Algorithmic Computation of Arrangements of Lines

David Munkacsi

Leibniz Universität Hannover

Tagung der Fachgruppe Computeralgebra
2022

Introduction

In his 2019 talk[1] in Hannover, Masahiko Yoshinaga asked the question if there are arrangements of lines in the real projective plane $\mathbb{RP}^2$ which we can color in a way that each line is either red or blue and at the intersection points we either have lines of only one color, or an even number of lines of each color.

We call such an arrangement a *Mod 2–net*.

He also presented the only known example over $\mathbb{R}$, the *icosidodecahedral arrangement*.

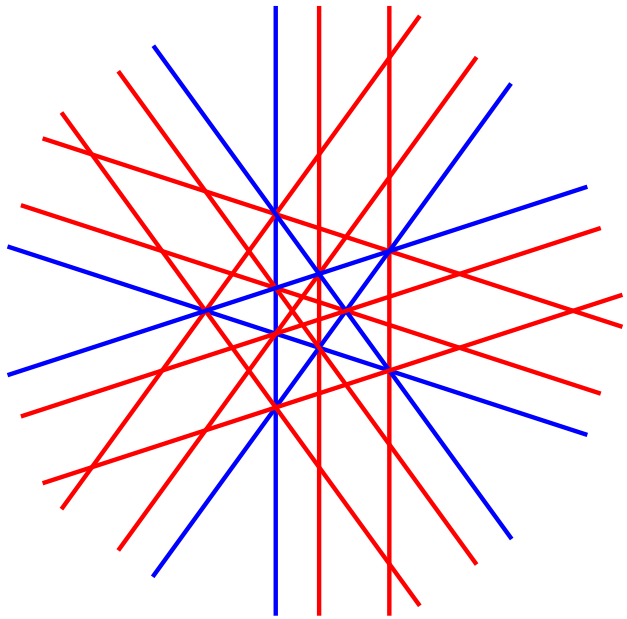


Image credit: Masahiko Yoshinaga, 2019

[1] M. Yoshinaga. "Icosidodecahedron and Milnor fiber of arrangements". Talk at the Workshop on Hyperplane Arrangements and Reflection Groups, Leibniz Universität Hannover. 2019.

Preliminaries

Let V be an affine or projective space over a field $\mathbb{K}$.

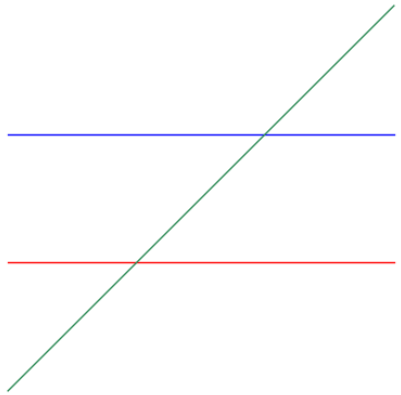
A *hyperplane* is an affine or projective subspace of V with codimension 1.

An *arrangement (of hyperplanes)* $\mathcal{A}$ is a finite set of hyperplanes in V.

We call an arrangement $\mathcal{A}$ *central* if $\bigcap_{H \in \mathcal{A}} H \neq \emptyset$.

We call an arrangement *essential* if it contains $\dim(V)$ linearly independent hyperplanes.

There is a correspondence between central essential arrangements in $\mathbb{R}^3$ and essential arrangements in $\mathbb{RP}^2$.



Let

$$L(\mathcal{A}) = \left\{ \bigcap I \;\middle|\; I \subset \mathcal{A},\ \bigcap I \neq \emptyset \right\}.$$

By convention, $\bigcap_{\emptyset} = V$.

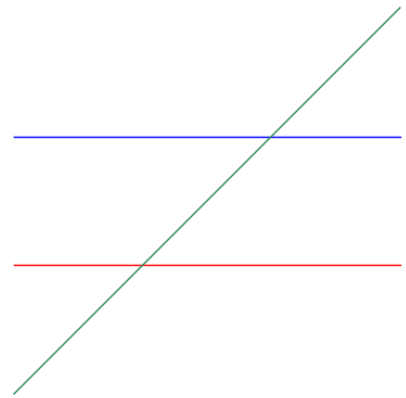
Define a partial order on $L(\mathcal{A})$ by

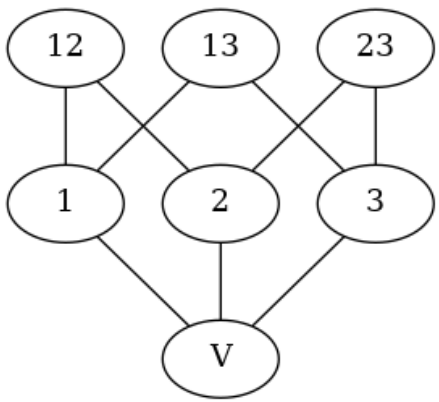
$$X \leq Y \Leftrightarrow X \supseteq Y.$$

Note that this is a reverse inclusion.

With this, $L(\mathcal{A})$ is a poset, the *intersection poset* of $\mathcal{A}$.

We call a property *combinatorial* if it has the same value for arrangements with isomorphic intersection posets.





Strong Mod 2–nets

Let $\mathcal{A}$ be an arrangement of n hyperplanes in $\mathbb{KP}^2$.
We say that $\mathcal{A}$ satisfies the *strong Mod 2–net property* if we can color each of its lines with one of the colors red and blue, such that the following conditions are satisfied:

1. For each point, and for both colors, the number of lines of a given color going through the point is even.
2. There is at least one red and one blue line.
3. There are at least two distinct intersection points.

We call the property *strong*, because originally for a Mod 2–net, points where an uneven number of lines intersect were allowed in the case that all those lines are of the same color.

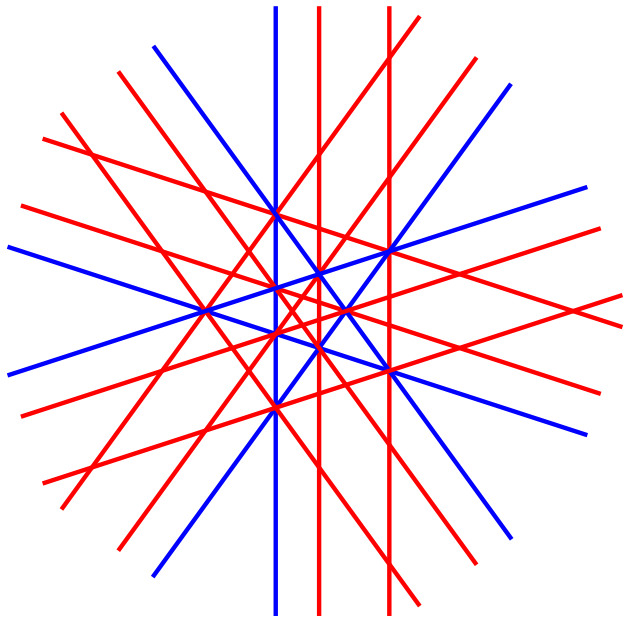


Image credit: Masahiko Yoshinaga, 2019

Conjecture

An arrangement satisfies the strong Mod 2–net property if and only if it is a Mod 2–net.

Motivation

For each $H \in \mathcal{A}$, let α_H be a linear form for which $H = ker(\alpha_H)$. The *defining polynomial* of $\mathcal{A}$ is

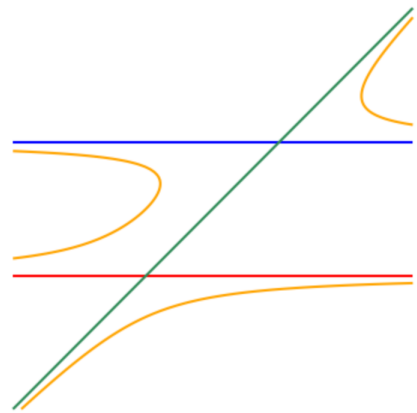
$$Q = Q(\mathcal{A}) = \prod_{H \in \mathcal{A}} \alpha_H.$$

We define the *Milnor fiber* of an arrangement $\mathcal{A}$ in V as

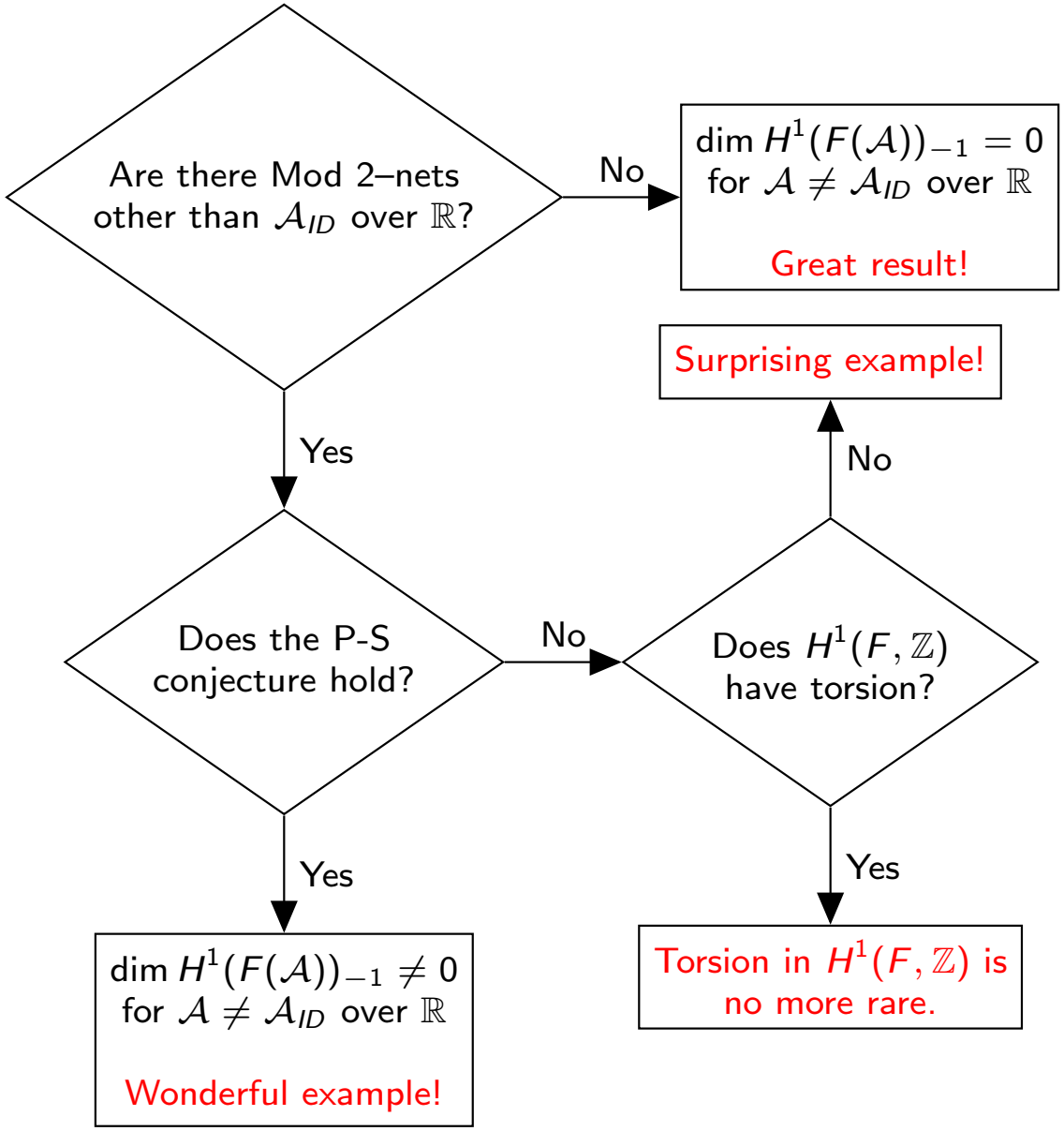
$$F = F(\mathcal{A}) = \{x \in V \mid Q(x) = 1\}.$$

The topology of the Milnor fiber over $\mathbb{C}$ is an important area of research.

Real arrangements with the (strong) Mod 2–net property can deliver counterexamples to a conjecture of Papadima and Suciu[2] about the first cohomology of the Milnor fiber.



[2] Conjecture 1.9 in S. Papadima and A. I. Suciu. "The Milnor fibration of a hyperplane arrangement: from modular resonance to algebraic monodromy". In: Proceedings of the London Mathematical Society 114.6 (2017)

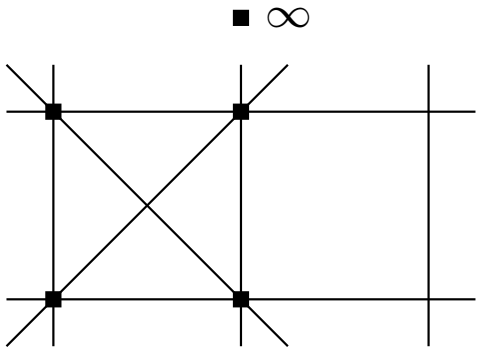


$\mathcal{A}_{ID}$ is the icosidodecahedral arrangement.

Yoshinaga: $\mathcal{A}$ has Mod 2–net structure iff $H^1(A^\bullet_{\mathbb{F}_2}(\mathrm{d}\mathcal{A}), w_0) \neq 0$.

P-S conjecture (the relevant part): Let $\mathcal{A}$ be an arrangement of rank ≥ 3. Then: $\dim H^1(F(\mathcal{A}))_{-1} = \dim_{\mathbb{F}_2} H^1(A^\bullet_{\mathbb{F}_2}(\mathrm{d}\mathcal{A}), w_0)$.

Algorithmic approach



Basis: Greedy algorithm of Michael Cuntz[3].

Main idea: Search for strong Mod 2–nets over a finite field $\mathbb{F}_q$, see if they are realizable over $\mathbb{R}$ or $\mathbb{C}$, that is, if there is an arrangement over these fields with the same intersection poset.

Algorithm

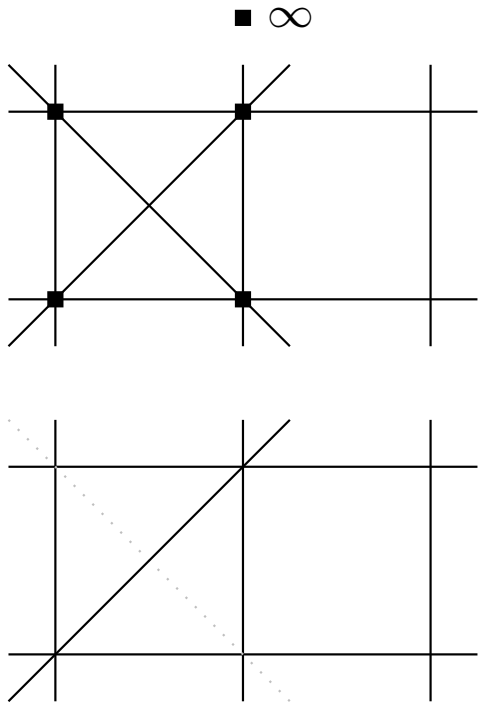
- Start with a random arrangement of n lines in $\mathbb{F}_q\mathbb{P}^2$
- Until we have found a Mod 2–net:
  - Replace a line with a line through 2 randomly chosen intersection points
  - If the new arrangement is better, keep it, otherwise revert to the old state
  - If for a long time, no improvement has been found, start anew with a random arrangement

[3] M. Cuntz. "A greedy algorithm to compute arrangements of lines in the projective plane". In: Discrete & Computational Geometry. (2021)

Algorithmic approach

Basis: Greedy algorithm of Michael Cuntz[3].

Main idea: Search for strong Mod 2–nets over a finite field $\mathbb{F}_q$, see if they are realizable over $\mathbb{R}$ or $\mathbb{C}$, that is, if there is an arrangement over these fields with the same intersection poset.



Algorithm

- Start with a random arrangement of n lines in $\mathbb{F}_q\mathbb{P}^2$
- Until we have found a Mod 2–net:
  - Replace a line with a line through 2 randomly chosen intersection points
  - If the new arrangement is better, keep it, otherwise revert to the old state
  - If for a long time, no improvement has been found, start anew with a random arrangement

[3] M. Cuntz. "A greedy algorithm to compute arrangements of lines in the projective plane". In: Discrete & Computational Geometry. (2021)

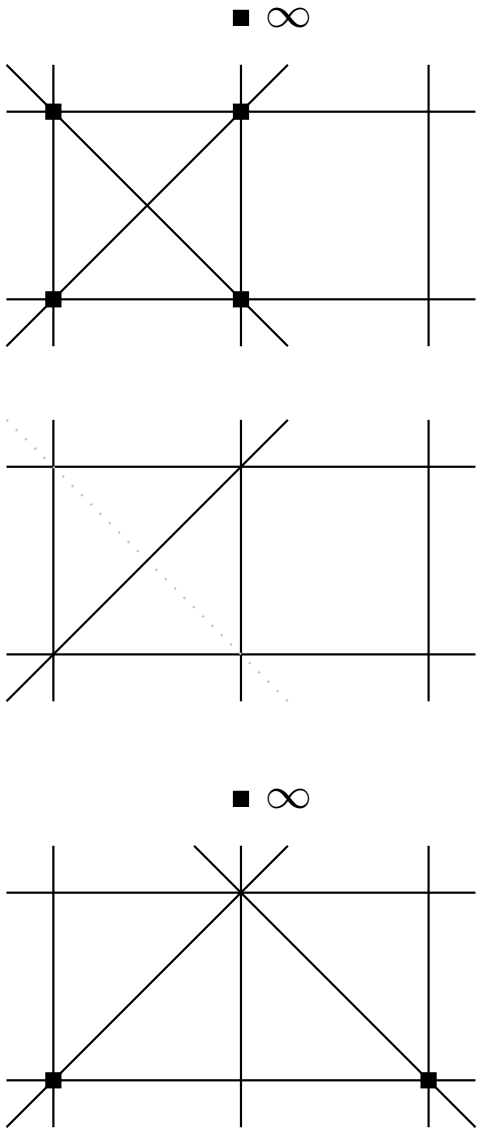
Algorithmic approach

Basis: Greedy algorithm of Michael Cuntz[3].

Main idea: Search for strong Mod 2–nets over a finite field $\mathbb{F}_q$, see if they are realizable over $\mathbb{R}$ or $\mathbb{C}$, that is, if there is an arrangement over these fields with the same intersection poset.

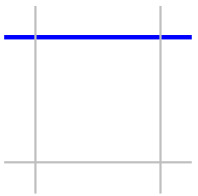
Algorithm

- Start with a random arrangement of n lines in $\mathbb{F}_q\mathbb{P}^2$
- Until we have found a Mod 2–net:
  - Replace a line with a line through 2 randomly chosen intersection points
  - If the new arrangement is better, keep it, otherwise revert to the old state
  - If for a long time, no improvement has been found, start anew with a random arrangement



[3] M. Cuntz. "A greedy algorithm to compute arrangements of lines in the projective plane". In: Discrete & Computational Geometry. (2021)

Determining the goodness א of an arrangement



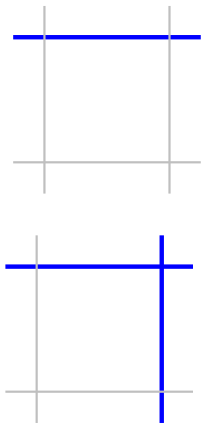
How do we determine if an arrangement is better than another? We use a *goodness function*: it should return a smaller value for the better arrangement; it returns 0 if the arrangement fulfills the criteria.

It is not trivial to find a goodness function w.r.t. the Mod 2–net property.

Main idea:
At an intersection point where only one line is uncolored, the color of the colored lines determines the color of the uncolored line.

Determining the goodness א of an arrangement

How do we determine if an arrangement is better than another? We use a *goodness function*: it should return a smaller value for the better arrangement; it returns 0 if the arrangement fulfills the criteria.



It is not trivial to find a goodness function w.r.t. the Mod 2–net property.

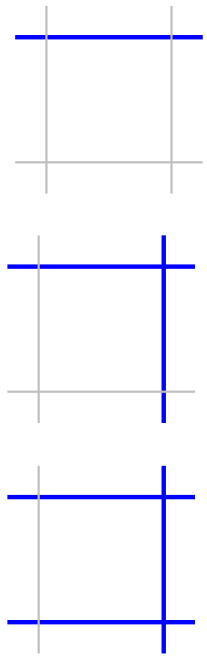
Main idea:
At an intersection point where only one line is uncolored, the color of the colored lines determines the color of the uncolored line.

Determining the goodness א of an arrangement

How do we determine if an arrangement is better than another? We use a *goodness function*: it should return a smaller value for the better arrangement; it returns 0 if the arrangement fulfills the criteria.

It is not trivial to find a goodness function w.r.t. the Mod 2–net property.

Main idea:
At an intersection point where only one line is uncolored, the color of the colored lines determines the color of the uncolored line.

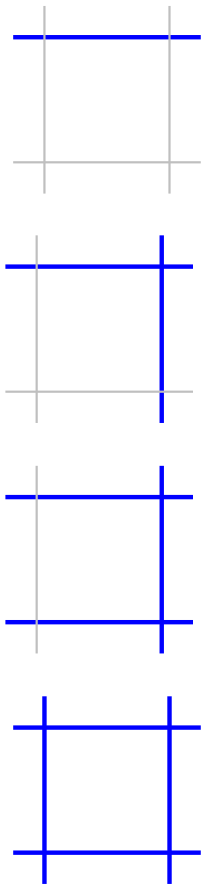


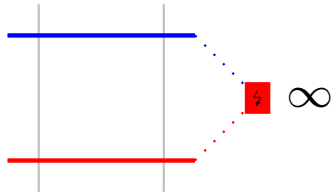
Determining the goodness א of an arrangement

How do we determine if an arrangement is better than another? We use a *goodness function*: it should return a smaller value for the better arrangement; it returns 0 if the arrangement fulfills the criteria.

It is not trivial to find a goodness function w.r.t. the Mod 2–net property.

Main idea:
At an intersection point where only one line is uncolored, the color of the colored lines determines the color of the uncolored line.



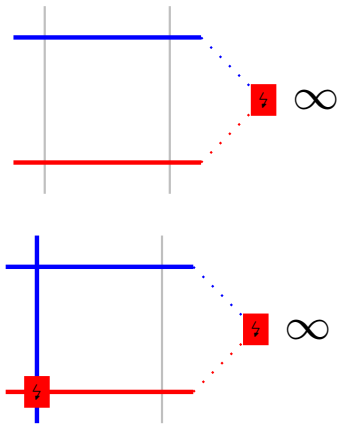


Second idea:

In the final coloring, both colors must be present. To achieve this, we begin each coloring with two different colors.

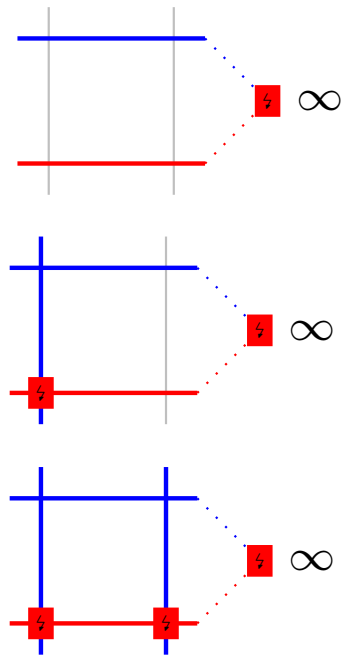
Second idea:

In the final coloring, both colors must be present. To achieve this, we begin each coloring with two different colors.



Second idea:

In the final coloring, both colors must be present. To achieve this, we begin each coloring with two different colors.



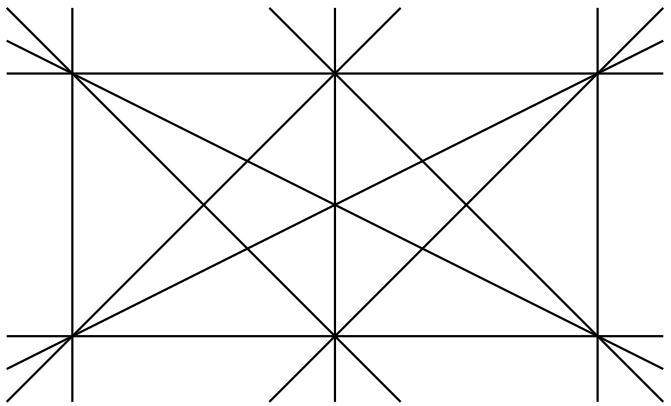
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



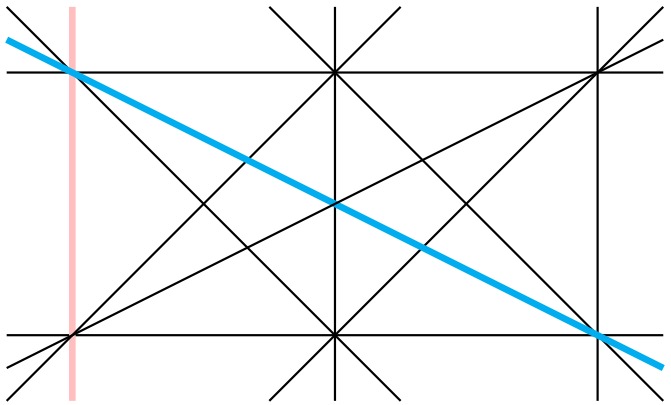
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



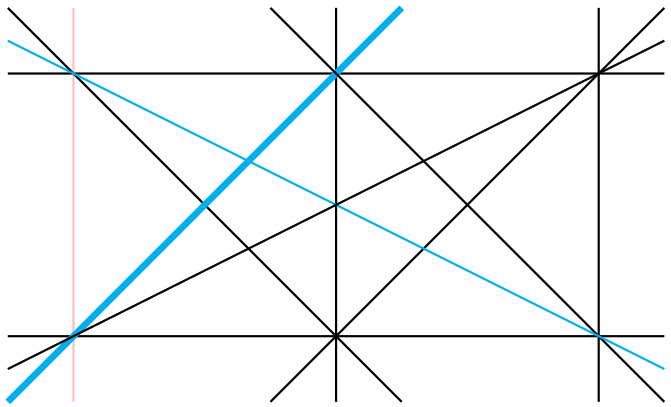
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



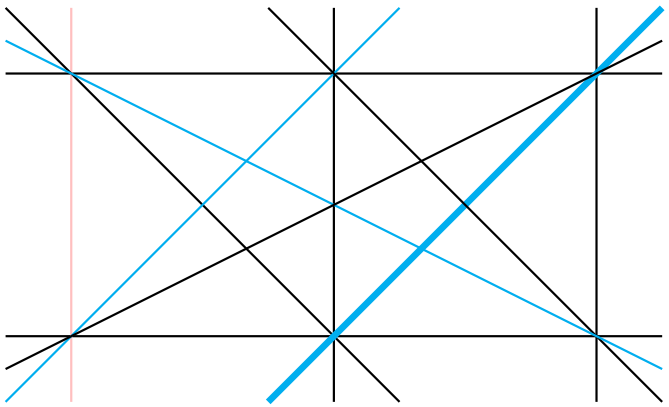
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



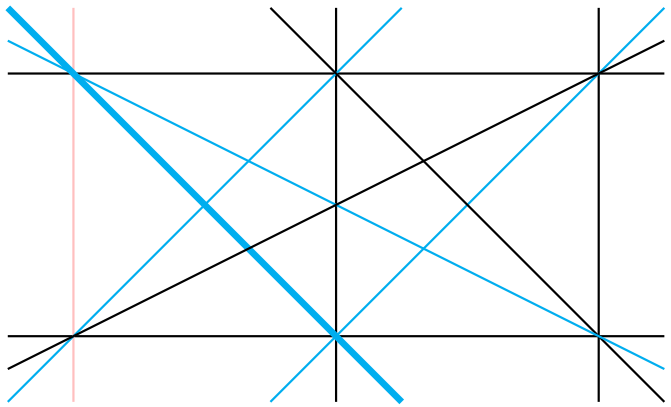
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



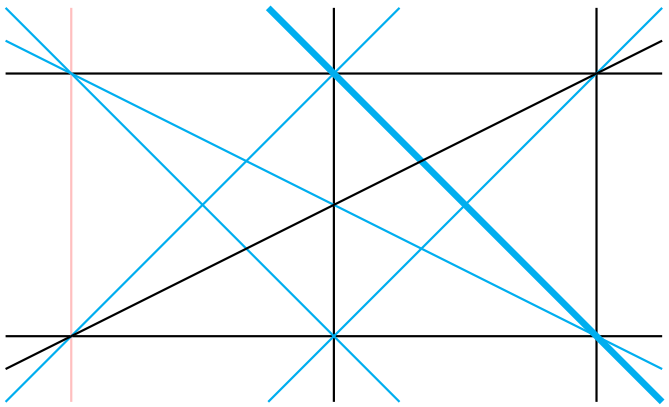
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



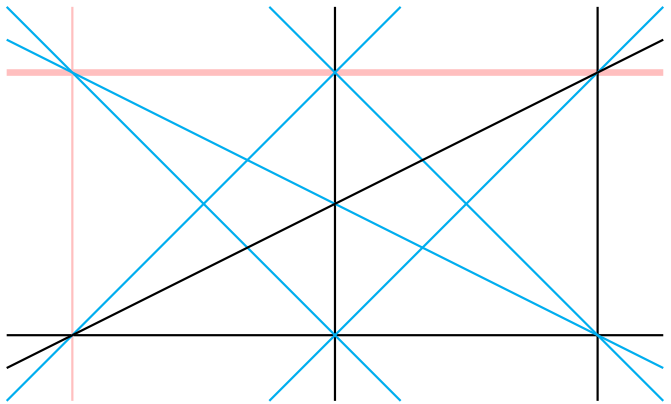
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



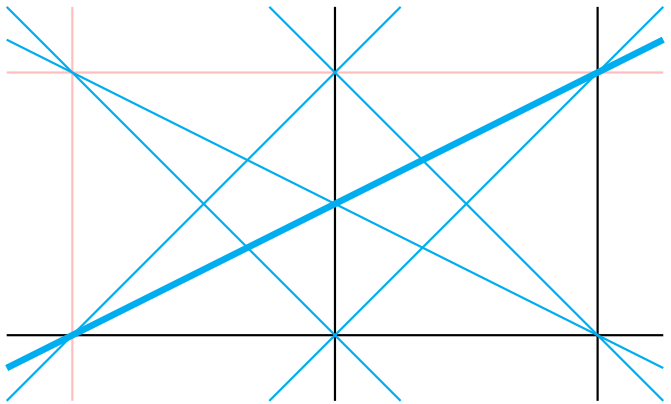
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



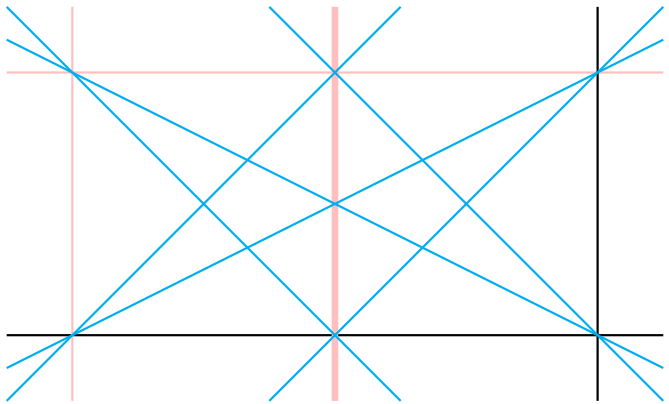
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



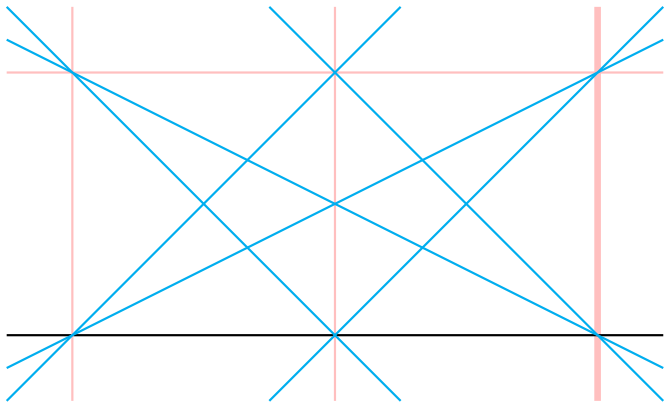
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



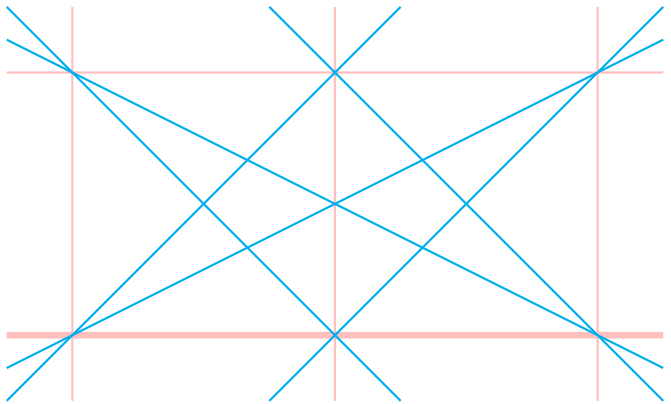
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



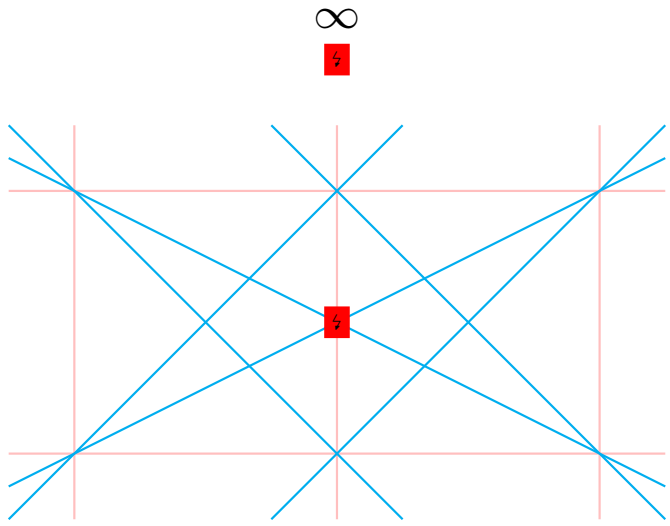
The algorithm of the goodness function $\aleph$:

Begin by coloring 2 lines, 1 with each color, and see what is determined.

If uncolored lines remain, we continue recursively by trying to color the next empty line with both possible colors and seeing which results in a better coloring.

To measure the goodness, count how often we got "bad points" when coloring a new line.

Best is $\aleph = 0$, if we have a full coloring with it, the arrangement is a Mod 2–net.



$\aleph = 2$

Further refinements of the algorithm

The above goodness function is capable of detecting Mod–2 nets.

Problem: the main algorithm is "too greedy" and oftentimes gets stuck at dead ends.

Idea 1 "octopus arms": If the new arrangement is not better than the old one, do not immediately discard the new one, but for a small k, take further steps resulting in arrangements $\text{new}_1, \ldots, \text{new}_k$. If one of the arrangements is actually better than the old one, keep it; otherwise revert to the old one.

Idea 2 "more randomness": If the new arrangement is not better than the old one, with a small probability it should still be accepted. The worse the new arrangement is, the smaller this probability should be.

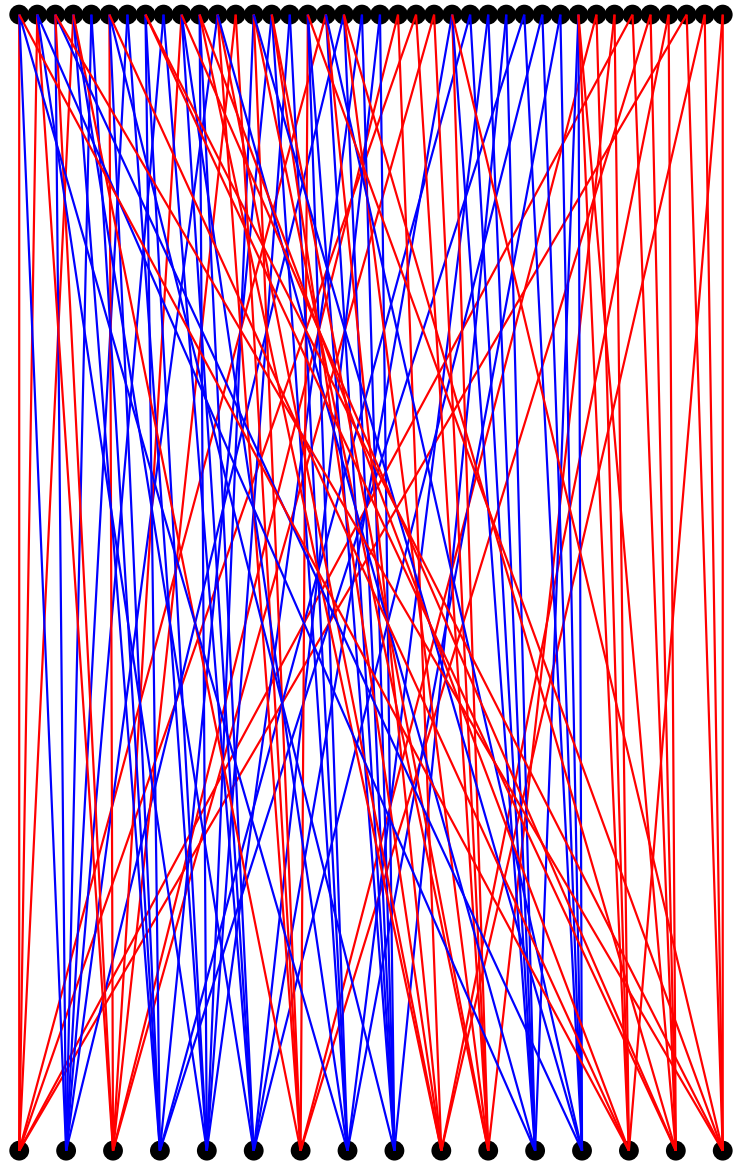
Results

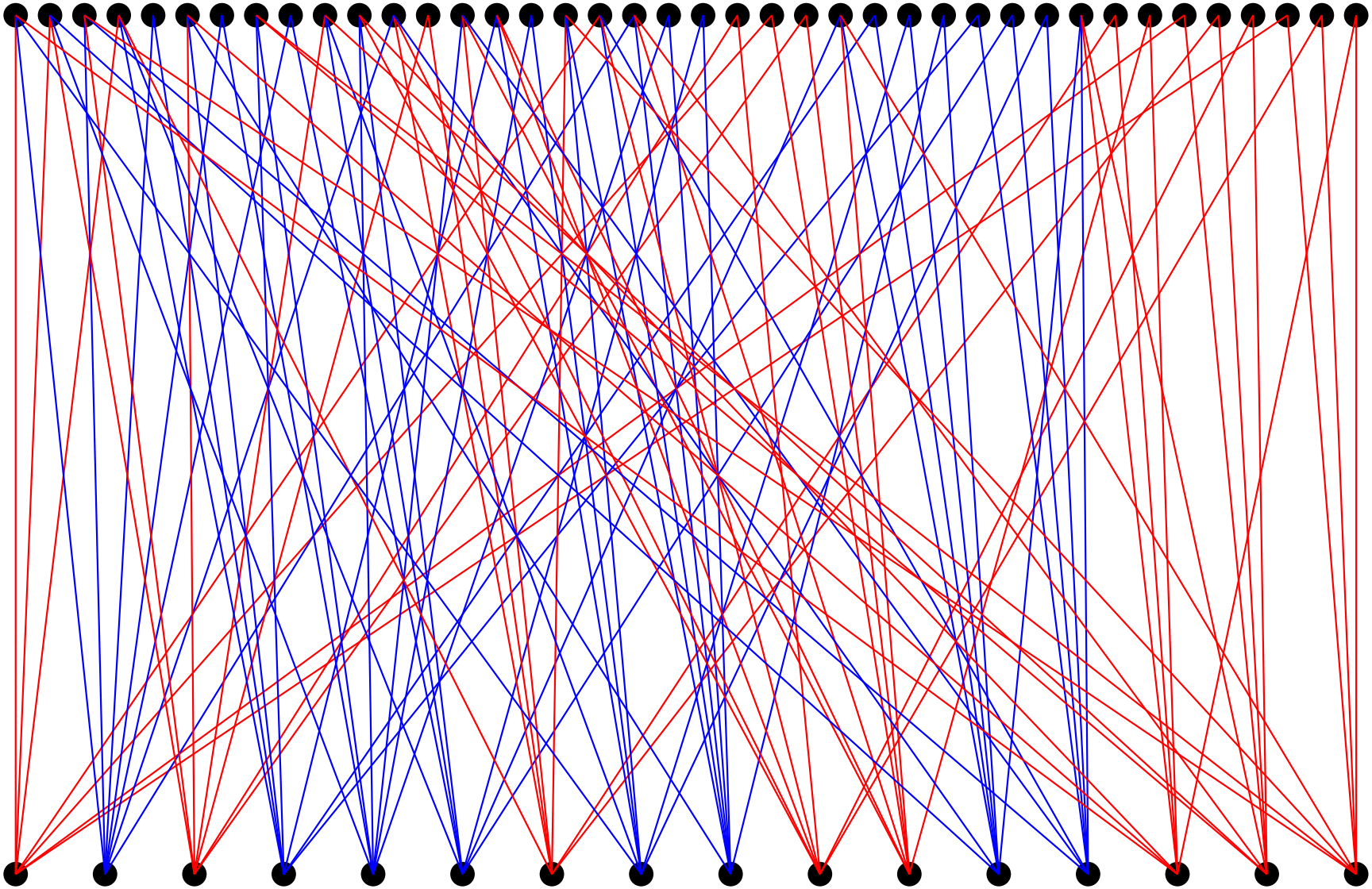
With the algorithm we were able to find:

- Several hundred arrangements isomorphic to the icosidodecahedral arrangement, the only known example of a Mod 2–net over $\mathbb{R}$.
- Several hundred arrangements isomorphic to the Hessian arrangement. This was already a known Mod 2–net, only realizable over $\mathbb{C}$.
- A new class of arrangements with the Mod 2–net property consisting of 16 lines over $\mathbb{F}_7$ that we called *Class 16.1*. Sadly, it is not realizable over a field of characteristic 0.

Instead of directly dealing with arrangements, concentrating on the combinatorial structure of the intersection poset could yield more results.

We were able to show with combinatorial methods that an arrangement satisfying the Mod 2–net property over any field must consist of at least 12 lines.





The intersection poset of arrangements from Class 16.1