[n[[fi]]]
 *)

points = {{-1118133957, -301190593, 623525659} / -1718803,
          {297975816, -223307064, 98118468} / 735612,
          {-673030086, -728705334, 37612818} / 1979526,
          {344354332, 36422102, -38695074} / -179536,
          {-2975105691, -151380717, 201240609} / 893661,
          {-905270226, 37256484, -58391328} / -252282,
          {1864529864, -95609436, 1904301748} / -4337944,
          {-5177816860, 87797320, 123350760} / 659360,
          {26431456633, -621334533, -141516979} / -2929163,
          {-4227479168, -184871596, 49277412} / -654512,
          {-6712500320, 262712268, -70025796} / 930096,
          {-8483513296, 443720094, -322851786} / 1637400,
          {-1137363884, 73602414, -59305542} / 273474,
          {2990380819, 233074311, -187800883} / 866001,
          {-6384315680, -108058240, -151816320} / -811520,
          {24428105528, 689051524, 16838532} / 2915672,
          {-4855682872, 2469404080, 4349744016} / -9822248,
          {-5290136854, -3977405774, -4048118406} / 11137102,
          {-298814190, 1629846402, 501450570} / -4126578,
          {-4895427080, -19825280, -175095040} / -719680,
          {8206447740, -75757896, 351156552} / 1408596,
          {-1501838894, 23060596, -73032232} / -290758,
          {-2720846106, -46121192, 146064464} / 581516,
          {3670446990, -16108040, -142264720} / -584740};


(* The following expressions define 15, 16, and 17 faced
   unistable polyhedra (see also Figure 8 for 3D printed models).
   n* variables completely define the bodies;
   f* and v* variables are provided for convinience.
 *)

f15 = {{3,2,1,4},{25,16,15,26},{5,4,1,6},{23,8,9,14,17,24},{11,6,1,2,12},{12,2,3,13},{22,13,3,4,5,7,8,23},
       {10,7,5,6,11},{9,8,7,10},{24,17,16,25},{26,15,18,19},{21,14,9,10,11,12,13,22},
       {20,18,15,16,17,14,21},{19,18,20},{19,20,21,22,23,24,25,26}};

v15 = {{1,3,5},{1,5,6},{1,6,7},{1,3,7},{3,7,8},{3,5,8},{7,8,9},{4,7,9},{4,9,12},{8,9,12},{5,8,12},
       {5,6,12},{6,7,12},{4,12,13},{2,11,13},{2,10,13},{4,10,13},{11,13,14},{11,14,15},{13,14,15},
       {12,13,15},{7,12,15},{4,7,15},{4,10,15},{2,10,15},{2,11,15}};

n15 = {{-64,0,-37},{80,0,0},{-46,0,-52},{29,0,-47},{-74,-2,-18},{-84,0,6},{-90,7,40},{-23,0,-61},
       {2,0,-61},{63,0,-30},{80,-1,35},{-75,-7,50},{40,-6,87},{66,-2,67},{24,8,107}};

f16 = {{25,13,9,5,26},{26,5,4,1,27},{20,1,4,3,8,14},{18,6,12,11,7,8,3,2,19},{17,6,18},{27,1,20,19,2,28},
       {28,2,3,4,5,9,23,22,24},{16,10,12,6,17},{15,7,11,10,16},{14,8,7,15},{23,9,13,21},{10,11,12},
       {24,22,21,13,25},{14,15,16,17,18,19,20},{21,22,23},{24,25,26,27,28}};
```

```
v16 = {{2,3,6},{4,6,7},{3,4,7},{2,3,7},{1,2,7},{4,5,8},{4,9,10},{3,4,10},{1,7,11},{8,9,12},{4,9,12},
       {4,8,12},{1,11,13},{3,10,14},{9,10,14},{8,9,14},{5,8,14},{4,5,14},{4,6,14},{3,6,14},{11,13,15},
       {7,13,15},{7,11,15},{7,13,16},{1,13,16},{1,2,16},{2,6,16},{6,7,16}};

n16 = {{6,-7,-60},{8,-29,-46},{8,-42,-20},{-23,34,88},{-5,3,80},{-3,79,0},{-18,82,-25},{-1,-23,72},
       {6,-56,29},{8,-57,5},{3,14,-63},{1,-44,57},{0,47,-50},{-3,33,74},{0,32,-60},{-1,60,-37}};

f17 = {{21,11,10,2,4,22},{12,2,10,9,13},{25,1,26},{24,3,5,1,25},{14,8,7,15},{28,26,1,5,29},{15,7,6,16},
       {18,16,6,19},{13,9,8,14},{30,4,2,12,20,27},{23,3,24},{29,5,3,23,22,4,30},{19,6,7,8,9,10,11,17},
       {20,12,13,14,15,16,18,17,11,21},{17,18,19},{27,20,21,22,23,24,25,26,28},{27,28,29,30}};

v17 = {{3,4,6},{1,2,10},{4,11,12},{1,10,12},{4,6,12},{7,8,13},{5,7,13},{5,9,13},{2,9,13},{1,2,13},{1,13,14},
       {2,10,14},{2,9,14},{5,9,14},{5,7,14},{7,8,14},{13,14,15},{8,14,15},{8,13,15},{10,14,16},{1,14,16},
       {1,12,16},{11,12,16},{4,11,16},{3,4,16},{3,6,16},{10,16,17},{6,16,17},{6,12,17},{10,12,17}};

n17 = {{0,-11,92},{49,0,-39},{-79,4,-41},{-89,1,-21},{70,0,3},{-52,2,-63},{69,0,20},{63,-1,41},{64,0,-18},
       {20,0,-52},{-95,1,4},{-91,-11,36},{29,-8,79},{28,8,85},{50,-3,62},{-94,12,35},{-21,1,-71}};
```