

Four Mathematical Designs for EGMO 2020, the European Girls' Mathematical Olympiad in the Netherlands

Tom Verhoeff

Dep. Math. & CS, Eindhoven University of Technology; T.Verhoeff@tue.nl

Abstract

I describe four mathematical designs for EGMO 2020, the European Girls' Mathematical Olympiad hosted by the Netherlands: an on-stage collaborative activity for the team parade, a sculpture, an animation movie, and a braided puzzle bracelet. All the designs are related and are (also) intended to attract attention to some nice accessible mathematics with interesting open problems.

Introduction

The European Girls' Mathematical Olympiad (EGMO, [1]) is a competition modeled after the International Mathematical Olympiad (IMO, [5]). The aims of the IMO are (cited from the IMO Regulations):

- to discover, encourage and challenge mathematically gifted young people in all countries;
- to foster friendly international relationships among mathematicians of all countries;
- to create an opportunity for the exchange of information on school syllabuses and practices throughout the world;
- to promote mathematics generally.

According to the EGMO Regulations, the aim of the EGMO is to give more girls an opportunity to perform mathematically on an international stage, and so to discover, encourage and challenge mathematically gifted young women in all European countries.

Each country participating in the EGMO sends a team consisting of a leader, a deputy leader, and (at most) four female high-school students. The EGMO program includes an opening ceremony, two competition days, social events, excursions, and a closing ceremony where awards are presented. The first EGMO was held in Cambridge, United Kingdom, in April 2012, and the ninth EGMO was organized by the Netherlands on April 15–21, 2020 [2]. Due to the Corona-virus pandemic EGMO 2020 was reconceived as a virtual (online) event.

This paper concerns four related designs prepared for EGMO 2020:

- the team parade
- a sculpture
- a computer animation
- a puzzle bracelet

In the following sections, we present the various design constraints, inspirations, considerations, trade-offs, and resulting designs. This paper is written from the author's standpoint as a designer prior to the event.

Team Parade at the Opening Ceremony

EGMO 2020 is expected to have about 60 participating teams [3], most from Europe, but also from elsewhere. During the opening ceremony, each team gets some stage time with their deputy leader and country flag. The organizer's goal is to complete the team parade in about 15 minutes, or 15×60 seconds. Without margin, this gives each team 15 seconds to enter, to be on, and to exit the stage. Every second per team extra makes the team parade last one minute longer. Furthermore, just being on stage is not much fun, neither for the team, nor for the audience. So, we want to offer a framework for the team parade that will make it a smooth and interesting happening.



While attending the opening ceremony of the 50-th IMO in Germany in 2009, I already contemplated the team parade, which back then took a long time and was not very interesting to watch. In [8], I described an activity for an IMO team to perform on stage during the team parade. To summarize:

- The team of six participants lines up on stage wearing colored shirts, say three white on the left and three orange on the right (this is for the Dutch team).
- They perform a carefully choreographed sequence of neighbor swaps, where two adjacent team members of different colors trade places.
- After 19 such swaps, it turns out that they end up with three orange shirts on the left and three white on the right, and all $\binom{6}{3} = 20$ permutations have been shown exactly once.

This proposal was worked out in the lecture performance *Lehmer's Dance* [7]. EGMO teams are smaller, but even then it makes little sense to have every team do the same thing.

I then had the idea to let the teams collaborate on permuting a sequence, where each team does only one or two neighbor swaps. An important question is: What to put in that sequence? If only team members permute, then the maximum number of distinct permutations is 24 (when all members wear distinguishable colors). With 60 participating teams, we would need to triplicate most permutations.

That is why I considered sequences of *five* objects, say by also involving the deputy leader. When all objects are distinguishable, there are $5! = 120$ permutations. In that case, each team would need to do two swaps. That number can be reduced by making some objects indistinguishable. When two objects are made indistinguishable, say by considering the letter sequence **O E G M O**, the number of permutations drops to $\binom{5}{1\ 1\ 1\ 2} = \frac{5!}{2!} = 60$. However, there is a concern that among the permutations of these letters there could be objectionable combinations. Working with colors instead of letters would solve that, but an extra layer of meaning would help to attract attention.

An alternative (eventually chosen by the organizers) is to consider the sequence  **2020**, where  is the EGMO-2020 logo. This sequence has two pairs of indistinguishable objects and, hence, $\binom{5}{1\ 2\ 2} = \frac{5!}{2! \times 2!} = 30$ permutations. That does mean that, roughly, every permutation has to be shown twice. An additional issue is that for this sequence it is impossible to present every permutation exactly once by doing just neighbor swaps [9]. It is possible with two duplications, but not with fewer. Moreover, it can be arranged that with one more swap the initial permutation reappears (a cycle). This actually makes it more attractive than working with **A B C D E** or **O E G M O**, because it involves new and partly unsolved mathematics.

To analyze this further, it is helpful to introduce the *neighbor-swap graph*, which has the permutations as vertices, and neighbor swaps as its edges. That is, two vertices are connected by an edge when the corresponding permutations differ by a single neighbor swap (see Figure 1). We are looking for paths in this neighbor-swap graph that visit all vertices with a minimum number of duplications. It is known that, in general (cf. [9, §2]), this graph admits a Hamiltonian path when *either* the graph is linear (a chain) *or* at least two of the 'color' frequencies are odd. Furthermore, it admits a Hamiltonian *cycle* when in addition there are at least three colors and their frequencies are not one, one, and an even number.

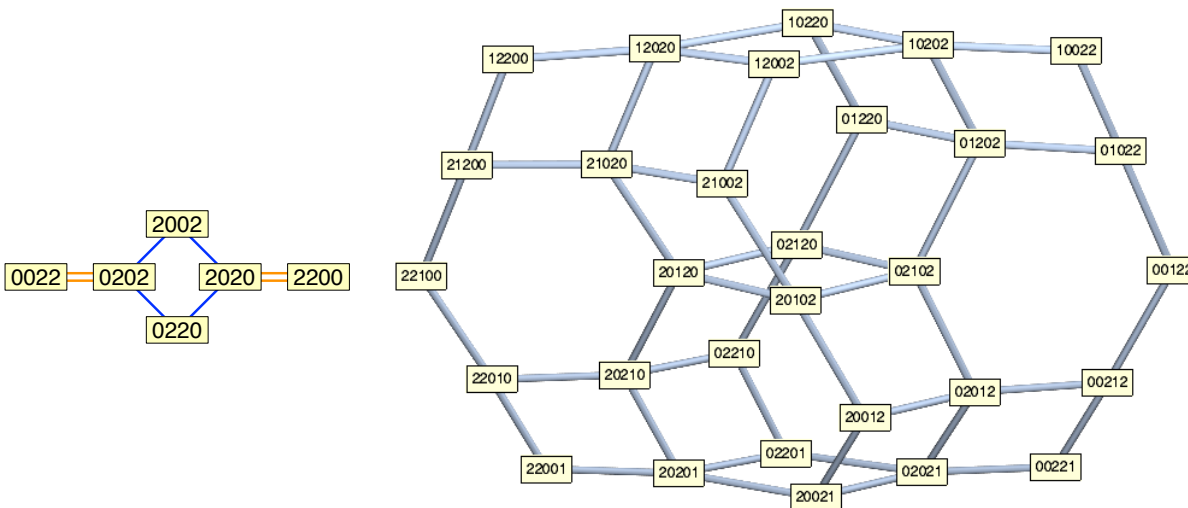


Figure 1: Neighbor-swap graphs for permutations of **2020** (left) and **12020** (right)

Replacing the EGMO-2020 logo by the digit **1**, we are dealing with permutations of **12020** having ‘color’ frequencies [2, 1, 2]. Its neighbor-swap graph (see Figure 1, right) has 30 vertices and 48 edges. It does not admit a Hamiltonian path or cycle (nor does there exist a path that visits each vertex exactly twice). But (this is Lehmer’s 1965 conjecture) it could admit an imperfect Hamiltonian cycle with some vertices that are visited twice in the so-called ‘spur’ pattern, that is, with a single vertex (the spur’s tip) between the duplicates (the spur’s base); cf. Figure 1, left. In [9, Lemma 6 and Conjecture 8], it is conjectured that in general the so-called *stutter permutations* can serve as spur tips. The stutter permutations of **12020** are **00221** and **22001**, having doubled digits from the left. In this particular case, it is easy to verify that such a cycle exists. Figure 2 shows the eight Hamiltonian cycles on the non-stutter permutations, that is, without the two spurs (at the bottom left and right; each spur can be added in two ways, e.g., as ... 22010, 22001, 22010, ...).

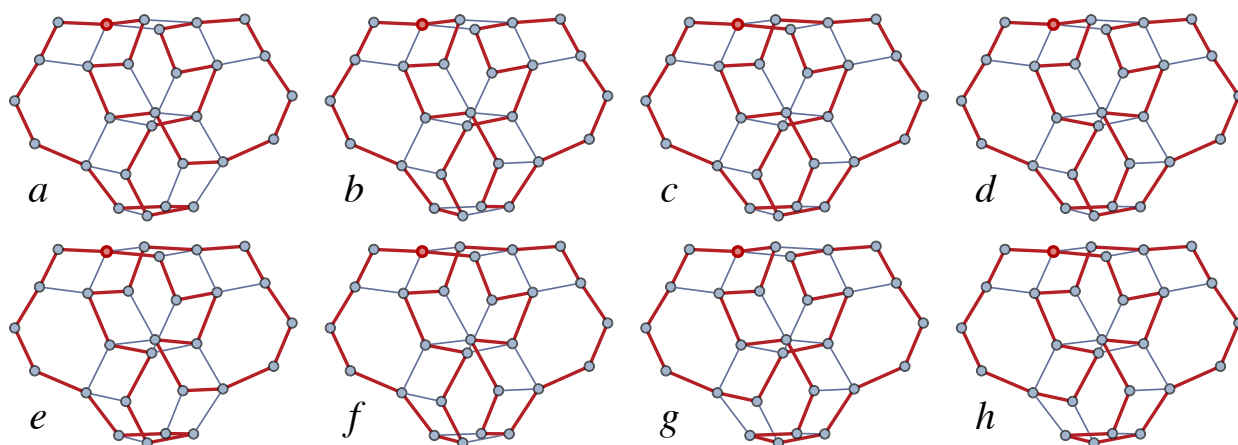


Figure 2: Hamilton cycles on the non-stutter permutations in the neighbor-swap graph for **12020** (which is marked red), in the order that *Mathematica* found them

The neighbor-swap graph in Figure 1 has an order-4 symmetry group generated by (i) swapping the role of **0** and **2** (a half turn about the vertical axis in the figure) and (ii) reversing the permutations (a half turn about the horizontal axis in the figure). The Hamiltonian cycles in Figure 2 match up in two pairs (a–g and e–h) that are related by a half turn about the vertical axis. The remaining and four have this half turn as a symmetry.

To give each team something to do, we can combine two imperfect cycles. For reasons of diversity, I would like to select two cycles such that together they maximize the number of edges visited. By inspection of Figure 2, there are 18 edges common to all cycles (Figure 3, left), and 8 edges appear in none of the cycles (Figure 3, middle). The size of the symmetric difference between pairs of cycles is thus at most $44 - 18 - 8 = 18$. The actual size of the symmetric difference between pairs of cycles is shown in Fig 3 (right).

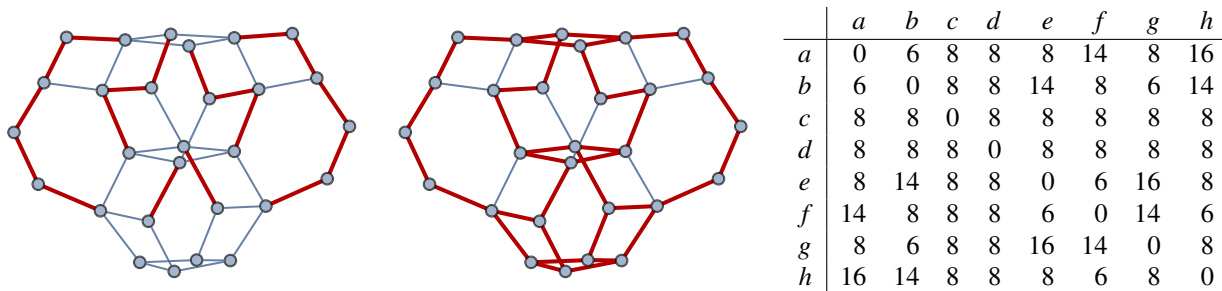


Figure 3: Cycles *a*–*g*: common edges (left), all edges (middle), size of symmetric differences (right)

Somewhat arbitrarily, I chose cycles *e* and *g* achieving the largest symmetric difference of 16. These have two edges in common that are not forced (at the bottom of the graph). There are still various options to include the stutter permutations via spurs. Since the paths will be traversed in a particular direction, it would be good to try to traverse the common edges in opposite direction (as much as possible).

The start and end vertex will be **12020**. I would like to avoid visiting that permutation halfway again, because returning there is the ultimate resolution. This can be accomplished by suitably combining the cycles, and just before closing the first cycle, transfer to the second cycle. Another goal is to delay the spurs, with their double visit, as much as possible. The result is the following walk involving 62 swaps:

```

12020 → 10220 → 01220 → 01202 → 02102 → 20102 → 20012 → 02012 →
00212 → 00221 → 00212 → 00122 → 01022 → 10022 → 10202 → 12002 →
21002 → 21020 → 20120 → 02120 → 02210 → 02201 → 02021 → 20021 →
20201 → 20210 → 22010 → 22001 → 22010 → 22100 → 21200 → 12200 →
12020 → 12200 → 21200 → 22100 → 22010 → 20210 → 02210 → 02120 →
02102 → 01202 → 01220 → 10220 → 10202 → 10022 → 01022 → 00122 →
00212 → 02012 → 02021 → 00221 → 02021 → 02201 → 20201 → 22001 →
20201 → 20021 → 20012 → 20102 → 20120 → 21020 → 21002 → 12002 →
12020

```

The two permutations shown in grey are skipped, to avoid returning to the initial/final vertices in the middle. The underlined vertices are the stutter permutations (the vertices before and after them are the same: unavoidable duplications).

To simplify execution, the four adjacent pairs in a permutation are labeled from left to right as A B C D. Thus, the whole walk can be expressed as:

BADBACAB DDCBACBA DBACDCAC DBDDCBAB BABCBCAD BDADCABC BDBBCABB CDCBDBAD

On stage, there will be five objects showing the EGMO 2020 logo and the digits **2020**. Taped on the stage floor, between those objects are the labels A through D. Just before a team goes on stage, they receive the letter of the swap they need to perform. When there are fewer than 62 teams, the Dutch team will perform multiple swaps. The audience will not be told what is going to happen, nor will the teams know the overall plan. They need to figure this out, as the swaps are performed. This will be supported by the animation discussed below, and an explanation afterwards.

There is one objection to the proposed walk in the graph: the second cycle begins by retracing the first cycle in reverse order for three swaps (see the part . . .CBAB BABC. . .). This can be avoided by ‘turning the graph upside-down’ (by a half-turn about a horizontal axis), and using **10022** and **12200** as spur tips. Then we can use:

DBACAABC DBCDACDB DCAABCDB CDACDBAB BDCCBABA CADABDCB CDBACADA BDCBAACA

Sculpture: 3D Rendering of Neighbor-swap Graph

To get a better grasp of the neighbor-swap graph for the permutations of **00122**, it is helpful to design a 3D rendering that has the same symmetry group (but now in terms of geometric transformations, rather than combinatoric transformations). Symmetries of graph: swap role of 0 and 2; reverse order

A natural 3D embedding is obtained by mapping (for instance) the index of **1** in the permutation into the z -coordinate. Then drop the **1** from the permutation and map the index of the leftmost **0** into the x -coordinate and the index of the rightmost **0** into the y -coordinate. This results in the rendering shown on the far left in Figure 4. Permutation **10022** is placed at $(1, 2, 1)$, and **22001** at $(3, 4, 5)$.

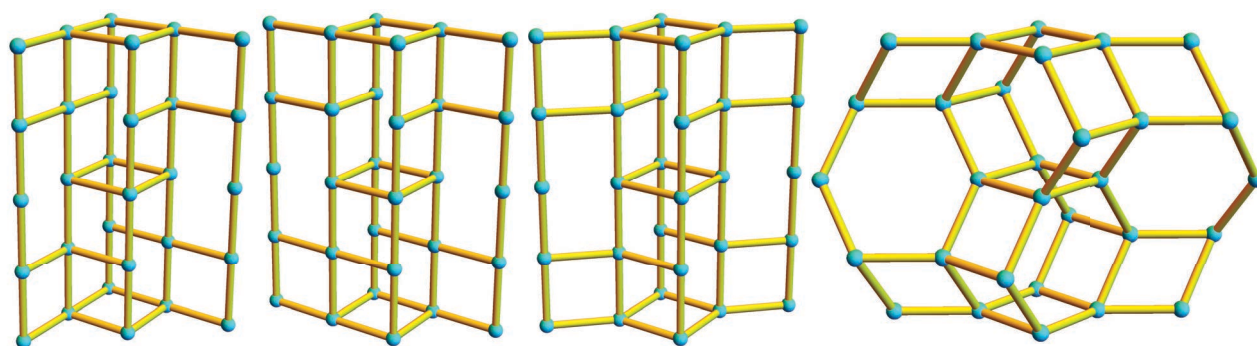


Figure 4: 3D renderings of the neighbor-swap graph for **00122**

This has only one (non-trivial) symmetry: a half-turn about the horizontal axis $(1, 1, 0)$. By flipping the left wing backwards (Figure 4, middle left), we obtain a rendering that has a half-turn about the vertical axis as a symmetry, but no longer the horizontal half-turn. Putting the two wings in the halfway position (Figure 4, middle right) results in a structure with both the horizontal and vertical half-turns as symmetries. Neighbor-swap graphs turn out to have only two types of shortest cycles: quadrangles and hexagons. The hexagonal cycles can be made more prominent by squeezing the design vertically, keeping all the edge lengths the same (Figure 4 far right).

The latter design can be parameterized by two angles: the angle 2α of the quadrangle on the base (which is in fact is rhombus) and the angle 2β of the four central hexagons. This forces the angle 2γ of the hexagons in the wings. Unfortunately, β and γ cannot be made equal, unless the rhombs degenerate (zero area). But the areas of the hexagons can be equalized (with $2\beta \approx 108.8^\circ$ and $2\gamma \approx 131.4^\circ$, as shown in Figure 4).

I am still investigating the possibility to construct the last design using branching miter joints [10] and polygonal beams (preferably with a square or triangular cross section) such that all the longitudinal beam edges nicely meet at the joints.

Computer Animation

As visual support for the team parade and as a further illustration of walks in neighbor-swap graphs, I made a computer animation. The animation starts with an empty scene, and as the team parade progresses, the

relevant edges and vertices are added to the scene. Halfway, all vertices will have appeared. The second walk through the graph removes edges and vertices as they are revisited, tearing down the graph until it has completely disappeared.

The camera starts zoomed in on the first vertex and zooms out as more vertices appear. Then the camera starts to rotate around the (partial) graph, giving the viewer a better feeling for its three-dimensional structure. This rotation will also gradually reveal the storage location of all the parts. Since not all edges are traversed by the two walks, I thought it would be a good idea to partly explore edges incident on the current vertex. For that purpose, all edges consist of two cylinders that are controlled separately. Only when an untraversed edge has been explored from both ends will it (dis)appear fully.

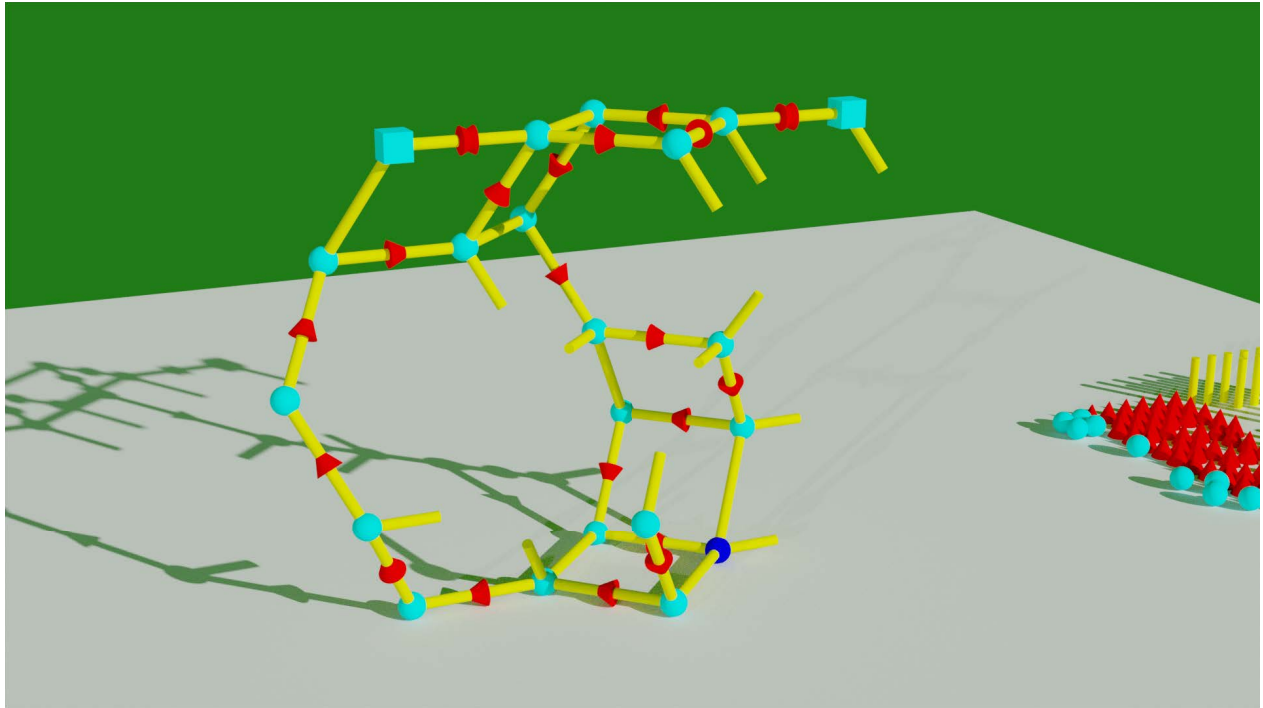


Figure 5: *Frame from the computer animation (draft version)*

The animation was programmed in Python and rendered using Blender [6]. A preliminary version of the movie can be found as supplemental material in the Bridges archive.

Braided Puzzle Bracelet

Another kind of animation that could be done involves rendering the objects as they are being permuted. This is like a dance, where the objects repeatedly change places in pairs. Figure 6 (top) shows a film of such a dance for four colored dots arranged vertically, with one frame per permutation. By connecting the dots horizontally, we get a continuous view of time: the scene is one-dimensional (the y -coordinate) and time is the other dimension (the x -coordinate); see Figure 6 (middle).

The scene can also be made two-dimensional, in which case the choreographer has to decide how objects switch places: they move around each other, either clockwise (CW) or counter-clockwise (CCW). The resulting continuous film can be viewed as a braid (Figure 6, bottom), where each object corresponds to a strand. To create a true braid, where over-under passings alternate, the choices for CW and CCW must be made appropriately.

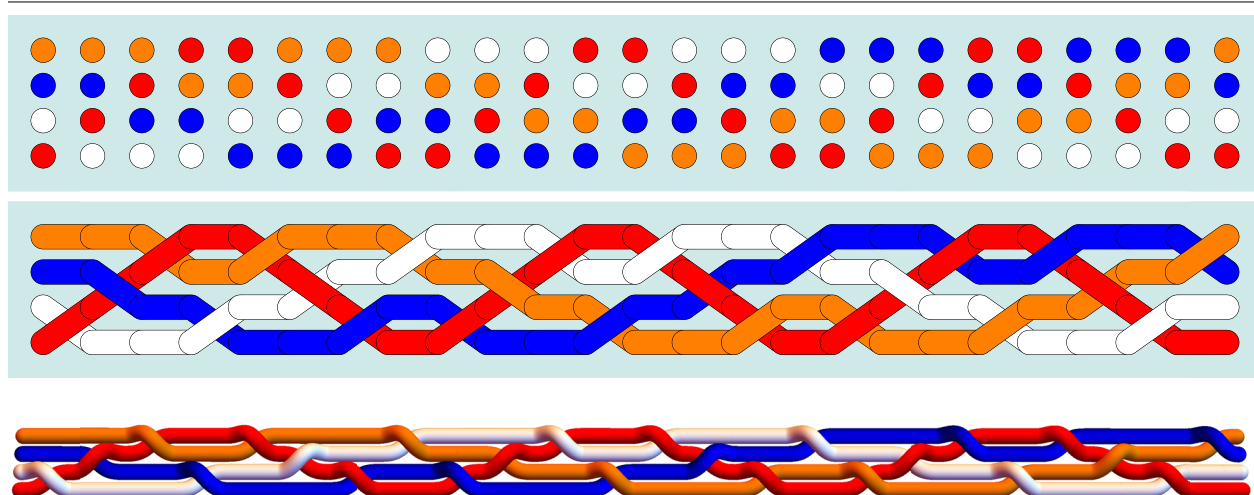


Figure 6: Cycle of 24 permutations of 4 colored objects: dots (top), 2D braid (middle), 3D braid (bottom)

When animating such a dance, there are various ways in which location can change over time. That is, there are all kinds of velocity profiles imaginable, and these determine the look of the crossing. Figure 7 shows three velocity profiles: circular (constant angular velocity), straight (projection is straight), helical (smooth start and stop), helical flattened (where under-strand stays flat, and the over-strand moves more up).

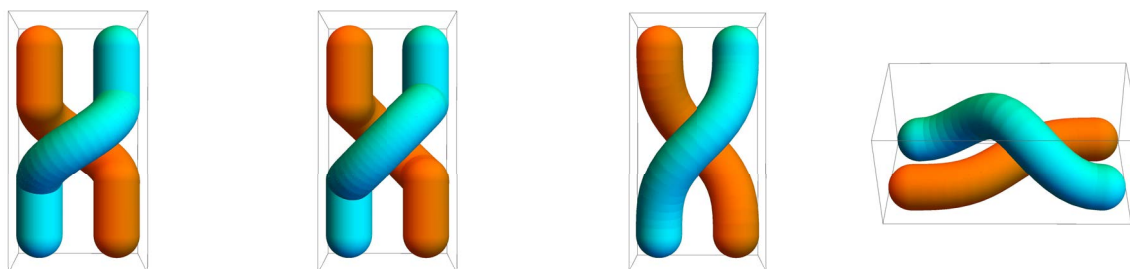


Figure 7: Various velocity profiles for an exchange (crossing): circular, straight, helical, helical+flattened

The flattened version of the braid can be wound into a bracelet (technically, a bangle; Figure 8, left and middle). To ease putting it on and taking it off and also to allow unweaving, the strands have been cut. The cuts were chosen such that they do not appear on the outer strands, to reduce the risk catching on something. Moreover, by not aligning the cuts, the puzzle of weaving the separate strands together becomes more challenging. Aligned cuts would also be esthetically less pleasing.

Initially, we thought of applying this to the permutations of **12020**, giving rise to a bangle with five strands. We did not pursue this for several reasons. As a puzzle it is considerably harder because of duplicate colors. The extra strand increased the cost. The four-strand version is based on the elegant and understandable recursive Steinhaus–Johnson–Trotter algorithm. The plan is to produce monochrome sets in five different colors, each team getting bangles in only one color. They are then encouraged to exchange strands with other teams, to obtain multi-colored bangles. That way, the puzzle bangles also serve as a social mixing catalyst.

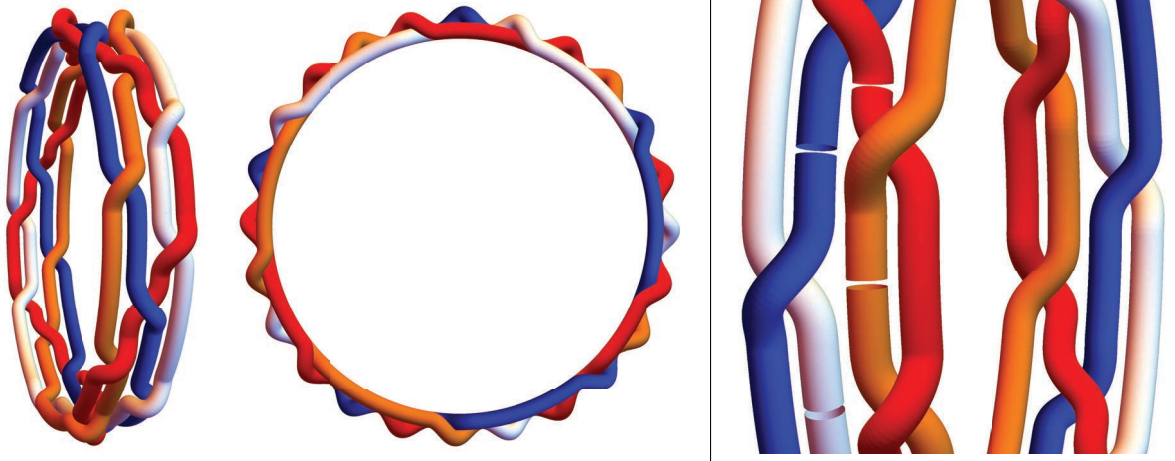


Figure 8: *Braid wound into a bangle; close-up of cuts (right)*

Conclusions

I described four mathematical designs for the European Girls' Mathematical Olympiad (EGMO) in the Netherlands: the team parade, a sculpture, a computer animation, and a braided puzzle bracelet. All these designs are based on the combinatorics of permuting a sequence of objects. A secondary goal is to attract attention to a nice piece of accessible mathematics with interesting open problems.

Stringent international measures to curb spreading of the COVID-19 virus led to cancelation of the physical event. But the organizers reconceived EGMO 2020 as a virtual (online) event. The opening ceremony can be found on YouTube [4], including a virtual team parade (at 09:58), and an explanation by the author (at 15:15). The sculpture and bangles have only been prototyped but have not (yet) been produced.

References

- [1] "EGMO – European Girls' Mathematical Olympiad." <https://www.egmo.org/>
- [2] "EGMO 2020 in the Netherlands (home page)." <https://egmo2020.nl/>
- [3] "EGMO 2020 Statistics." <https://www.egmo.org/egmos/egmo9/>
- [4] "EGMO 2020 Virtual Opening Ceremony." <https://youtu.be/cnZ5nw4jQJc>
- [5] "IMO – International Mathematical Olympiad." <https://www.imo-official.org/>
- [6] Blender Foundation. "Blender." <https://blender.org/>
- [7] R. van Berkel and T. Verhoeff. "Lehmer's Dance – A Lecture Performance." *Bridges Conference Proceedings*, Linz, Austria, July 16–20, 2019, pp. 375–378. <http://archive.bridgesmathart.org/2019/bridges2019-375.html>
- [8] T. Verhoeff. "Combinatorial Choreography." *Bridges Conference Proceedings*, Towson MD, USA, August 25–29, 2012, pp. 607–612. <http://archive.bridgesmathart.org/2012/bridges2012-607.html>
- [9] T. Verhoeff. "The spurs of D. H. Lehmer: Hamiltonian Paths in Neighbor-swap Graphs of Permutations." *Designs, Codes and Cryptography*, vol. 84, no. 1, Jul. 2017, pp. 295–310. <https://doi.org/10.1007/s10623-016-0301-9>
- [10] T. Verhoeff and K. Verhoeff. "Branching Miter Joints: Principles and Artwork." *Bridges Conference Proceedings*, Pécs, Hungary, Jul. 24–28, 2010, pp. 27–34. <http://archive.bridgesmathart.org/2010/bridges2010-27.html>