A dynamic programming approach to generation of strong traces

Nino Bašić
Faculty of Mathematics, Natural Sciences and Information Technologies
University of Primorska

32nd TBI Winterseminar
Bled, 16 February 2017

Polyhedral self-assembly

Disclaimer: I am a mathematician.

In 2013 Gradišar et al. (National Institute of Chemistry, Slovenia) successfully designed a self-assembly tetrahedral polypeptide called TET12.

It is a linear chain of twelve peptides, separated by flexible links, such that certain pairs of peptides "glued" together and formed coiled coil dimers.

The end result was a stable tetrahedron in which each of its six edges was a coiled coil dimer.

Polyhedral self-assembly

Disclaimer: I am a mathematician.

In 2013 Gradišar et al. (National Institute of Chemistry, Slovenia) successfully designed a self-assembly tetrahedral polypeptide called TET12.

It is a linear chain of twelve peptides, separated by flexible links, such that certain pairs of peptides "glued" together and formed coiled coil dimers.

The end result was a stable tetrahedron in which each of its six edges was a coiled coil dimer.

Polyhedral self-assembly

Every peptide is a chain composed of several amino acids. The TET12 is composed of 476 amino acids and can be encoded as a string of length 476 using one-letter codes:

$$\begin{aligned}\mathrm{TET12} = {} & \mathrm{START} + \mathrm{APH} + \mathrm{LINK} + \mathrm{P3} + \mathrm{LINK} + \mathrm{BCR} + \mathrm{LINK} + \mathrm{GCN_{sh}} + \\ & \mathrm{LINK} + \mathrm{APH} + \mathrm{LINK} + \mathrm{P7} + \mathrm{LINK} + \mathrm{GCN_{sh}} + \mathrm{LINK} + \mathrm{P4} + \\ & \mathrm{LINK} + \mathrm{P5} + \mathrm{LINK} + \mathrm{P8} + \mathrm{LINK} + \mathrm{BCR} + \mathrm{LINK} + \mathrm{P6} + \mathrm{STOP},\end{aligned}$$

where

$$\begin{aligned}\mathrm{APH} &= \texttt{"MKQLEKELKQLEKELQAIEKQLAQLQWKAQARKKKLAQLKKKLQA"},\\ \mathrm{BCR} &= \texttt{"DIEQELERAKASIRRLEQEVNQERSRMAYLQTLLAK"},\\ \mathrm{GCN_{sh}} &= \texttt{"QLEDKVEELLSKNYHLENEVARLKKLVG"},\\ \mathrm{P3} &= \texttt{"SPEDEIQQLEEEIAQLEQKNAALKEKNQALKYG"},\\ \mathrm{P4} &= \texttt{"SPEDKIAQLKQKIQALKQENQQLEEENAALEYG"},\\ \mathrm{P5} &= \texttt{"SPEDENAALEEKIAQLKQKNAALKEEIQALEYG"},\end{aligned}$$

Polyhedral self-assembly

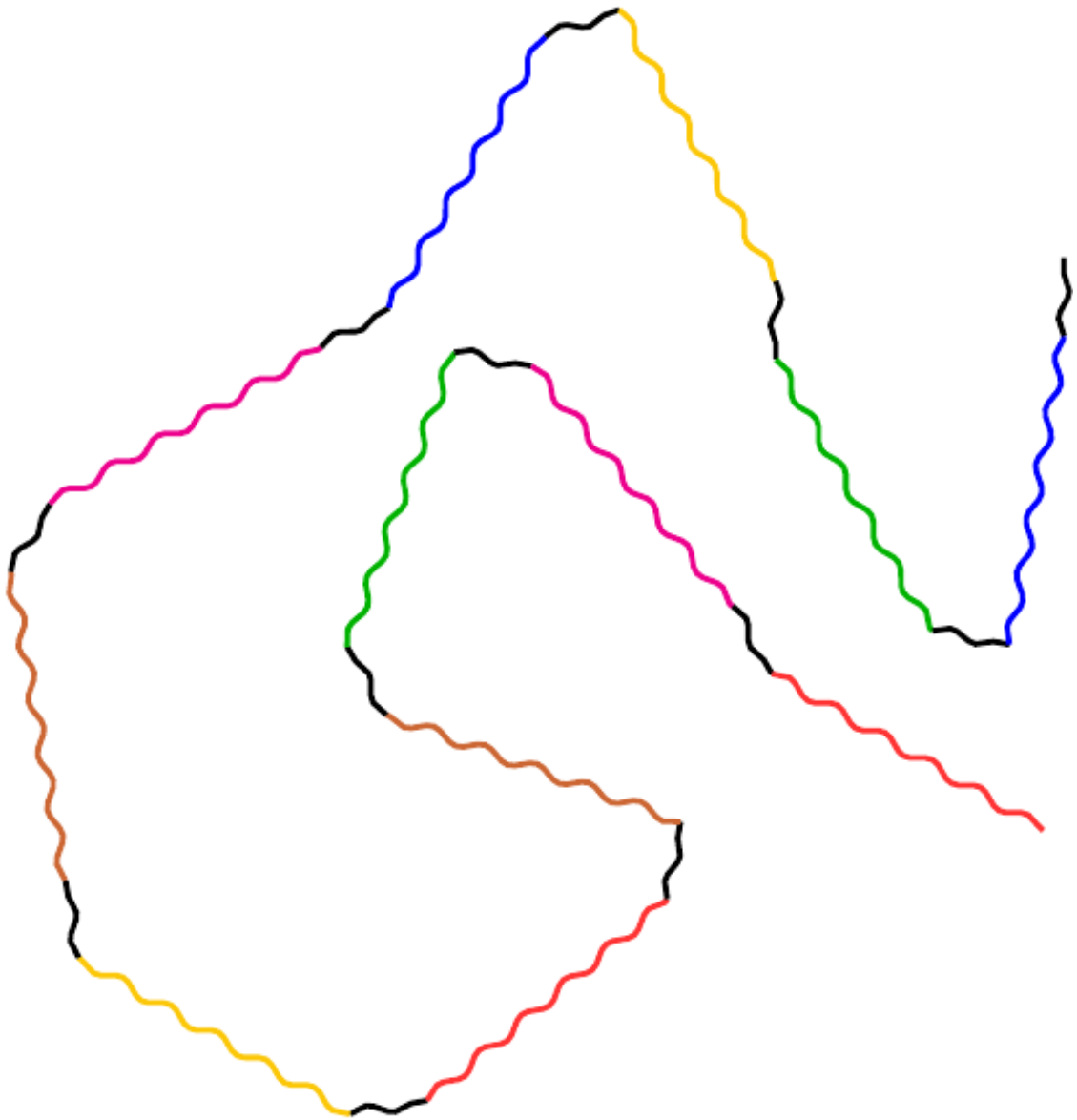
Every peptide is a chain composed of several amino acids. The TET12 is composed of 476 amino acids and can be encoded as a string of length 476 using one-letter codes:

$$\begin{aligned}\text{TET12} = {} & \text{START} + \text{APH} + \text{LINK} + \text{P3} + \text{LINK} + \text{BCR} + \text{LINK} + \text{GCN}_{\text{sh}} + \\ & \text{LINK} + \text{APH} + \text{LINK} + \text{P7} + \text{LINK} + \text{GCN}_{\text{sh}} + \text{LINK} + \text{P4} + \\ & \text{LINK} + \text{P5} + \text{LINK} + \text{P8} + \text{LINK} + \text{BCR} + \text{LINK} + \text{P6} + \text{STOP},\end{aligned}$$

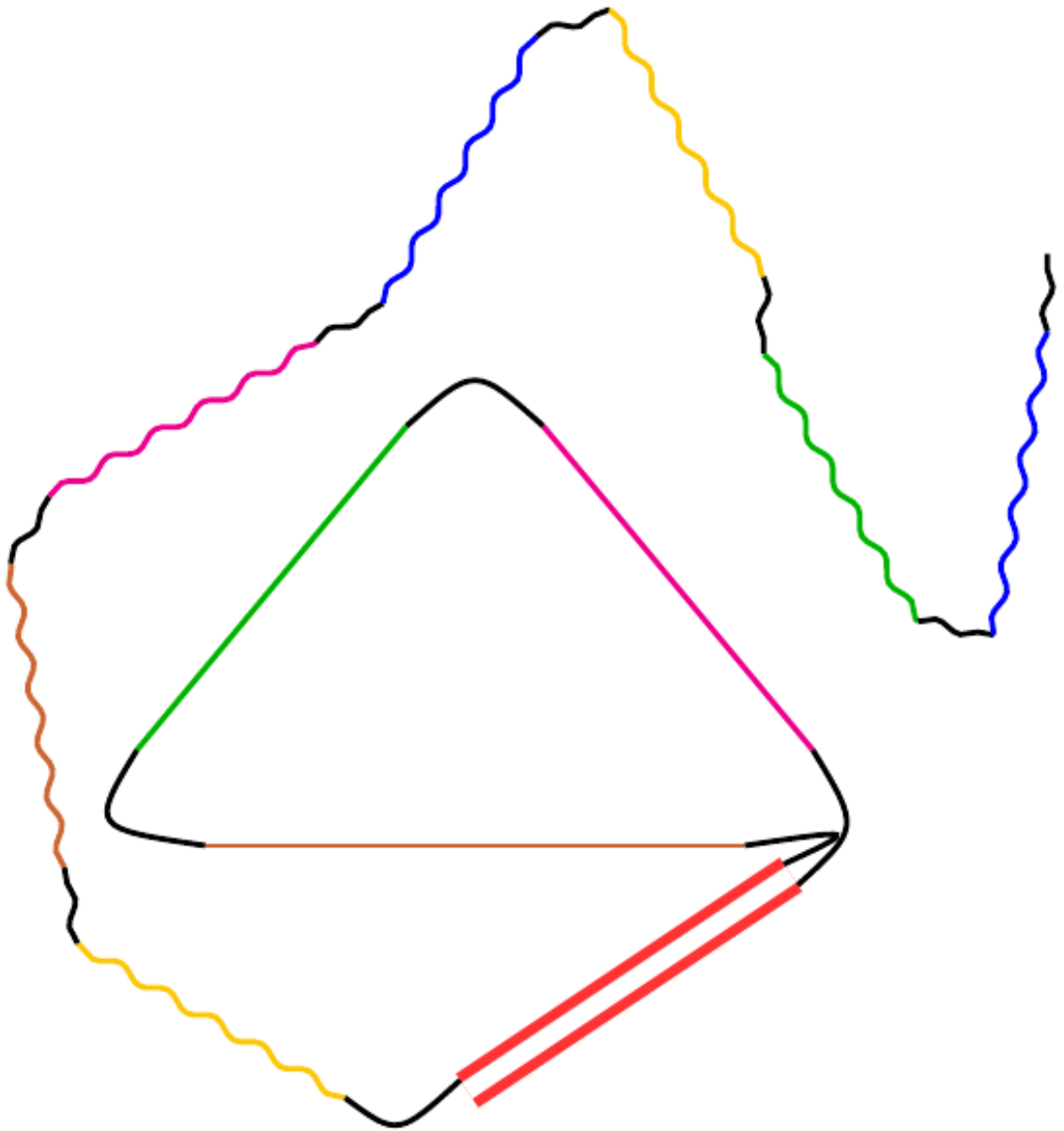
where

$$\begin{aligned}\text{P6} &= \texttt{"SPEDKNAALKEEIQALEEENQALEEKIAQLKYG"},\\ \text{P7} &= \texttt{"SPEDEIQALEEKNAQLKQEIAALEEKNQALKYG"},\\ \text{P8} &= \texttt{"SPEDKIAQLKEENQQLEQKIQALKEENAALEYG"},\\ \text{START} &= \texttt{"MYHHHHHHSRAG"},\\ \text{LINK} &= \texttt{"SGPG"} \text{ and}\\ \text{STOP} &= \texttt{"SGTS"}.\end{aligned}$$

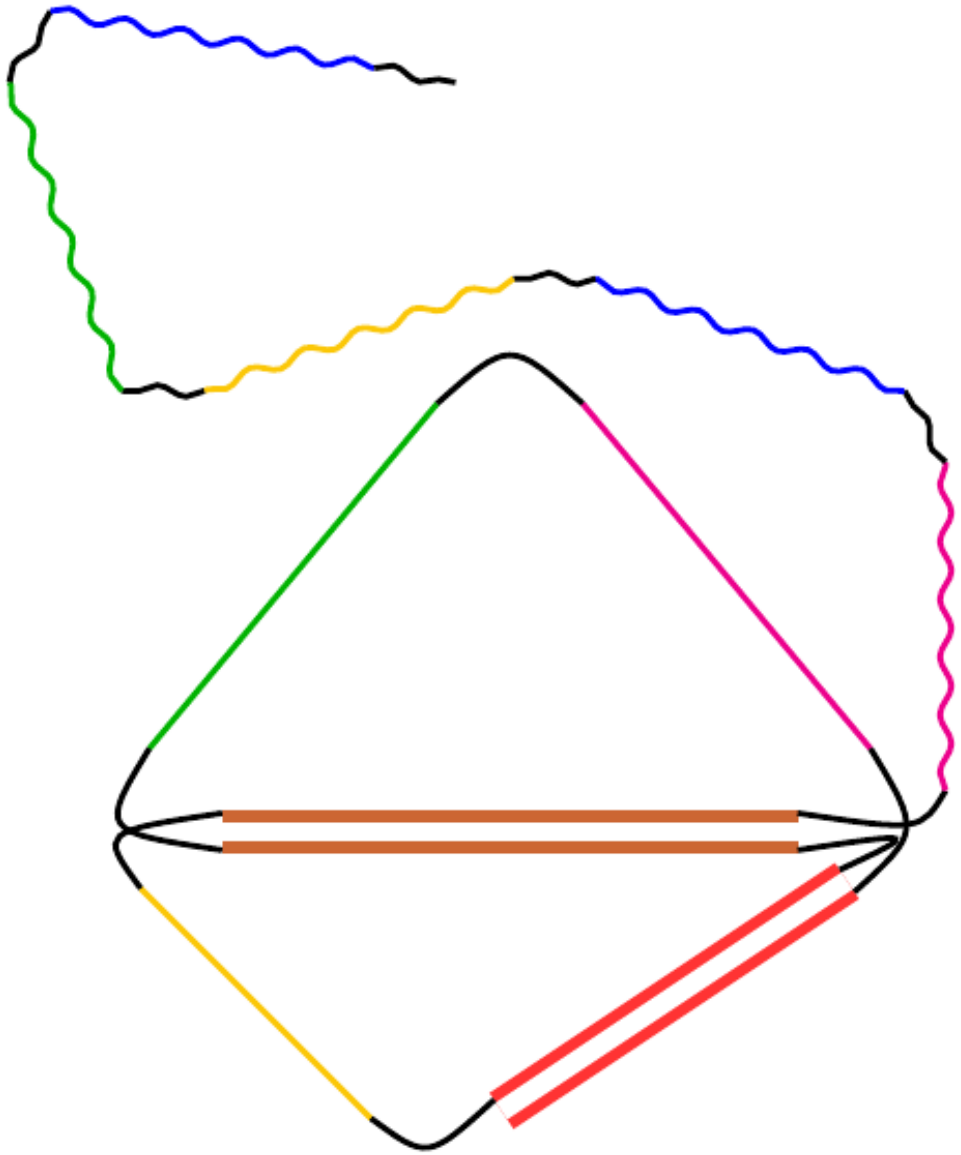
Polyhedral self-assembly



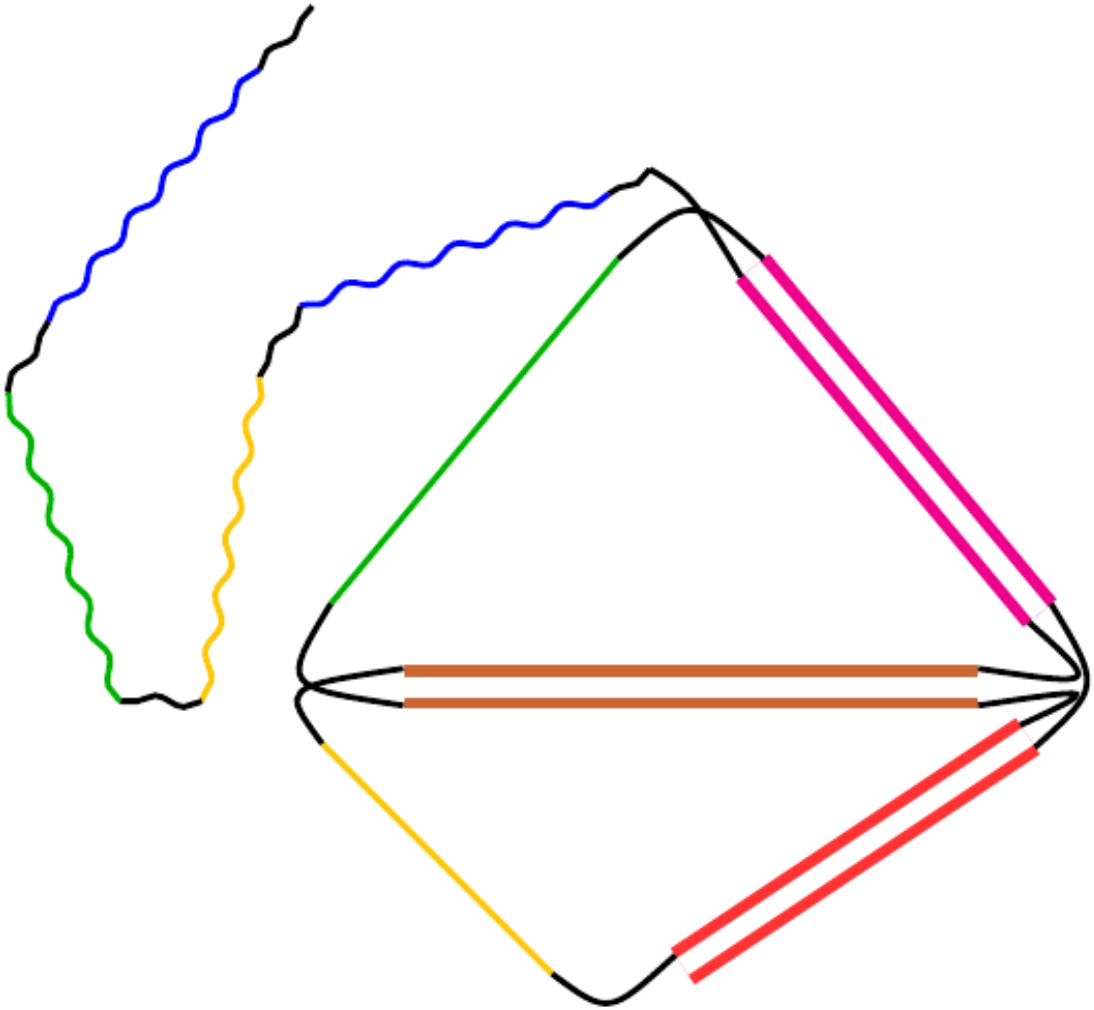
Polyhedral self-assembly



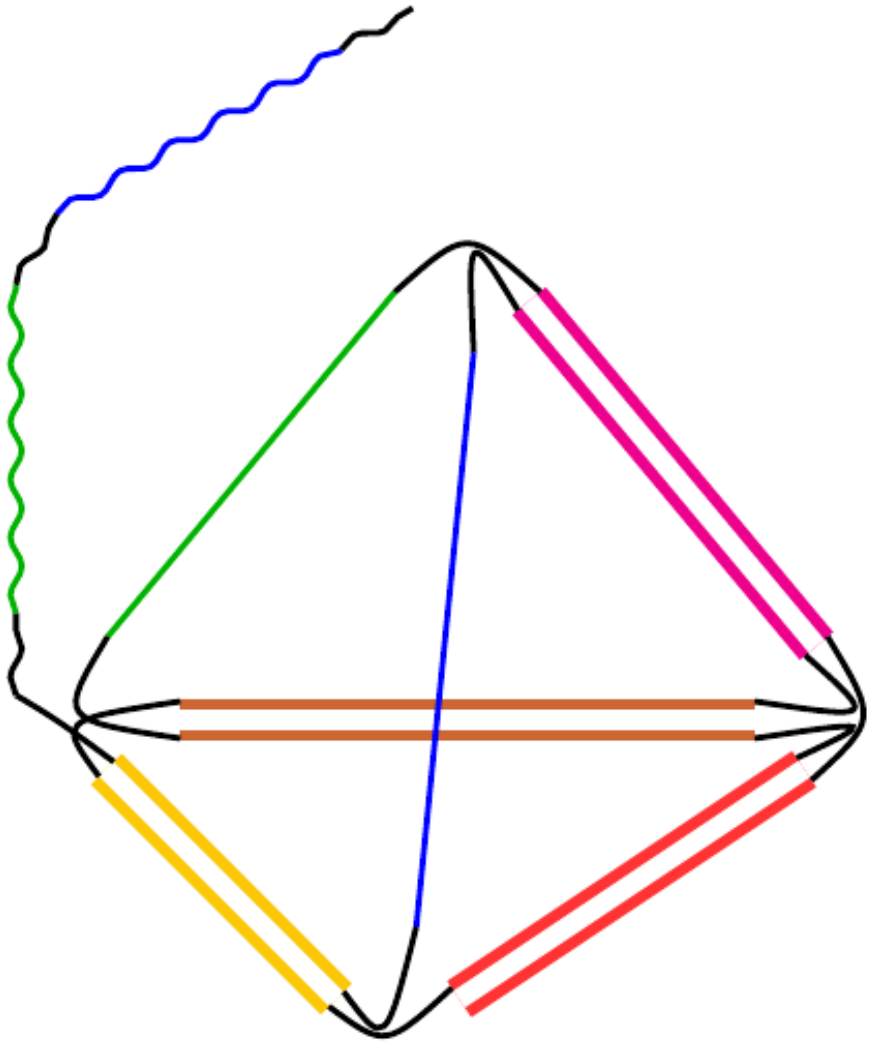
Polyhedral self-assembly



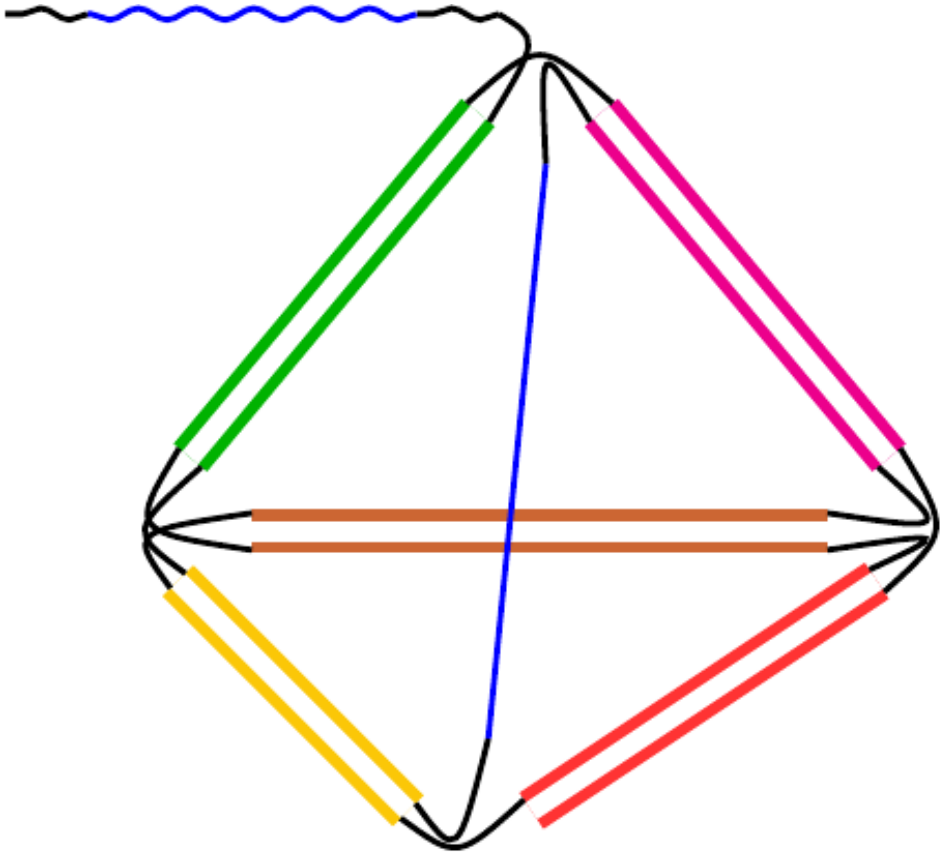
Polyhedral self-assembly



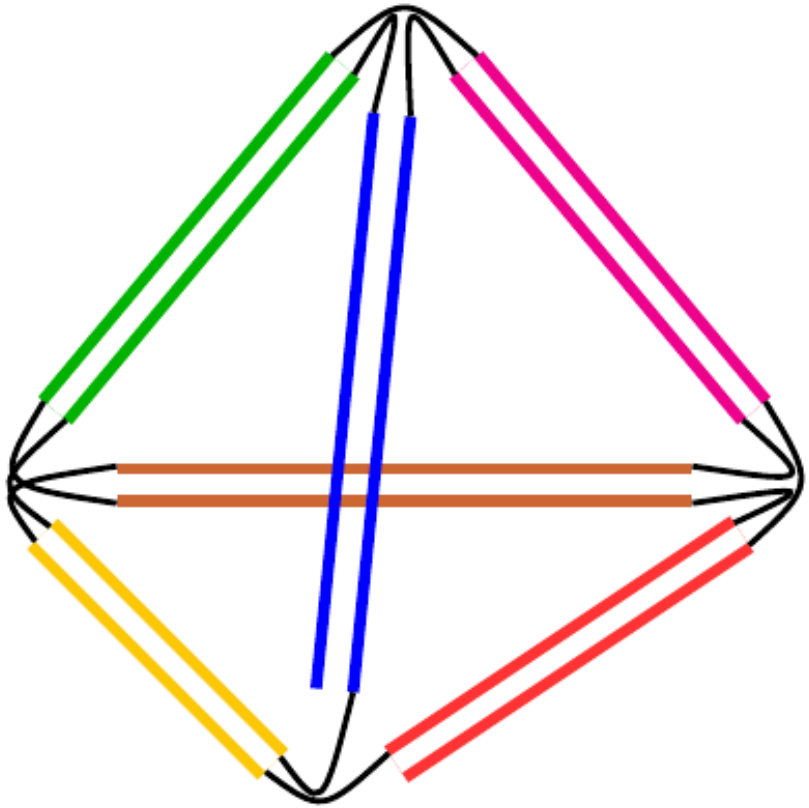
Polyhedral self-assembly



Polyhedral self-assembly



Polyhedral self-assembly



Multiset of orthogonal peptide pairs

Some of the peptide pairs in this chain are "compatible", i.e., they will interlock and form stable coiled coil dimers. On the other hand, those pairs that do not have strong affinity to each other will not form a coiled coil dimer. Let $\mathcal{P}$ be a multiset of $2m$ peptides. In the case of TET12 the multiset $\mathcal{P}$ is

$$\{\text{APH}, \text{APH}, \text{BCR}, \text{BCR}, \text{GCN}_{\text{sh}}, \text{GCN}_{\text{sh}}, \text{P3}, \text{P4}, \text{P5}, \text{P6}, \text{P7}, \text{P8}\}.$$

We can imagine a graph with a vertex for every peptide in $\mathcal{P}$ and an edge between two of them if and only if they are compatible. If this graph is a disjoint union of m copies of K_2 it induces a partition on $\mathcal{P}$ into pairs. Then $\mathcal{P}$ is called a multiset of orthogonal peptide pairs.

Multiset of orthogonal peptide pairs

The problem of determining whether a pair of peptides forms a stable coiled coil dimer and the problem of finding a large multiset of orthogonal peptide pairs are both interesting from a chemical point of view.

In our model this information will be given in advance. We know from experimental evidence that the following pairs are orthogonal peptide pairs:

$(\mathrm{P3}, \mathrm{P4}), (\mathrm{P5}, \mathrm{P6}), (\mathrm{P7}, \mathrm{P8}), (\mathrm{GCN}_{\mathsf{sh}}, \mathrm{GCN}_{\mathsf{sh}}), (\mathrm{APH}, \mathrm{APH})$ and $(\mathrm{BCR}, \mathrm{BCR})$.

Certain pairs consist of two copies of the same peptide. They are called homodimers. Otherwise they are called heterodimers.

This information alone is not sufficient to fully describe the glueing process that leads to the tetrahedron.

Multiset of orthogonal peptide pairs

The problem of determining whether a pair of peptides forms a stable coiled coil dimer and the problem of finding a large multiset of orthogonal peptide pairs are both interesting from a chemical point of view.

In our model this information will be given in advance. We know from experimental evidence that the following pairs are orthogonal peptide pairs:

$(\mathrm{P3}, \mathrm{P4}), (\mathrm{P5}, \mathrm{P6}), (\mathrm{P7}, \mathrm{P8}), (\mathrm{GCN}_{\mathsf{sh}}, \mathrm{GCN}_{\mathsf{sh}}), (\mathrm{APH}, \mathrm{APH})$ and $(\mathrm{BCR}, \mathrm{BCR})$.

Certain pairs consist of two copies of the same peptide. They are called homodimers. Otherwise they are called heterodimers.

This information alone is not sufficient to fully describe the glueing process that leads to the tetrahedron.

Multiset of orthogonal peptide pairs

The problem of determining whether a pair of peptides forms a stable coiled coil dimer and the problem of finding a large multiset of orthogonal peptide pairs are both interesting from a chemical point of view.

In our model this information will be given in advance. We know from experimental evidence that the following pairs are orthogonal peptide pairs:

$(\mathrm{P3}, \mathrm{P4}), (\mathrm{P5}, \mathrm{P6}), (\mathrm{P7}, \mathrm{P8}), (\mathrm{GCN}_{\mathsf{sh}}, \mathrm{GCN}_{\mathsf{sh}}), (\mathrm{APH}, \mathrm{APH})$ and $(\mathrm{BCR}, \mathrm{BCR})$.

Certain pairs consist of two copies of the same peptide. They are called homodimers. Otherwise they are called heterodimers.

This information alone is not sufficient to fully describe the glueing process that leads to the tetrahedron.

The glueing process

The polypeptide chain TET12 is directed (from START to STOP) and so are all peptides along it.

Two peptides may be glued together in such way that they both point in the same direction. In this case they form a parallel dimer. If they point in opposite directions, they form an anti-parallel dimer.

The problem of determining whether a given dimer is parallel or anti-parallel resides outside our current model. This information is provided as input data.

For the case of TET12, we know that (APH, APH) and (BCR, BCR) are antiparallel dimers, whilst, all other pairs are parallel dimers.

The glueing process

The polypeptide chain TET12 is directed (from START to STOP) and so are all peptides along it.

Two peptides may be glued together in such way that they both point in the same direction. In this case they form a parallel dimer. If they point in opposite directions, they form an anti-parallel dimer.

The problem of determining whether a given dimer is parallel or anti-parallel resides outside our current model. This information is provided as input data.

For the case of TET12, we know that (APH, APH) and (BCR, BCR) are antiparallel dimers, whilst, all other pairs are parallel dimers.

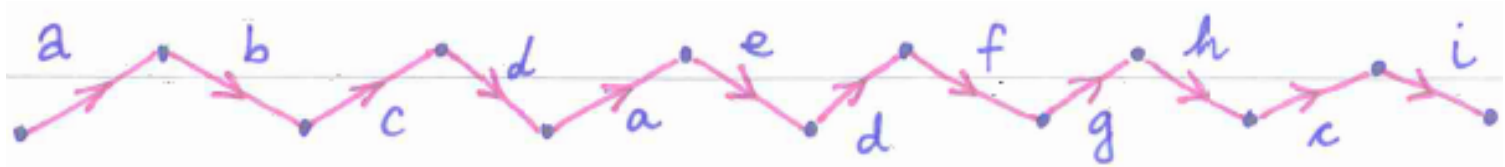
The glueing process

The polypeptide chain is modeled with a labeled directed path of length $2m$, where m is the number of peptide pairs. Instead of names $\mathrm{APH}, \mathrm{P3}, \mathrm{BCR}, \ldots$ we will use letter $a, b, c, \ldots$
Let $\vec{P}_{2m+1}$ be the directed path on $2m+1$ vertices. (It contains $2m$ arcs which represent peptides.) Let Σ be an alphabet with one symbol for each different peptide. For the case of $\mathrm{TET12}$,

$$\Sigma = \{a, b, c, d, e, f, g, h, i\}.$$

The map $w\colon A(\vec{P}_{2m+1}) \to \Sigma$ assigns a symbol to each arc. This labeled digraph can be represented as the sequence

$$(a, b, c, d, a, e, d, f, g, h, c, i).$$



The glueing mapping

The information on which peptides glue together is encoded in the glueing mapping

$$gl : \Sigma \to \Sigma \sqcup \Sigma^{-1},$$

where $\Sigma^{-1} = \{x^{-1} \mid x \in \Sigma\}$.

The mapping gl for TET12:

x	a	b	c	d	e	f	g	h	i
$gl(x)$	a^{-1}	f	c^{-1}	d	h	b	i	e	g

Note: If $gl(x) \in \Sigma$ then x and $gl(x)$ glue together in a parallel way. If $gl(x) \in \Sigma^{-1}$ then x and $gl(x)$ glue together in an anti-parallel way. Note that if $gl(x) \in \{x, x^{-1}\}$ then x is a homodimer.

The glueing mapping

Some notes for the mathematical audience

The mapping gl can be extended to $gl\colon \Sigma \sqcup \Sigma^{-1} \to \Sigma \sqcup \Sigma^{-1}$ by defining

$$gl(x^{-1}) = gl(x)^{-1} \quad \text{and} \quad (x^{-1})^{-1} = x.$$

Note that gl is an involution on $\Sigma \sqcup \Sigma^{-1}$.

Let $r\colon \Sigma \sqcup \Sigma^{-1} \to \Sigma \sqcup \Sigma^{-1}$ such that $r(x) = x^{-1}$. Then $\langle gl, r\rangle$ generates a group that is a subgroup of bijections on $\Sigma \sqcup \Sigma^{-1}$. The group $\langle gl, r\rangle$ acts on $\Sigma \sqcup \Sigma^{-1}$. The set $x^{\langle gl,r\rangle} = \{\alpha(x) \mid \alpha \in \langle gl, r\rangle\}$ is called the orbit of x. The elements of the orbit space

$$(\Sigma \sqcup \Sigma^{-1})/\langle gl, r\rangle = \{x^{\langle gl,r\rangle} \mid x \in \Sigma\}$$

correspond to peptide pairs.

The glueing sequence

A glueing sequence is a bijection

$$s\colon \{1, 2, \ldots, m\} \to (\Sigma \sqcup \Sigma^{-1})/\langle gl, r\rangle.$$

The mapping s is usually given as a vector with m elements from Σ which are representatives of each orbit. This mapping tells us that the peptide pair $s(1)$ glues first, followed by $s(2)$ and so on.

The question of determining the glueing sequence lies outside this model. The glueing sequence is crucial from the chemical viewpoint. If the sequence is wisely chosen then the polypeptide chain has a greater chance to form the desired polyhedron (note that in a real chemical experiment, a range of malformed byproducts may occur). Besides high folding yield, a good glueing sequence also folds rapidly on temperature quenching.

Note: Our model is general enough to also cover the DNA self-assembly.

The glueing sequence

A glueing sequence is a bijection

$$s\colon \{1, 2, \ldots, m\} \to (\Sigma \sqcup \Sigma^{-1})/\langle gl, r\rangle.$$

The mapping s is usually given as a vector with m elements from Σ which are representatives of each orbit. This mapping tells us that the peptide pair $s(1)$ glues first, followed by $s(2)$ and so on.

The question of determining the glueing sequence lies outside this model. The glueing sequence is crucial from the chemical viewpoint. If the sequence is wisely chosen then the polypeptide chain has a greater chance to form the desired polyhedron (note that in a real chemical experiment, a range of malformed byproducts may occur). Besides high folding yield, a good glueing sequence also folds rapidly on temperature quenching.

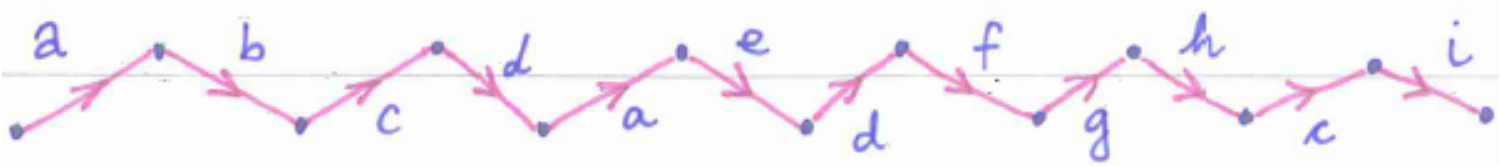
Note: Our model is general enough to also cover the DNA self-assembly.

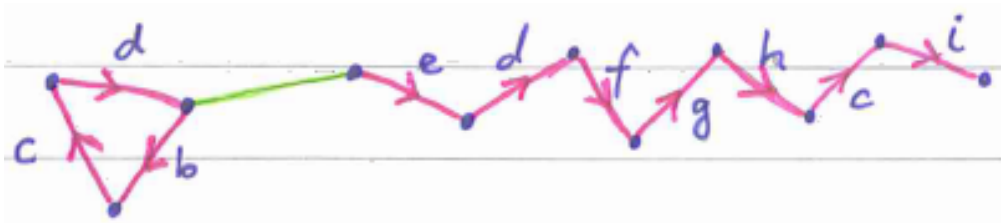
The glueing sequence

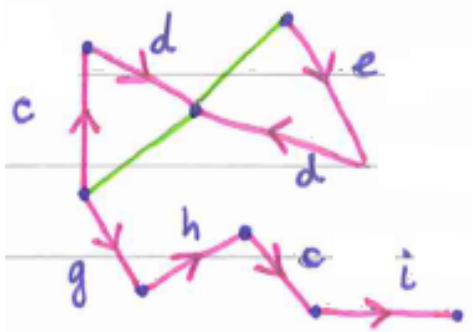
We know that the pair $(\mathrm{APH}, \mathrm{APH})$ is the most stable and will be the first to glue. It is followed respectively by $(\mathrm{P3}, \mathrm{P4})$, $(\mathrm{P5}, \mathrm{P6})$, $(\mathrm{BCR}, \mathrm{BCR})$, $(\mathrm{P7}, \mathrm{P8})$ and finally $(\mathrm{GCN}_{\mathsf{sh}}, \mathrm{GCN}_{\mathsf{sh}})$. In our abstract model we use letters $a, b, c, \ldots$ to represent peptides. The glueing sequence of $\mathrm{TET12}$ is

$$(a, b, g, c, e, d).$$

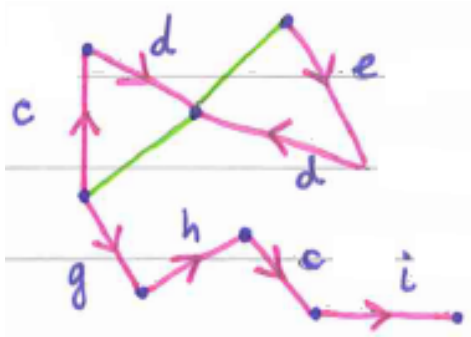
An example: TET12

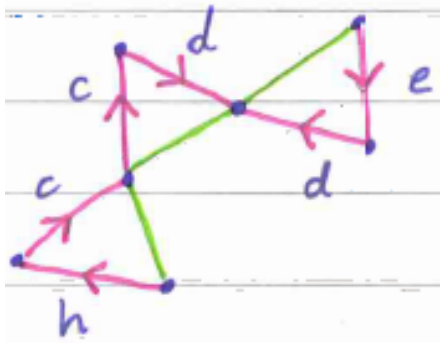


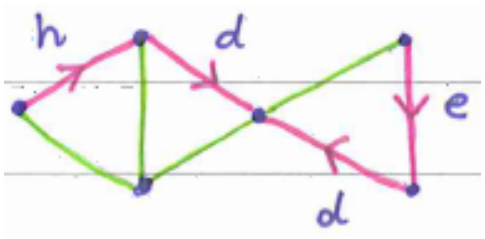




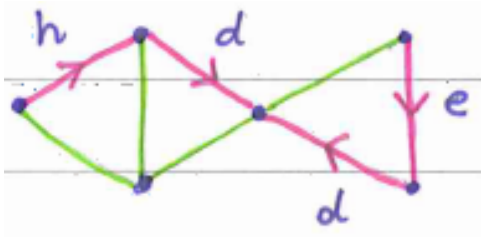
An example: TET12

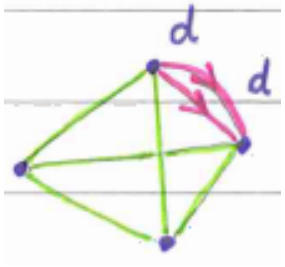


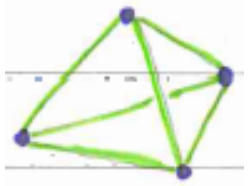




An example: TET12

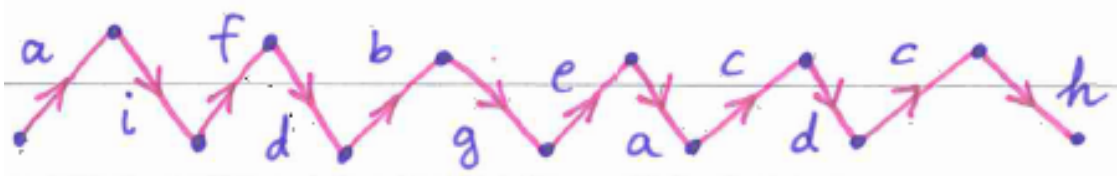




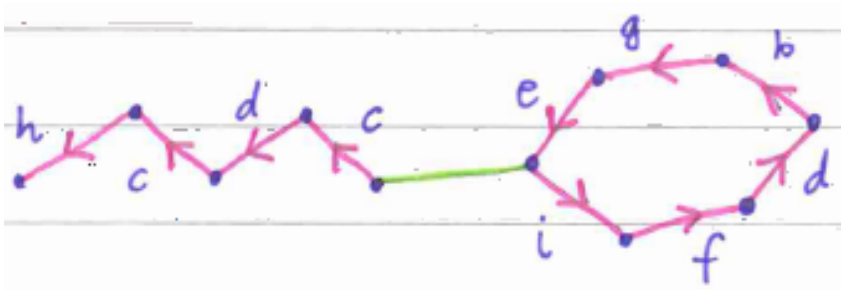


Another example: Some random labeling of $\vec{P}_{13}$

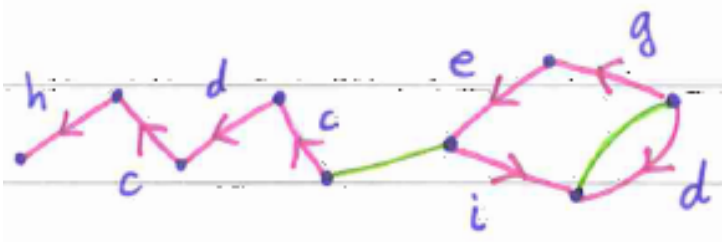
Note that the labeling of the path $\vec{P}_{13}$ cannot be chosen arbitrarily. The path below does not result in the tetrahedron even though its labels are a permutation of the original ones:

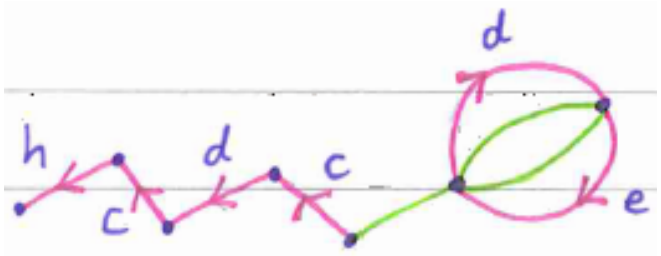


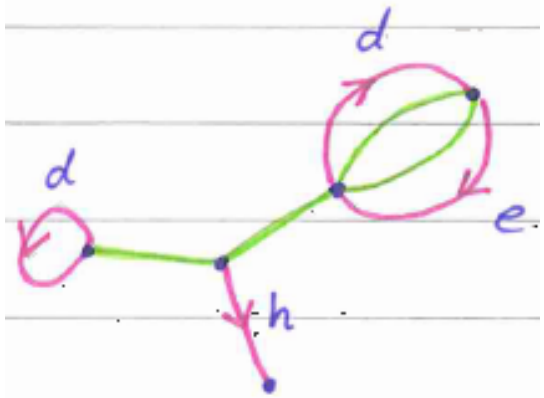
Take the same glueing mapping and glueing sequence as in the glueing process of the $\mathrm{TET}12$.



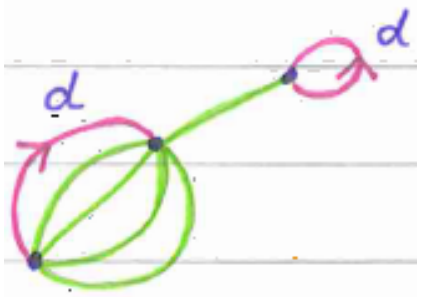
Another example: Some random labeling of $\vec{P}_{13}$

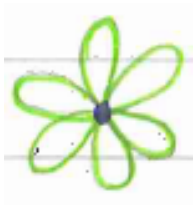






Another example: Some random labeling of $\vec{P}_{13}$





The end result in this case is the bouquet graph with 6 loops.

Conclusion: It is not trivial to find a labeled path that will result in the desired polyhedron.

The first mathematical model

The first mathematical model for this problem, which was developed with trivalent polyhedra in mind, was described by Klavžar and Rus. A year later, the model was refurbished by Fijavž, Pisanski and Rus to include all polyhedra.

Definition

A double trace in a graph G is a walk which traverses every edge exactly twice.

A double trace in a simple graph can be given as a sequence of vertices

$$W = w_0\, w_1\, w_2 \ldots w_{2m},$$

where indices are taken modulo $2m$.

The first mathematical model

It is easy to see that every graph G admits a double trace. If we replace every edge of G by a digon, we obtain an Eulerian multigraph G'. An Eulerian circuit in G' corresponds to a double trace in G. Because all vertices in G' are of even degree the following proposition follows:

Proposition

Every graph G has a double trace.

Warning: A double trace in a graph G does not in general give rise to a directed path (which represents a polypeptide) that will result in the graph G after the glueing process is performed.

The first mathematical model

It is easy to see that every graph G admits a double trace. If we replace every edge of G by a digon, we obtain an Eulerian multigraph G'. An Eulerian circuit in G' corresponds to a double trace in G. Because all vertices in G' are of even degree the following proposition follows:

Proposition

Every graph G has a double trace.

Warning: A double trace in a graph G does not in general give rise to a directed path (which represents a polypeptide) that will result in the graph G after the glueing process is performed.

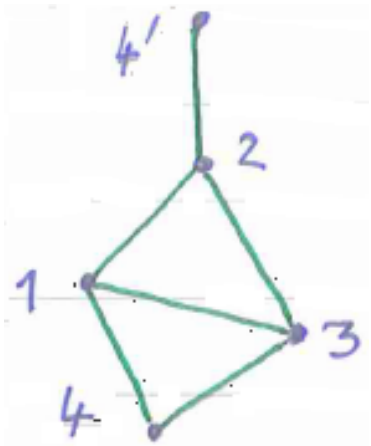
The first mathematical model

The skeleton of the tetrahedron is the graph K_4. Let $V(K_4) = \{1, 2, 3, 4\}$. The walk

$$W = 1\ 2\ 4\ 2\ 3\ 1\ 4\ 3\ 1\ 2\ 3\ 4\ 1$$

is a double trace in K_4. *How to obtain the polypeptide chain?*

The result of the glueing process on the polypeptide obtained from the above double trace:



The first mathematical model

Definition

Let $e = uv \in E(G)$ and let $W = w_0\, w_1\, w_2 \ldots w_{2m}$ be a double trace in G. If there exists an integer i such that $(w_{i-1}, w_i, w_{i+1}) = (v, u, v)$ then W has a retracing.

Let $u \in V(G)$ and let $v, v' \in G(u)$ such that $v \neq v'$. If there exist integers i and j, $i \neq j$, such that $w_i = w_j = u$ and $\{w_{i-1}, w_{i+1}\} = \{w_{j-1}, w_{j+1}\} = \{v, v'\}$ then W has a repetition through vertex u.

Klavžar and Rus defined the notion of a stable trace:

Definition

A stable trace is a double trace that has no retracing and no repetition through its vertices.

The first mathematical model

A stable trace of a graph G gives rise to a polypeptide that results in the graph G when the glueing process is performed.

Note: The double trace from previous example has both a retracing and a repetition. In both cases the vertex labeled with 4 is involved.

Rus and Klavžar proved the following theorem:

Theorem (Klavžar and Rus)

A graph G admits a stable trace if and only if $\delta(G) \geq 3$.

The second mathematical model

Fijavž, Pisanski and Rus generalised the notion of a stable trace:

Definition

Let G be a graph, $v \in V(G)$ and $N \subseteq G(v)$. Let W be a double trace in G. Then W has a N-repetition at v if for all integers i such that $w_i = v$ it holds that the pair $\{w_{i-1}, w_{i+1}\}$ is either contained in N or disjoint from N.

Definition

A n-stable trace is a double trace where for all $v \in V(G)$ and for all $N \subseteq G(v)$ such that $1 \leq |N| \leq n$ it holds that W has no N-repetition.

The second mathematical model

They defined the important notion of a strong trace:

Definition

Let G be a graph and let W be a double trace of G. If for every vertex $v \in V(G)$ it holds that W has a N-repetition at v if and only if $N = G(v)$ or $N = \emptyset$ then W is called a strong trace.

They also proved the following theorem which is based on a deep result from topological graph theory:

Theorem (Fijavž, Pisanski and Rus)

Every graph admits a strong trace.

The second mathematical model

Parallel and antiparallel traces

Let W be a double trace of a graph G. The double trace W traverses each edge $e \in E(G)$ exactly two times. If W traverses an edge e in the same direction both times then e is parallel with respect to W. Otherwise, it is antiparallel with respect to W.

If all edges of G are parallel with respect to W then W itself is called a parallel trace. Similarly, if all edges of G are antiparallel with respect to W then W is called an antiparallel trace. It is easy to see that a parallel / antiparallel edge of W gives rise to a parallel / antiparallel dimer in the self-assembled polypeptide.

Obtaining new traces from the old ones

It is possible to obtain a new double trace from existing one. We can change the direction of tracing (reverse the trace W) or start at a different vertex (shift the trace W). If the graph G posesses a symmetry, we can obtain a new trace by acting by a graph automorphism $\alpha \in \mathrm{Aut}(G)$ on W.

Definition

Double traces W and W' are equivalent if W' can be obtained from W by using any combination of the following operations:

- reversion of W;
- shifting W;
- applying a permutation on W induced by an automorphism of G.

Otherwise, traces W and W' are called non-equivalent.

Obtaining new traces from the old ones

Canonical traces

Let $\mathcal{T}(G)$ denote the set of all double traces of a graph G. The equivalence of double traces is an equivalence relation on the set $\mathcal{T}(G)$. This is clearly also an equivalence relation on any subset of $\mathcal{T}(G)$, such as stable traces, strong traces and so on. Assume that vertices $V(G) = \{v_0, \ldots, v_{n-1}\}$ of the graph G are linearly ordered as

$$v_0 < v_1 < \cdots < v_{n-1}.$$

This ordering induces a lexicographic ordering on the set of double traces of G, i.e., $W \leq W'$ if and only if $W = W'$ or there exists an index i, $0 \leq i \leq 2m$, such that $w_i < w'_i$ and $w_j = w'_j$ for all $j < i$. Every subset $\mathcal{S} \subseteq \mathcal{T}(G)$ has a lexicographically smallest member which is called the canonical representative of $\mathcal{S}$.

Obtaining new traces from the old ones

Let us define mappings $\rho, \sigma_i \colon \mathcal{T}(G) \to \mathcal{T}(G)$ as

$$\rho(w_0 \dots w_{2m}) = w_{2m} \dots w_0 \qquad \text{and} \qquad \sigma_i(w_0 \dots w_{2m}) = w_i \dots w_{2m+i}.$$

Note: $\sigma_0 = \sigma_{2m} = \mathrm{id}_{\mathcal{T}(G)}$.
Let $\alpha \in \mathrm{Aut}(G)$. The automorphism α acts on $\mathcal{T}(G)$ in the following way:

$$\alpha(w_0 \dots w_{2m}) = \alpha(w_0) \dots \alpha(w_{2m}).$$

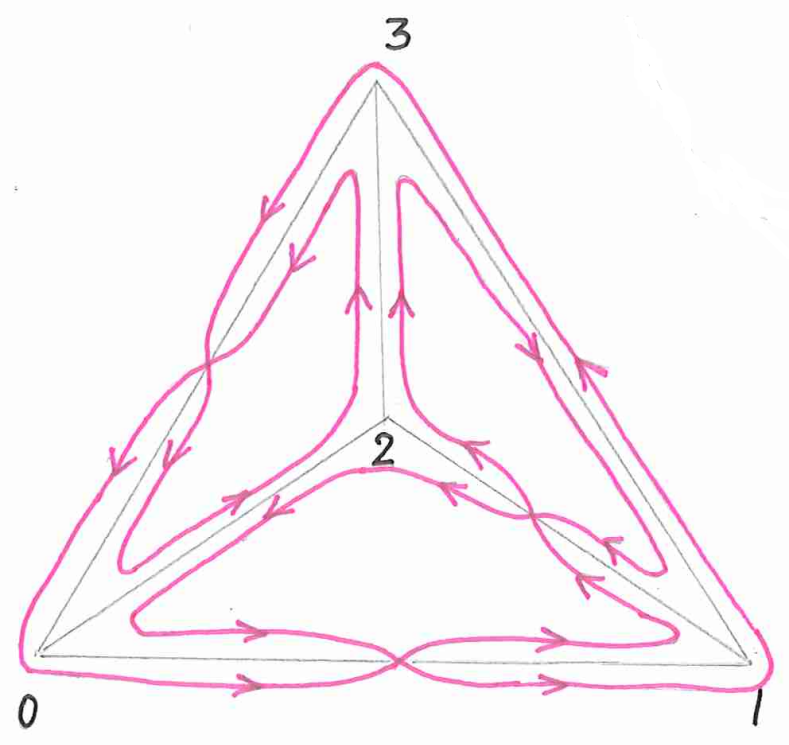
Then $\mathrm{Aut}(G)$, $R = \{\mathrm{id}, \rho\}$ and $S = \{\sigma_i \mid i = 0, \dots, 2m-1\}$ are three groups acting on $\mathcal{T}(G)$. Elements of the orbit space

$$\mathcal{T}(G)/(\mathrm{Aut}(G) \times R \times S)$$

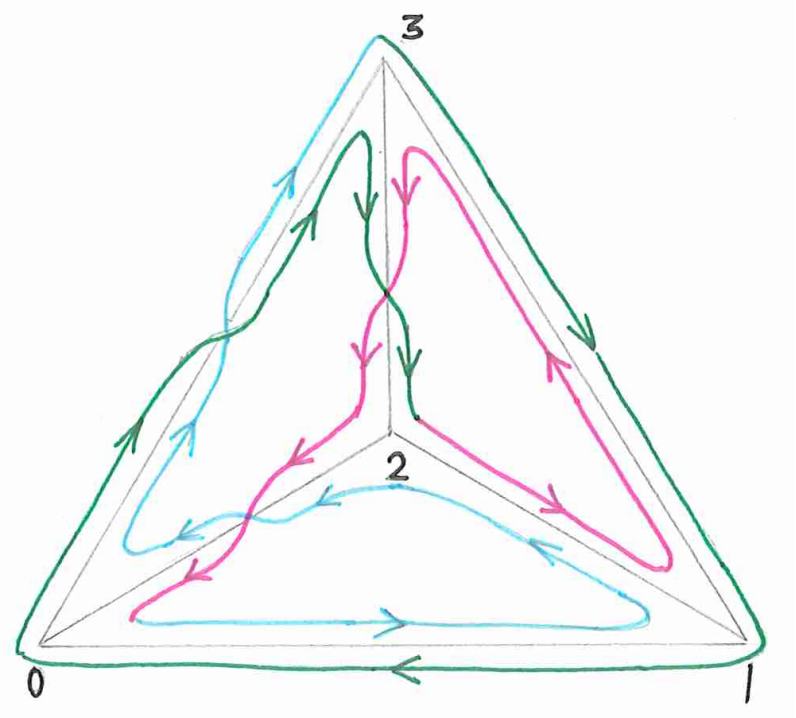
are precisely equivalence classes of double traces for the equivalence relation.

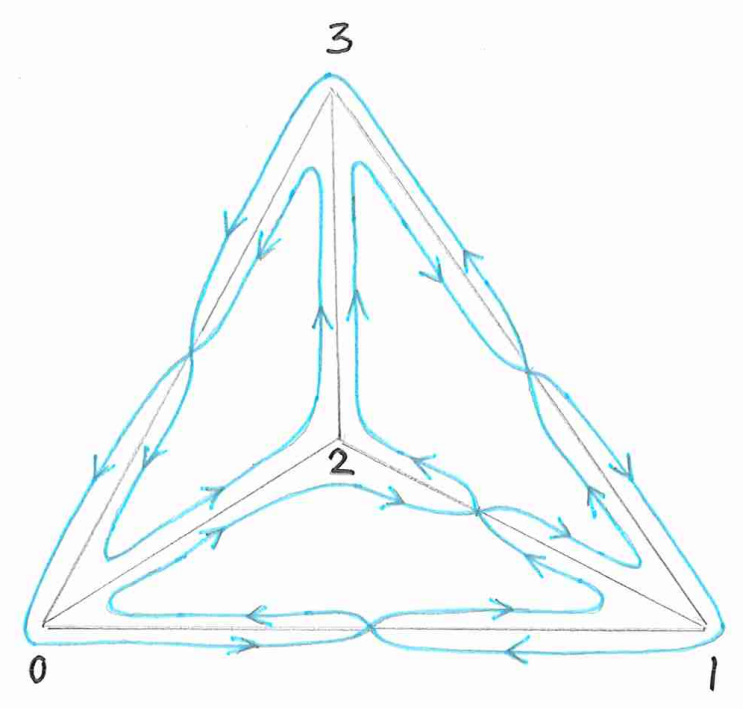
The tetrahedron has 3 non-equivalent strong traces

The group $\mathrm{Aut}(G) \times R \times S$ partitions the 672 different strong traces of the tetrahedron graph into 3 equivalence classes of sizes 288, 288 and 96:



The tetrahedron has 3 non-equivalent strong traces

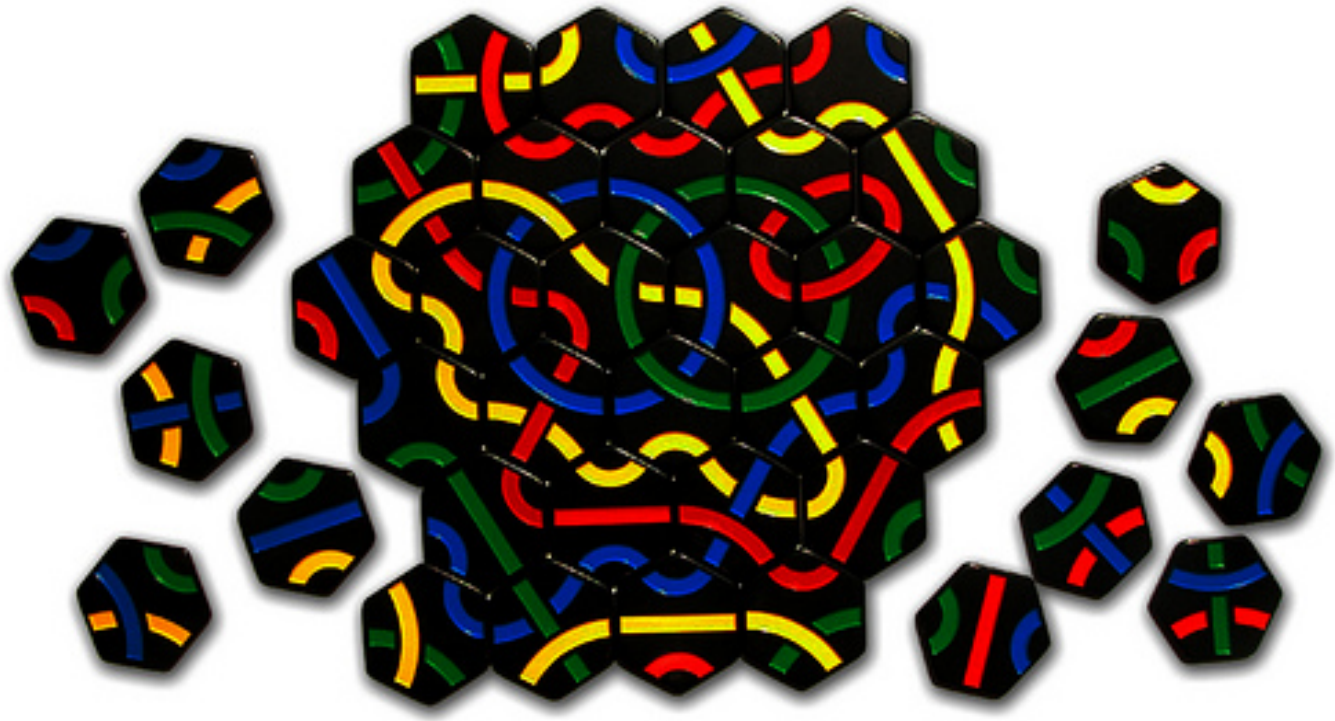




Enumeration of strong traces

There is a branch-and-bound algorithm that outputs each canonical strong trace of G.

There is another approach using dynamic programming.



Dynamic programming approach

We may view the strong traces as walks traversing all flags of a map M which is a combinatorial representation of a cellular embedding of the graph G. The trace can be therefore interpreted as a 2-regular graph whose vertices are flags of M. This motivates us to define the following:

Definition

Let $M = (\Phi, \tau_0, \tau_1, \tau_2)$ be a map. Choose a mapping $\lambda\colon E(G) \to \{-1, 1\}$. An (undirected) map trace, denoted Q_λ, is a 2-regular graph whose set of vertices is Φ and its edges are defined as follows:

- $\phi \sim \tau_1(\phi)$;
- $\phi \sim \tau_0(\phi)$ when $\lambda(e_\phi) = 1$ and $\phi \sim \tau_2\tau_0(\phi)$ when $\lambda(e_\phi) = -1$.

Dynamic programming approach

Map trace Q_λ in general consists of one or more even cycles. We are only interested in those traces that are connected:

Definition

A connected (undirected) map trace Q_λ is a map trace which has a single connected component.

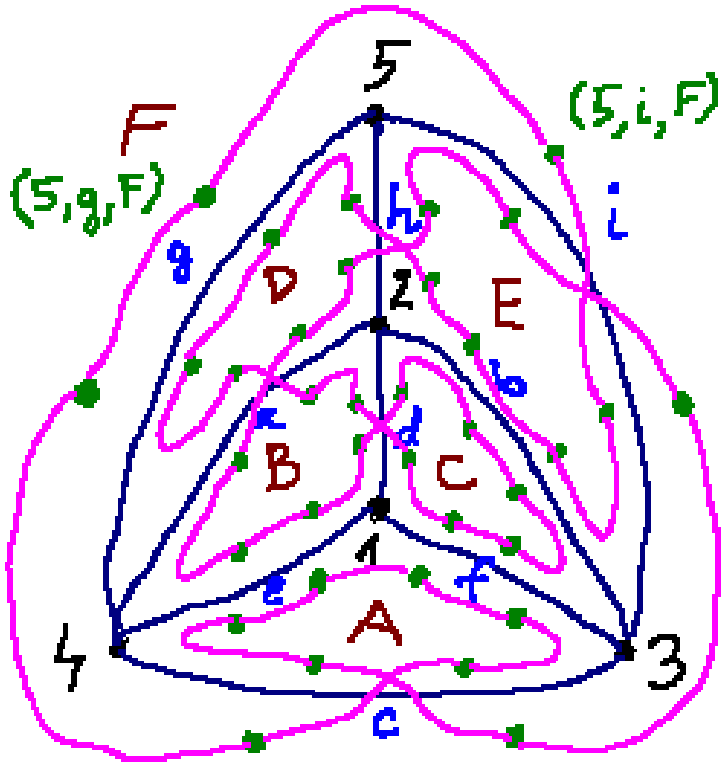
If we choose an initial flag ϕ_0 and a direction, we can assign a strong trace to a connected map trace Q_λ. Traverse the map trace Q_λ in the given direction starting at ϕ_0 to obtain the closed walk $(\phi_0, \phi_1, \phi_2, \ldots, \phi_{4m} = \phi_0)$. Then

$$W(Q_\lambda) = (v_{\phi_0}, v_{\phi_2}, v_{\phi_4}, \ldots, v_{\phi_{4m}})$$

is a strong trace. Most of the time, we are interested in non-equivalent strong traces. Because by shifting and reversing a strong trace W we obtain an equivalent strong trace, the intial flag ϕ_0 and the direction on the connected map trace Q_λ can be arbitrarily chosen.

Dynamic programming approach

Embedding of the bipyramid in the plane and one of its connected map traces:



The magenta line represents the map trace.

Not every strong trace is compatible with every map

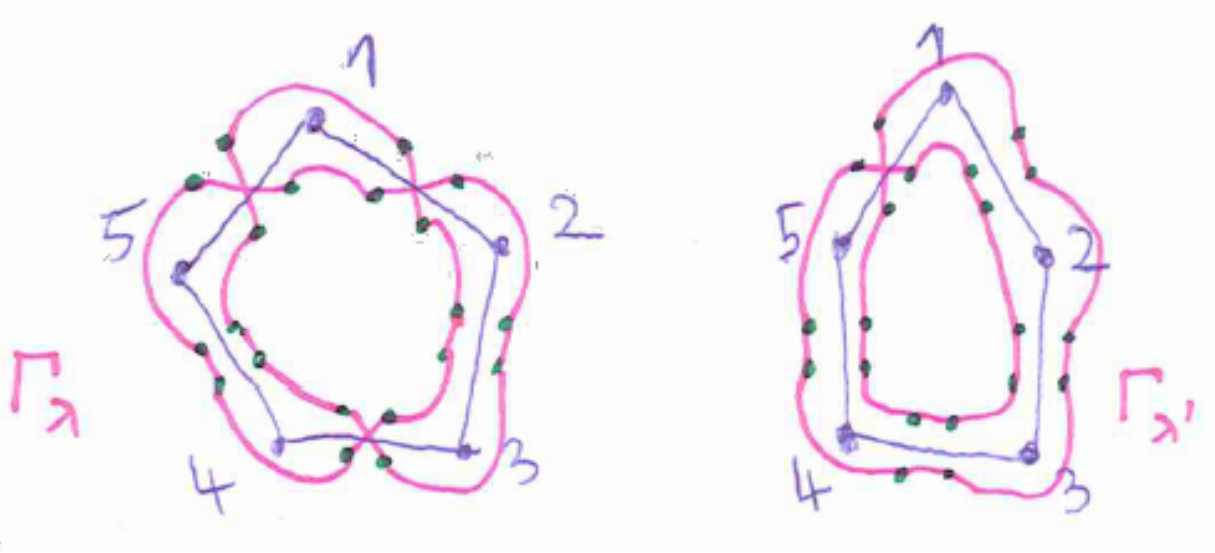
Definition

Map M is *compatible* with a strong trace W if there exists a map trace Q such that $W = W(Q)$.

Not every strong trace is compatible with every map. For example, if a strong trace of the bipyramid from the previous slide contains the subsequence $(1, 2, 5)$ then it is obviously not compatible with the map.

Connection between map traces and strong traces

Different map traces may yield the same strong trace as in the case of the cycle graph C_5 embedded in the plane:



If minimum degree in the graph is at least 3, the map trace is either:

- uniquely determined from the strong trace or
- it is not compatible with the given map M.

Note: For most applications it is sufficient to find strong traces that are compatible with a fixed embedding of the graph G.

Point of departure: Brute-force

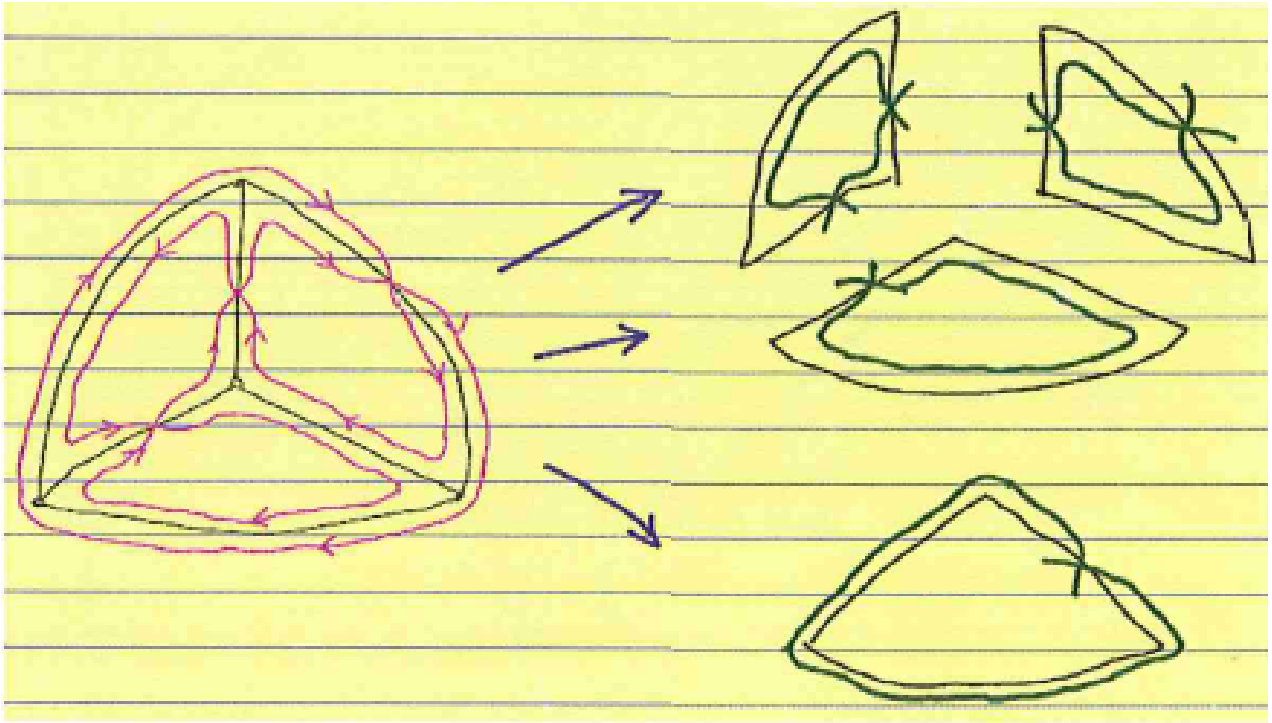
We can immediately obtain a simple brute-force algorithm with the time complexity that is exponential in the number of edges of the skeleton graph. The algorithm enumerates all strong traces that are compatible with a given map M:
Let M be a map with $4m$ flags and let $G = Skel(M)$. Let $L = \emptyset$.
Iterate over all subsets $\mathcal{E} \subseteq E(G)$. For a given set $\mathcal{E}$ define the mapping $\lambda_{\mathcal{E}} \colon E(G) \to \{-1, 1\}$, such that

$$\lambda_{\mathcal{E}}(e) = \begin{cases} 1, & e \in \mathcal{E}; \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

Check whether $Q_{\lambda_{\mathcal{E}}}$ is a connected map trace. If it is connected, take $W = W(Q_{\lambda_{\mathcal{E}}})$. Then determine the canonical strong trace W' that is equivalent to W and add W' to the set L.

Main idea

We cut the map (that has a strong trace drawn on it) along its edges. From the tetrahedron we obtain four 'jigsaw puzzle pieces':



The strong trace that was drawn on the map can be obtained from these jigsaw pieces by glueing them back together. This is the main idea behind the dynamic programming algorithm.

Signature of a partial map trace

By choosing a proper subset of faces of the map, we obtain a surface with boundary. This boundary is a disjoint union of cycles:

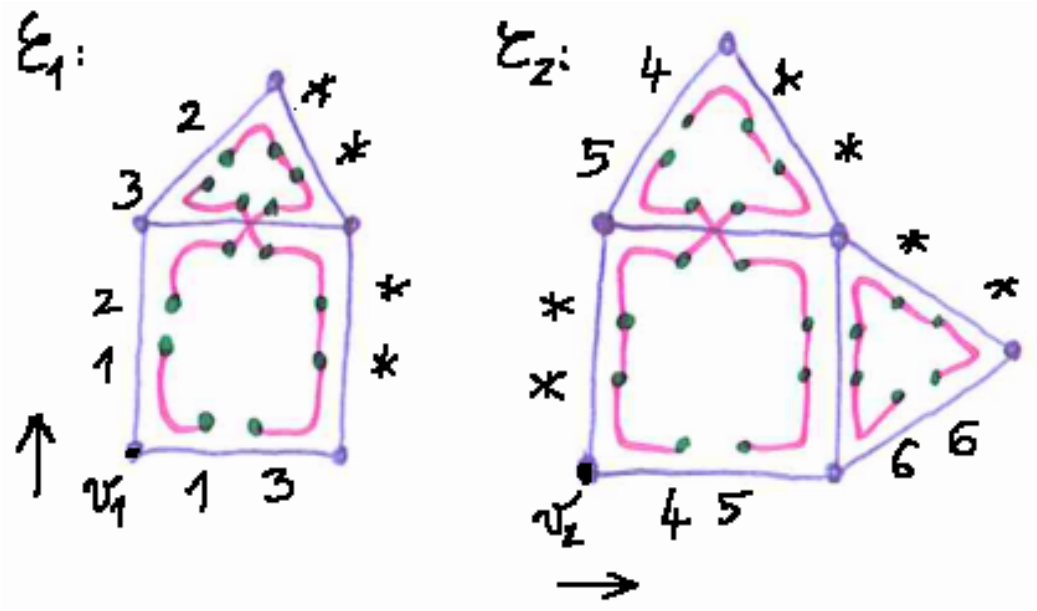
$$\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r.$$

On each of those cycles we choose a reference vertex $v_i \in \mathcal{C}_i$ and a direction. The signature, denoted by $\sigma(\widetilde{Q})$, of a partial map trace $\widetilde{Q}$ can be obtained in the following way:
Traverse, the cycles $\mathcal{C}_1, \mathcal{C}_2, \ldots$ of the boundary: start at the reference vertex and travel in the chosen direction. For each flag ϕ that you encounter, write down one symbol:

- If the degree of vertex ϕ in the partial map trace is 2, write down the symbol $\star$.
- If the degree of vertex ϕ is 1, find the other end of this path. If it was already labeled, write down its label. If it was not labeled yet, label it using the smallest positive integer that was not used so far.

Signature of a partial map trace

An example



Here 5 faces were chosen from a larger map. The boundary consists of two cycles denoted $\mathcal{C}_1$ and $\mathcal{C}_2$. Reference vertices and directions are also indicated. The signature of this partial map trace is

$$(1, 2, 3, 2, \star, \star, \star, \star, 3, 1, 4, 5, 6, 6, \star, \star, \star, \star, 4, 5, \star, \star).$$

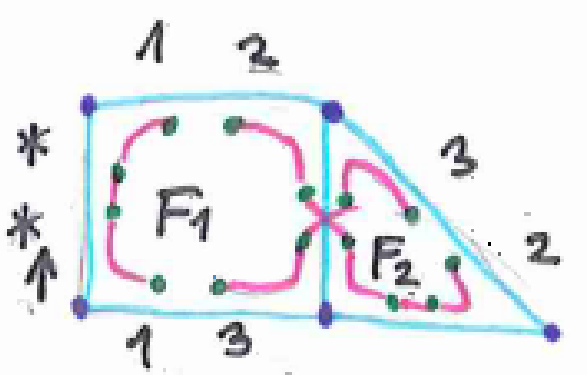
Merging partial map traces

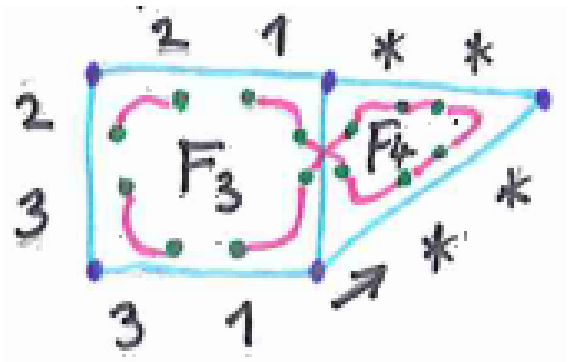
Partial map traces $\widetilde{Q}$ and $\widetilde{Q}'$ are compatible if for each edge $e \in E(M)$ that is shared between the boundaries of $\mathcal{F}$ and $\mathcal{F}'$ it holds that all vertices $\{\phi \mid e_\phi = e\}$ of the disjoint union of partial map traces $\widetilde{Q}$ and $\widetilde{Q}'$ have the same degree (which is either 1 or 2).

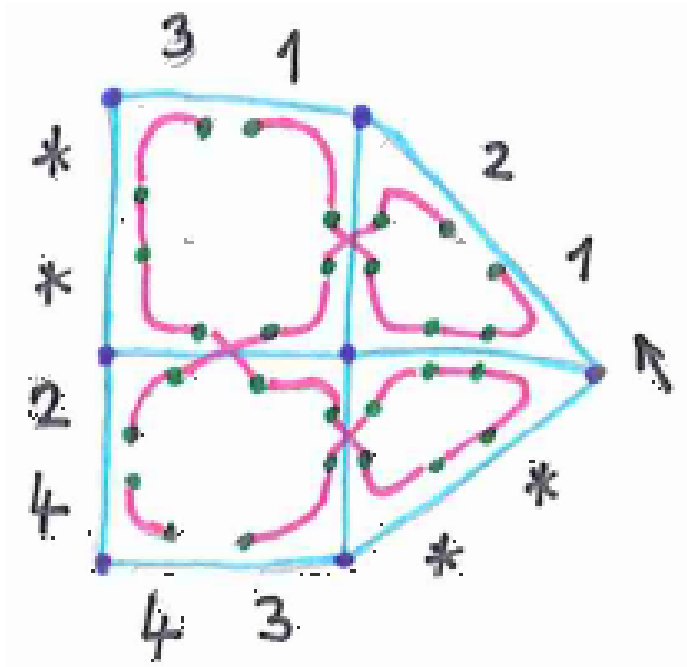
We can merge partial map traces $\widetilde{Q}$ and $\widetilde{Q}'$ if they are compatible. While there exists an edge $e \in E(M)$ that is shared between $\mathcal{F}$ and $\mathcal{F}'$ and a degree-1 vertex ϕ such that $e_\phi = e$, connect the vertices ϕ and $\tau_2\tau_0(\phi)$. In this way we obtain the merged partial map trace $\widetilde{Q} \circ \widetilde{Q}'$.

Merging partial map traces

An example







Suppose that faces F_1 and F_3 as well as faces F_2 and F_4 are adjacent.

Key observation

In order to obtain $\sigma(\widetilde{Q} \circ \widetilde{Q}')$, we only need $\sigma(\widetilde{Q})$ and $\sigma(\widetilde{Q}')$.

The dynamic programming algorithm has three phases which we call the preparation phase, the main phase and the final phase.

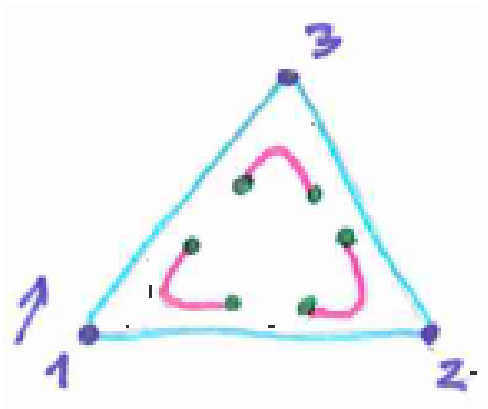
Key observation

In order to obtain $\sigma(\widetilde{Q} \circ \widetilde{Q}')$, we only need $\sigma(\widetilde{Q})$ and $\sigma(\widetilde{Q}')$.

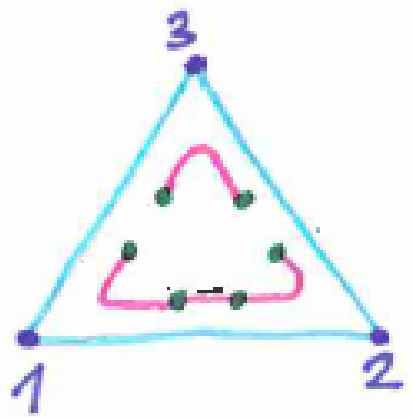
The dynamic programming algorithm has three phases which we call the preparation phase, the main phase and the final phase.

The preparation phase

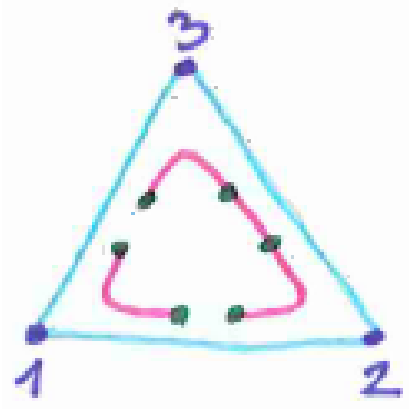
In the preparation phase of the algorithm, we prepare a list of signatures of all admissible partial map traces of each face. A triangular face has 7 admissible partial map traces and 1 non-admissible partial map trace:



$(1, 2, 2, 3, 3, 1)$

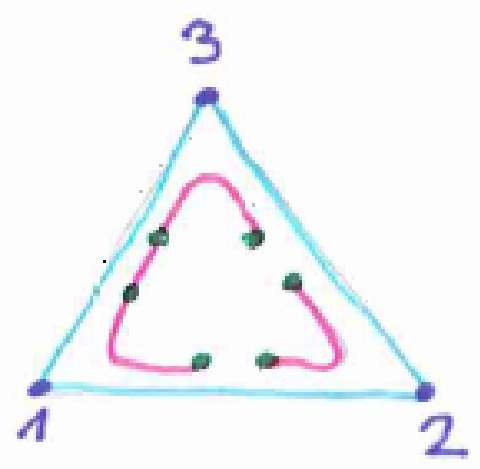


$(1, 2, 2, 1, \star, \star)$

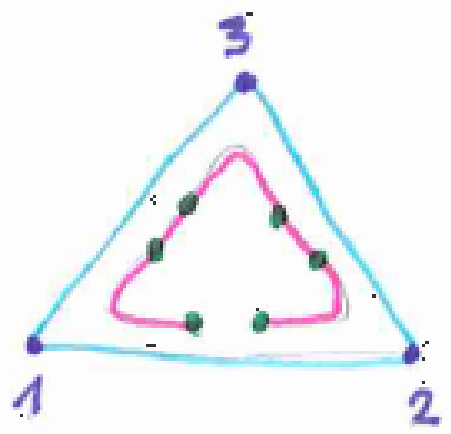


$(1, 2, \star, \star, 2, 1)$

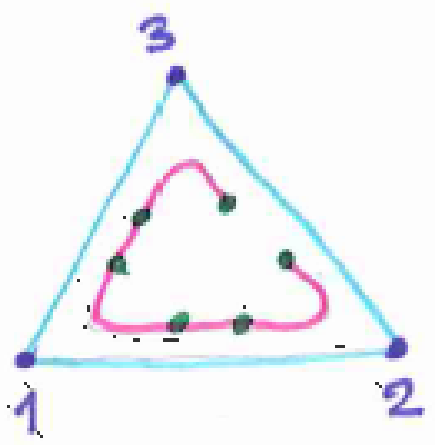
The preparation phase cont'd



$(\star, \star, 1, 2, 2, 1)$

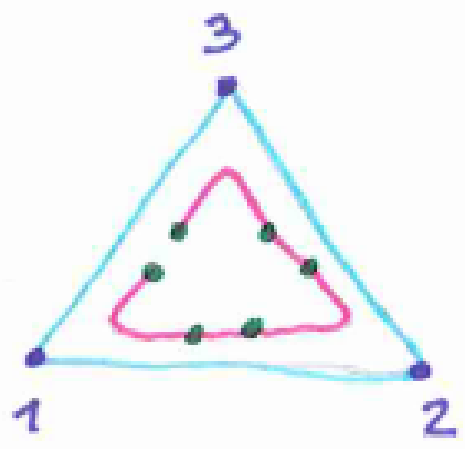


$(\star, \star, \star, \star, 1, 1)$



$(\star, \star, 1, 1, \star, \star)$

The preparation phase cont'd



$(1, 1, \star, \star, \star, \star)$

In the preparation phase we also choose a linear ordering $f_1 < f_2 < \cdots < f_{|F(M)|}$ on the faces of $F(M)$.

The main phase

Begin with a single face f_1 and iteratively add new faces to it, one by one.

$$\mathcal{F}_i = \{f_1, \ldots, f_i\}$$

For each $\mathcal{F}_i$, choose an ordering on the cycles that comprise the boundary of $\mathcal{F}_i$ and choose a reference vertex (and a direction) on each of the cycles. Those reference vertices and directions are common to all signatures of $\mathcal{F}_i$.

For each $i = 1, 2, \ldots, |F(M)| - 1$ create a dictionary $\mathcal{D}_i$:

- keys are signatures of all admissible partial map traces of $\mathcal{F}_i$.
- value that corresponds to a signature is the number of all partial map traces with the given signature.

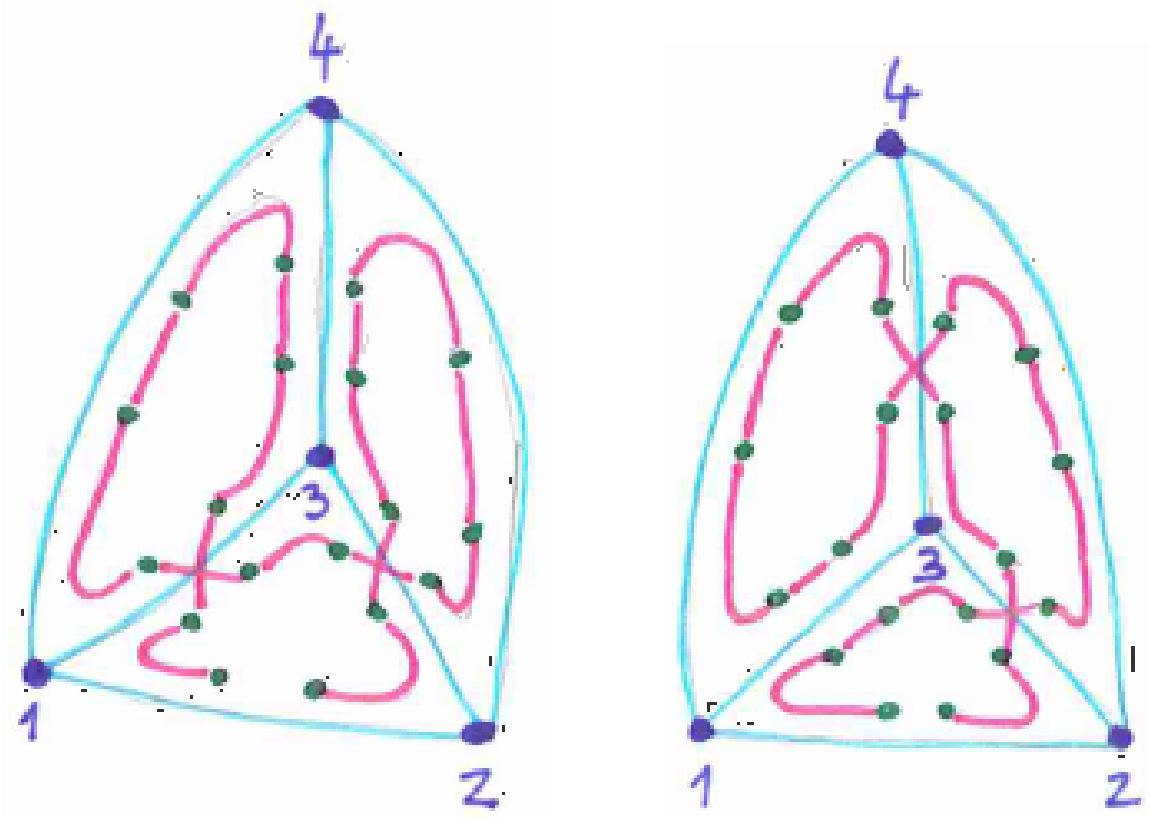
The initial dictionary $\mathcal{D}_1$ contains admissible signatures of the face f_1, values in this dictionary are equal to 1.

The main phase cont'd

The dictionary $\mathcal{D}_{i+1}$ can be obtain from the dictionary $\mathcal{D}_i$ and the list of admissible signatures of the face f_{i+1}. Combine each signature in $\mathcal{D}_i$ with each signature of f_{i+1}. If they are compatible and the resulting signature is admissible, increase the count of the resulting admissible signature in $\mathcal{D}_{i+1}$ by the corresponding value stored in $\mathcal{D}_i$.

The key observation

The same signature can be obtained in many different ways. The signature $(1, 1, \star, \star, \star, \star)$ from dictionary $\mathcal{D}_3$ belongs to more than one partial map trace:



The final phase

- In the final phase build the dictionary $\mathcal{D}_{|F(M)|}$. When the last face $f_{|M(G)|}$ is added, the boundary disappears.
- The number of different connected map trace is obtained.

Notes:

- It is possible to reconstruct the i-th map trace from the data structures that were used in the algorithm.
- The symmetries of the map M were not taken into account here.

Thanks to the organisers for this great event!