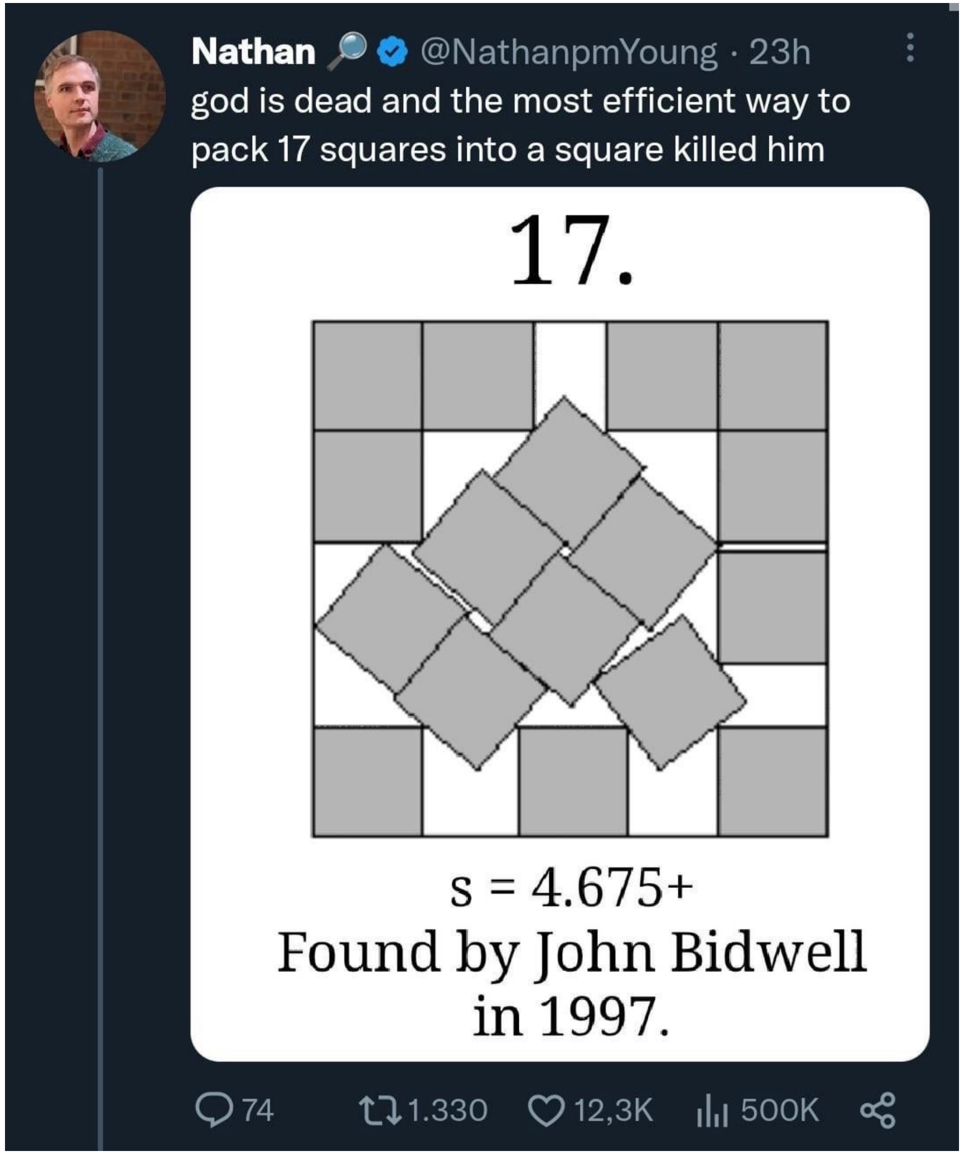
Automated Symmetric Constructions in Discrete Geometry

Bernardo Subercaseaux, Ethan Mackey, Long Qian, Marijn J. H. Heule
Carnegie Mellon University

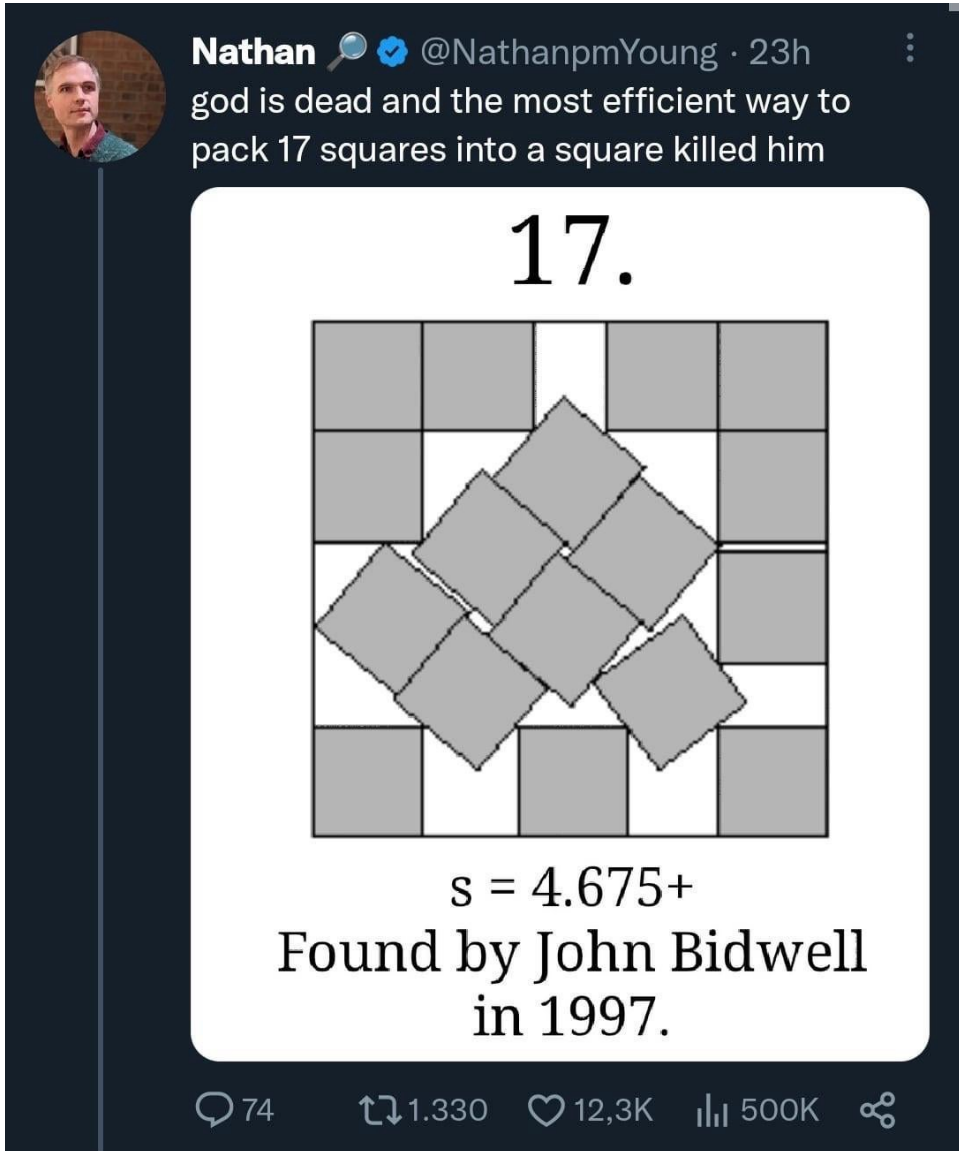
The smallest square we know that can fit 17 unit-squares has side ~4.675.

It's very ugly, and not known to be optimal.



The smallest square we know that can fit 17 unit-squares has side ~4.675.

It's very ugly, and not known to be optimal.



how can we look for

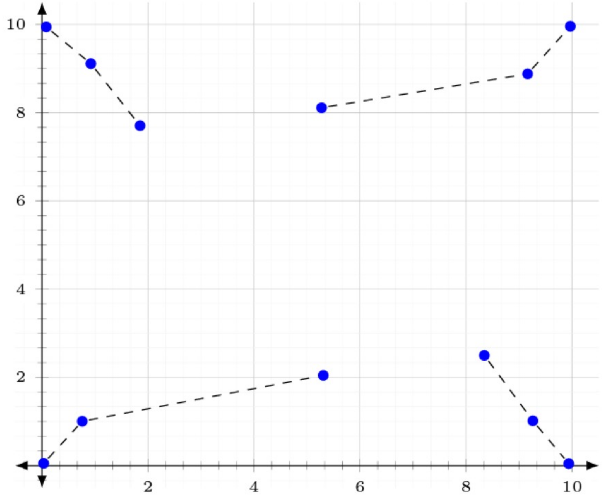
- beautiful solutions?
- clear patterns?
- generalizability?

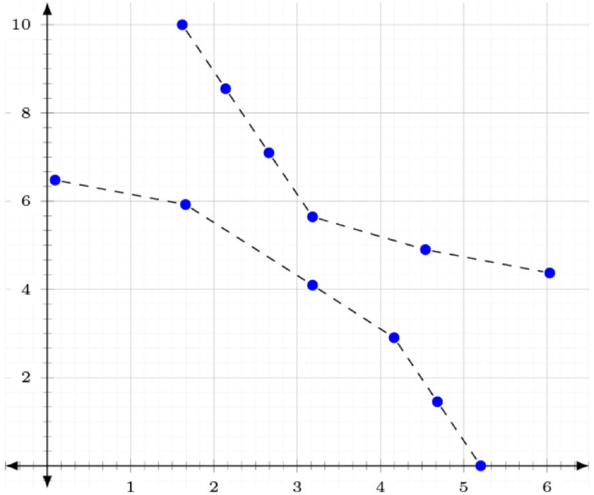
Story from CICM 2024

We were trying to understand the minimum number of convex pentagons for N points in the plane without three of them on a line

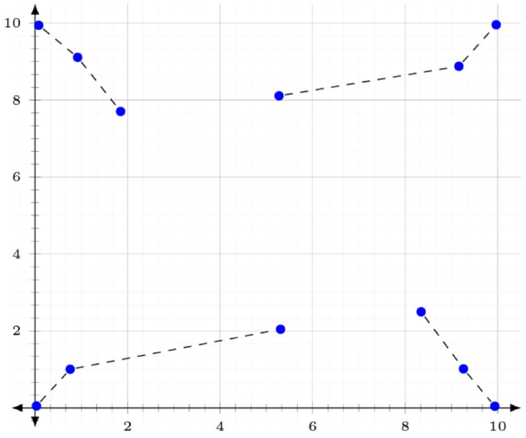
Story from CICM 2024

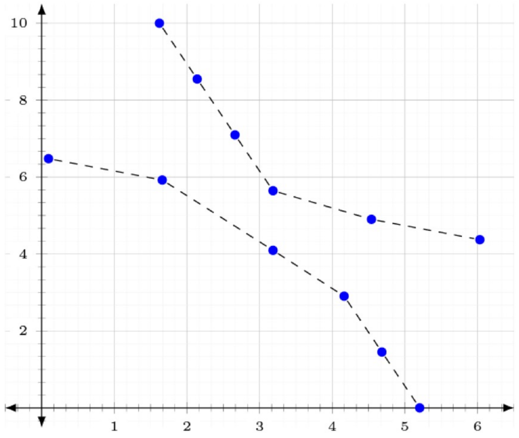
We were trying to understand the minimum number of convex pentagons for N points in the plane without three of them on a line



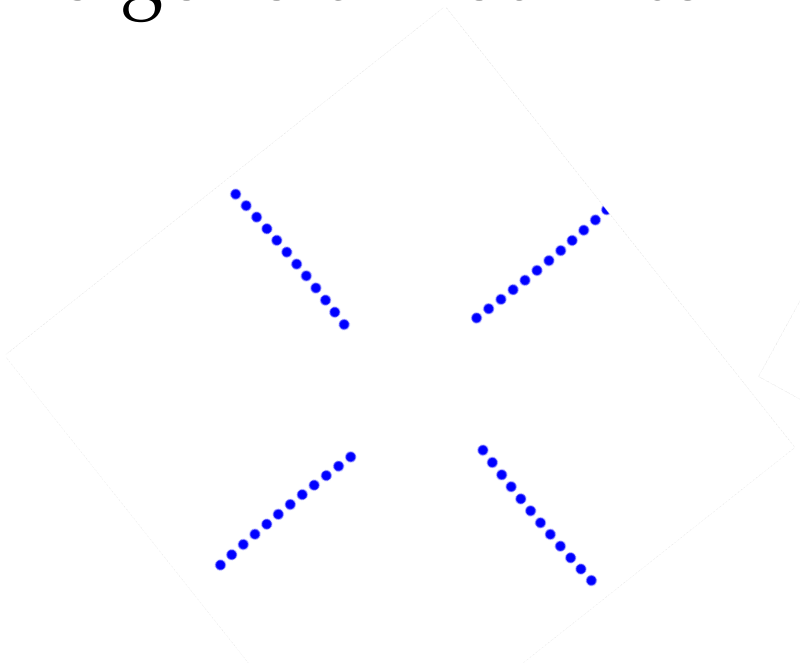


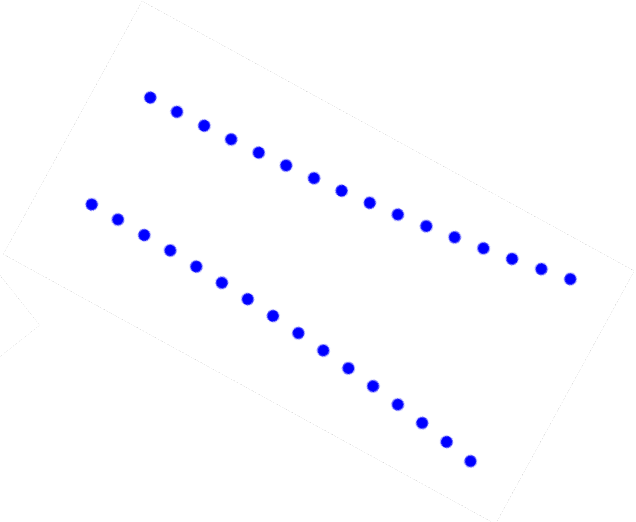
We used SAT solvers and other computational tools to find optimal solutions for small N





The solutions generalized into nice constructions for all N





Story from CICM 2024

We were lucky!

In general, even if nice patterned solutions exist

1) computer searches might find the ugly ones first
2) usually too many to enumerate them all and check for a nice one

Story from CICM 2024

We were lucky!

In general, even if nice patterned solutions exist

1) computer searches might find the ugly ones first
2) usually too many to enumerate them all and check for a nice one

Question: how do we factor in "niceness" into our constraints/objective?

Story from CICM 2024

We were lucky!

In general, even if nice patterned solutions exist

1) computer searches might find the ugly ones first
2) usually too many to enumerate them all and check for a nice one

Question: how do we factor in "niceness" into our constraints/objective?

Answer is context dependent,

but we will show that **symmetry** is a great place to start!

Happy Ending Problem (Erdős-Szekeres conjecture)

Given $k \geq 3$, how many points in the plane, without three on a line, guarantee a convex k-gon?

Happy Ending Problem (Erdős-Szekeres conjecture)

Given $k \geq 3$, how many points in the plane,
without three on a line, guarantee a convex k-gon?

Esther Klein

George Szekeres

Paul Erdős

Happy Ending Problem (Erdős-Szekeres conjecture)

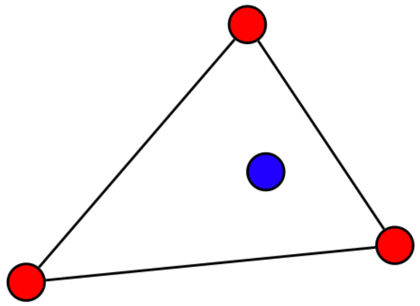
Given $k \geq 3$, how many points in the plane,
without three on a line, guarantee a convex k-gon?

Trivial: $g(3) = 3$

Happy Ending Problem (Erdős-Szekeres conjecture)

Given $k \geq 3$, how many points in the plane,
without three on a line, guarantee a convex k-gon?

Interesting: $g(4) = 5$

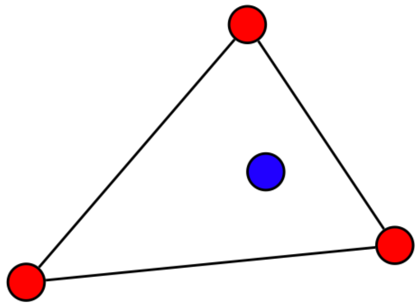


4 points is not enough

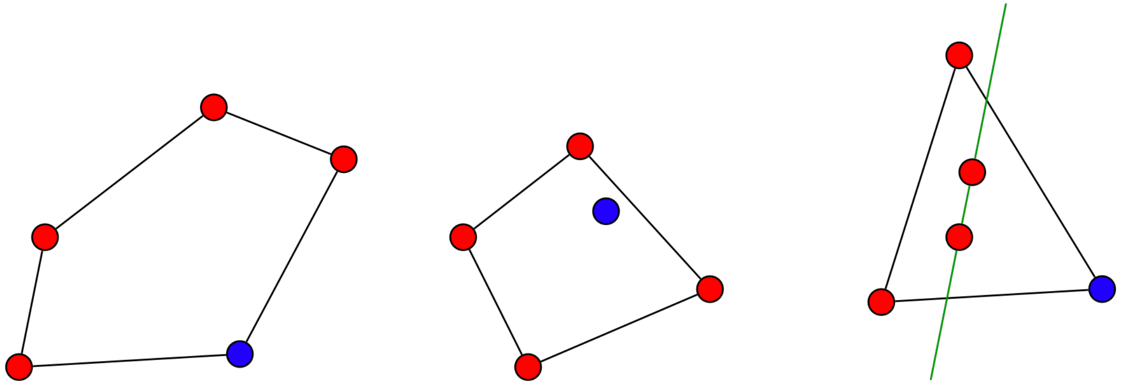
Happy Ending Problem (Erdős-Szekeres conjecture)

Given $k \geq 3$, how many points in the plane,
without three on a line, guarantee a convex k-gon?

Interesting: $g(4) = 5$



4 points is not enough



(a) (b) (c)

5 points are enough (Klein, ~1933)

Happy Ending Problem (Erdős-Szekeres conjecture)

$$g(5) = 9,\ g(6) = 17,\ g(7) = ???$$

Happy Ending Problem (Erdős-Szekeres conjecture)

$$g(5) = 9,\ g(6) = 17,\ g(7) = ???$$

The bound $g(6) = 17$ was published in 2006 by Lindsay Peters and George Szekeres.

Szekeres died at 95, months before the paper appeared in print, and within one hour of the passing of his wife Esther Szekeres-Klein.
You can read a nice obituary at tinyurl.com/SK-Obituary

Happy Ending Problem (Erdős-Szekeres conjecture)

$g(5) = 9, g(6) = 17, g(7) = ???$

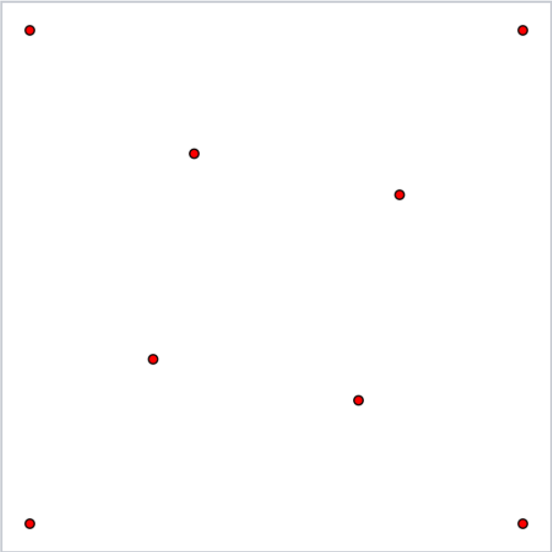
$g(k) = 2^{k-2} + 1$?!

Happy Ending Problem (Erdős-Szekeres conjecture)

$$g(5) = 9,\ g(6) = 17,\ g(7) = ???$$

Wikipedia page right now:

Nice!



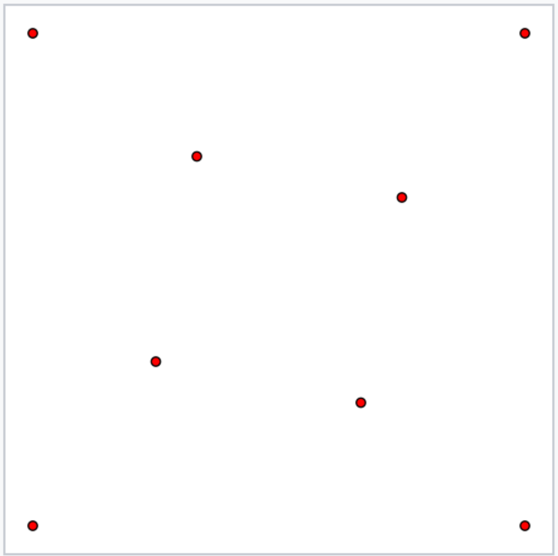
A set of eight points in general position with no convex pentagon

Happy Ending Problem (Erdős-Szekeres conjecture)

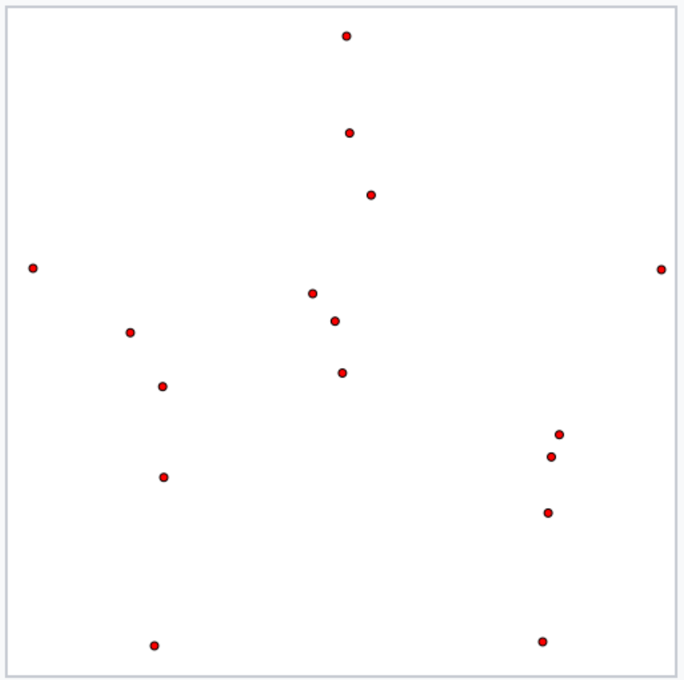
$$g(5) = 9,\ g(6) = 17,\ g(7) = ???$$

Wikipedia page right now:

Nice!



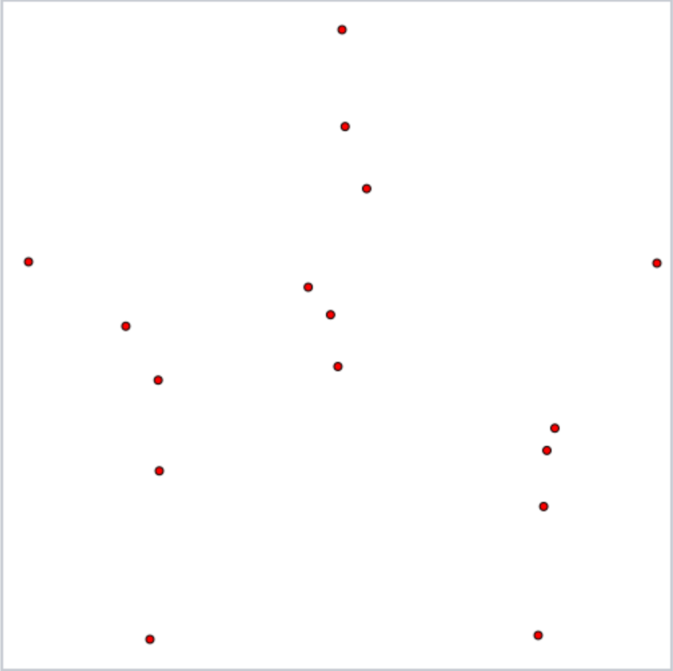
A set of eight points in general position with no convex pentagon



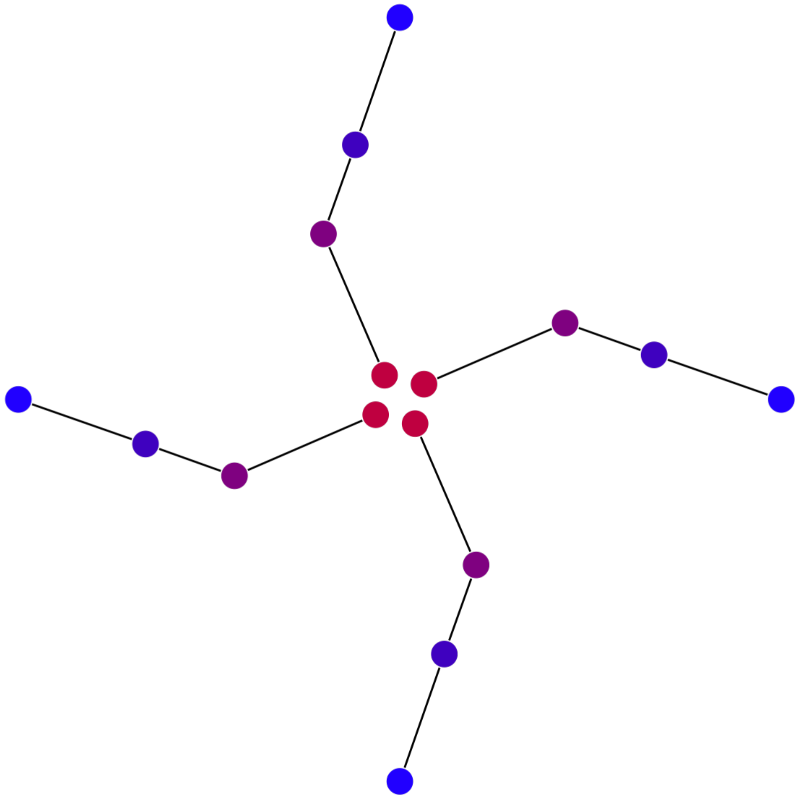
A set of sixteen points in general position with no convex hexagon

Not so nice ☹

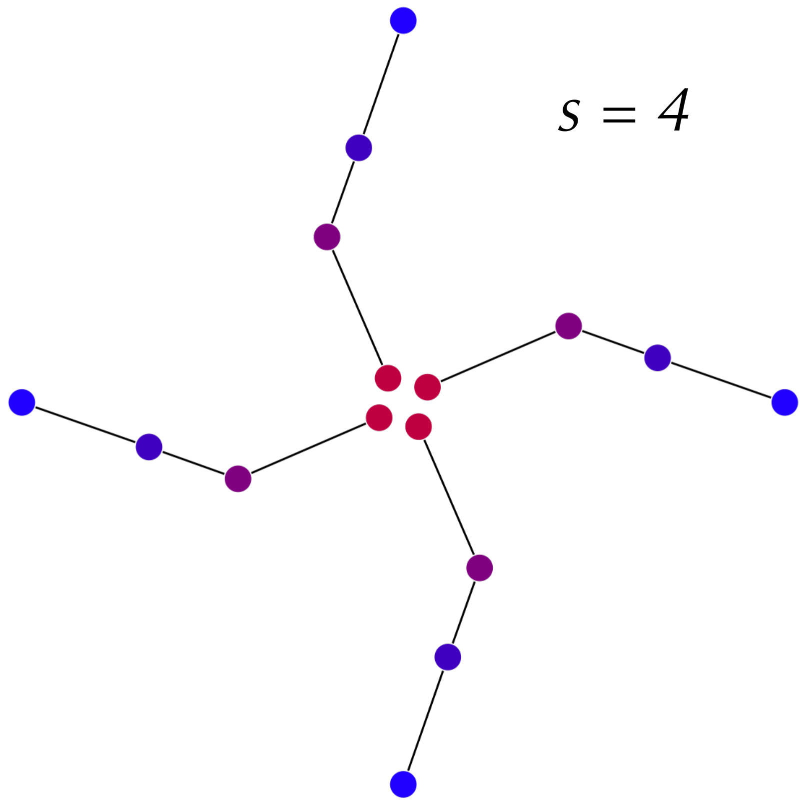
An example of what we achieve in our paper:

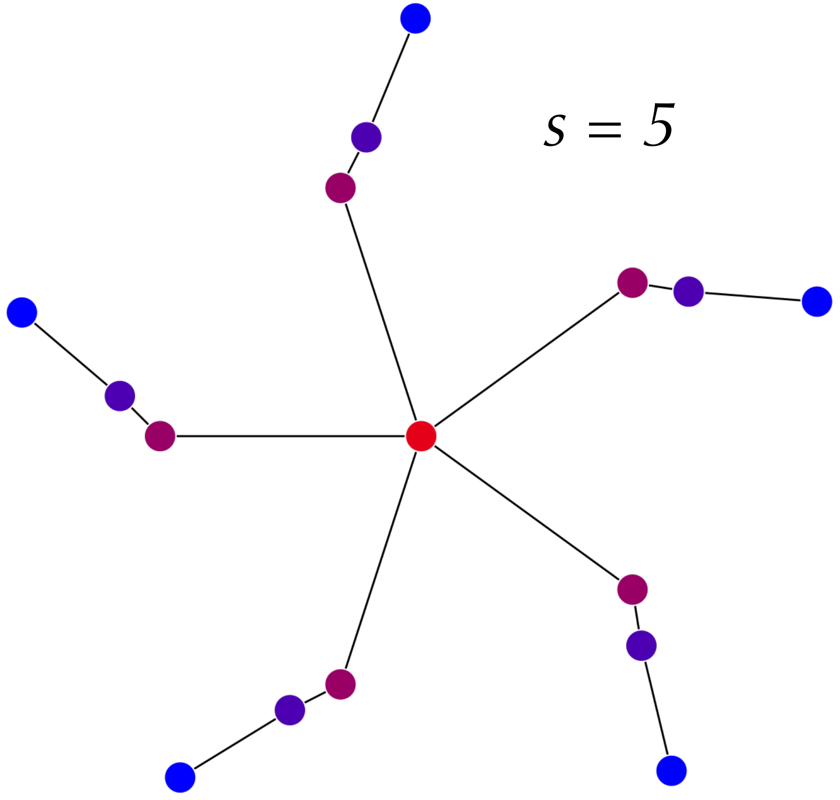


A set of sixteen points in general position with no convex hexagon

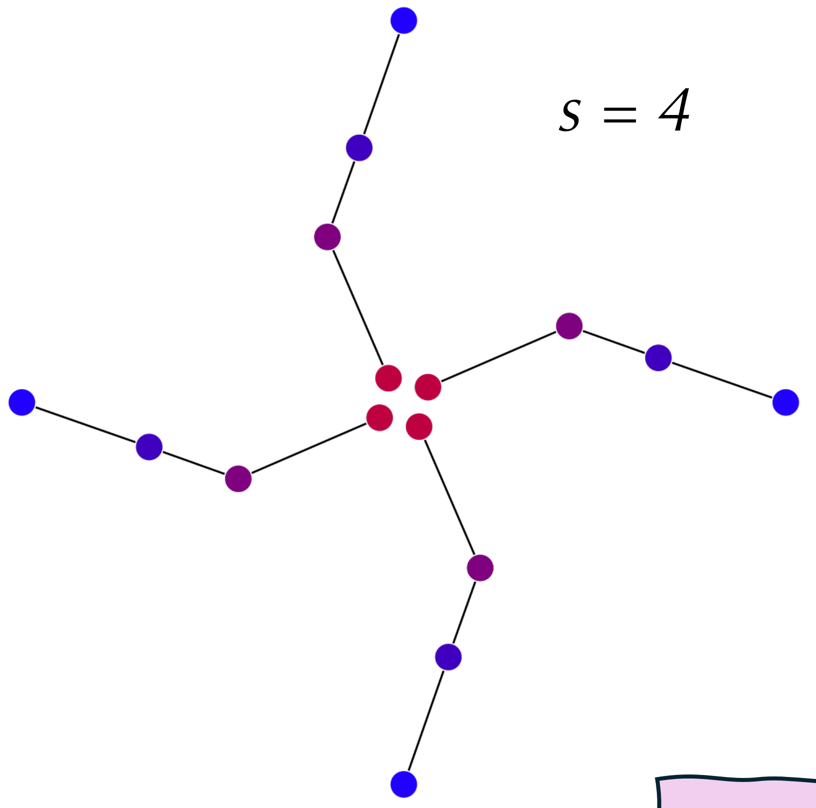


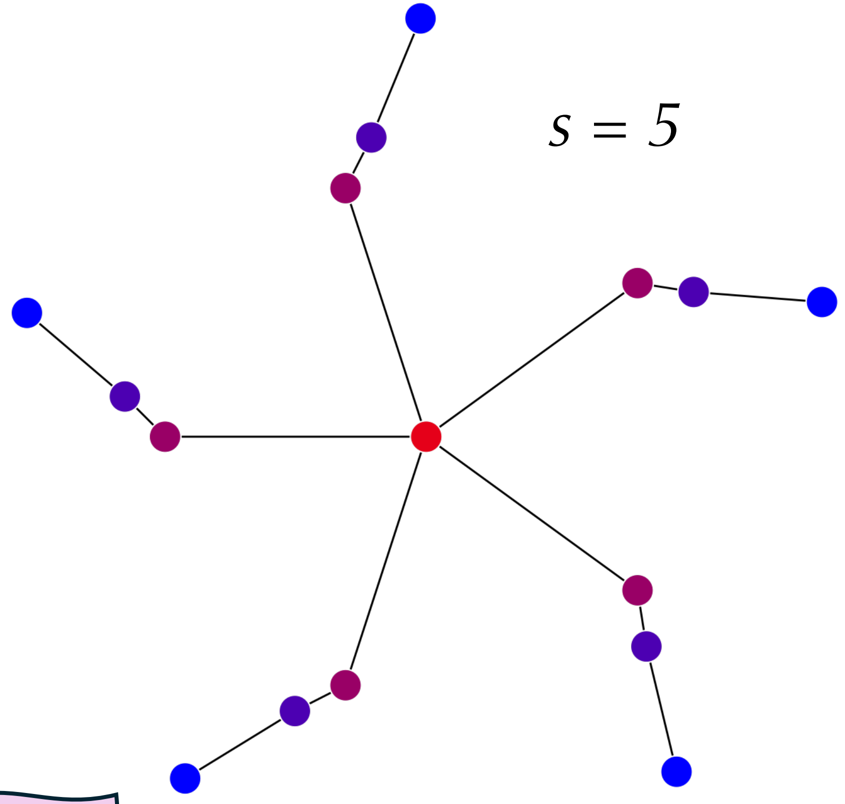
An example of what we achieve in our paper:





An example of what we achieve in our paper:





No 3-fold symmetric solution exists!

Automated Symmetric Constructions in Discrete Geometry

Bernardo Subercaseaux, Ethan Mackey, Long Qian, Marijn J. H. Heule
Carnegie Mellon University

All you need to know about SAT solving

Q: What is a SAT solver?
A: A highly optimized program for deciding if a Boolean formula is satisfiable

All you need to know about SAT solving

Q: What is a SAT solver?
A: A highly optimized program for deciding if a Boolean formula is satisfiable

Q: A Boolean formula?

A: $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2)$

All you need to know about SAT solving

Q: What is a SAT solver?
A: A highly optimized program for deciding if a Boolean formula is satisfiable

Q: A Boolean formula?

A: $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2)$

Q: Satisfiable?

A: Yes! $(x_1, x_2, x_3) := (0, 1, 1)$

All you need to know about SAT solving

Q: What is a SAT solver?
A: A highly optimized program for deciding if a Boolean formula is satisfiable

Q: A Boolean formula?

A: $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2)$

Q: Satisfiable?

A: Yes! $(x_1, x_2, x_3) := (0, 1, 1)$

Q: And how do I write this on my computer ?

A:

test.cnf

```
p cnf 3 2
1 -2 3 0
-1 2 0
```

All you need to know about SAT solving

Q: And then what?

A:

```
terminal

> kissat test.cnf
[...]
s SATIISFIABLE
v -1 2 3 0
```

or

```
terminal

> kissat test2.cnf
[...]
s UNSATISFIABLE
```

or

All you need to know about SAT solving

Q: And then what?

A:

terminal

```
> kissat test.cnf
[...]
s SATIISFIABLE
v -1 2 3 0
```

or

terminal

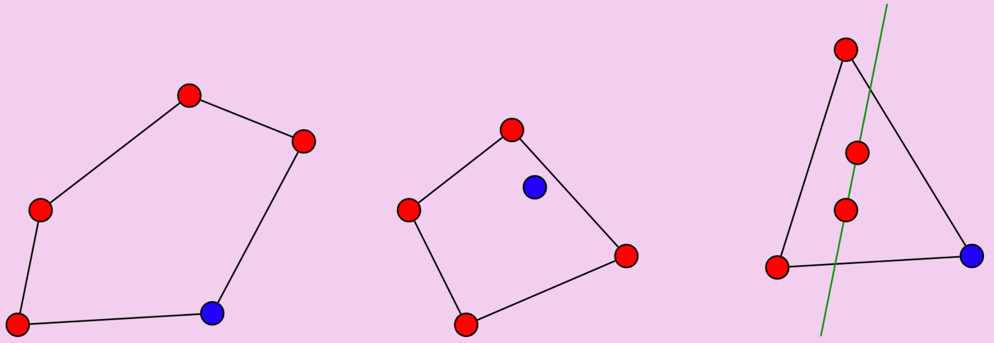
```
> kissat test2.cnf
[...]
s UNSATISFIABLE
```

or

You can also get a reproducible proof!

Now I'll talk about discrete geometry,
and using SAT for it.

I want you to keep in mind the insight of Esther Klein:

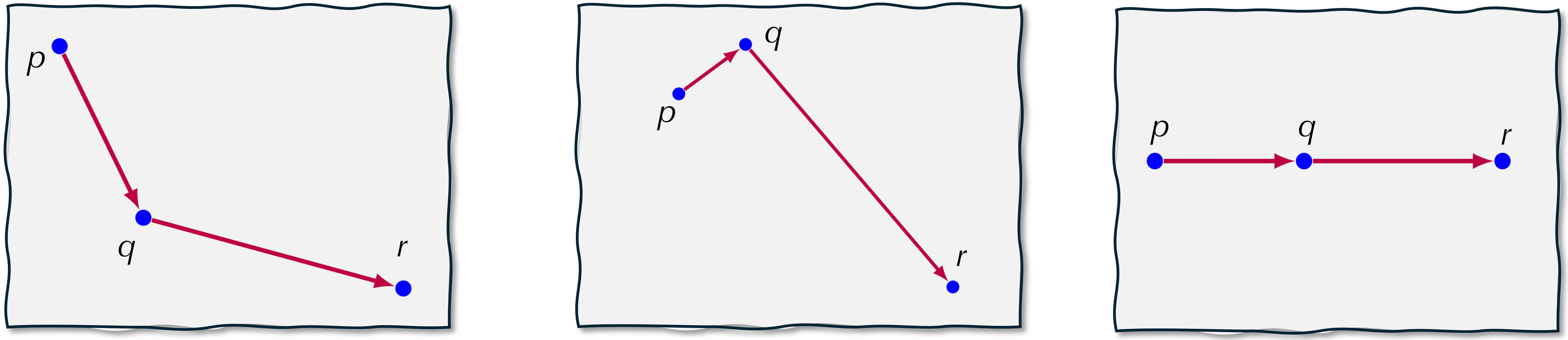


(a) (b) (c)

$\mathbb{R}^2$ might be uncountably infinite,

But a finite number of well-chosen cases might answer our questions.

Discrete Orientations (Knuth CC-Axioms)



Counterclockwise (CC)

$A_{p,q,r}$

p is **Above** directed line qr

Clockwise

$B_{p,q,r}$

p is **Below** directed line qr

Collinear

$C_{p,q,r}$

Discrete Orientations (~Knuth CC-Axioms)

The orientations of *(q, r, p)*, or *(q, p, r)* or *(p, r, q)*, *etc*…
can all be deduced from *(p, q, r)*.

Instead of writing e.g., $A_{p,\,r,\,q}$ we can write $\neg A_{p,\,q,\,r}$

We thus have $3\binom{n}{3}$ variables for n points.

Discrete Orientations (~Knuth CC-Axioms)

Some combinations of variables don't make geometric sense!

$$\boxed{C_{p,q,r} \wedge C_{q,r,s} \wedge A_{p,q,s}}$$

Axiom 1. $C_{p,q,r} \wedge C_{q,r,s} \rightarrow C_{p,q,s}$

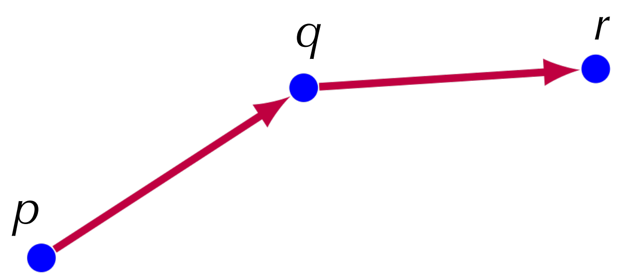
Axiom 2. $C_{p,q,r} \wedge C_{q,r,s} \rightarrow C_{p,r,s}$

Axiom 3. $C_{p,q,r} \wedge A_{q,r,s} \rightarrow A_{p,r,s}$

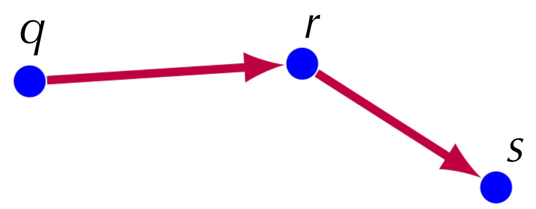
(and so on...)

Discrete Orientations (~Knuth CC-Axioms)

Some axioms discard a significant portion of “ungeometrical configurations”,
But require points to be sorted left-to-right.



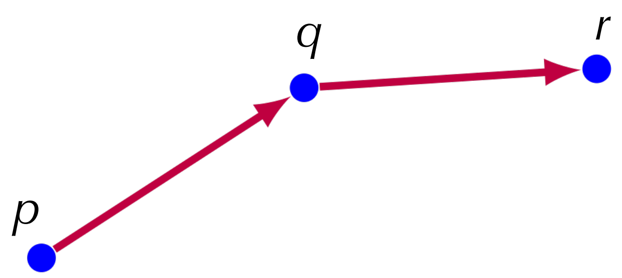
pqr is clockwise



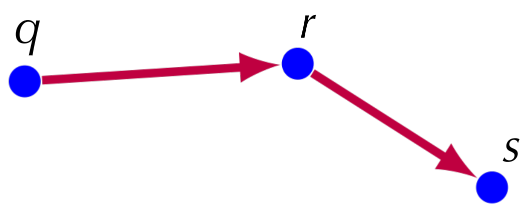
qrs is clockwise

Discrete Orientations (~Knuth CC-Axioms)

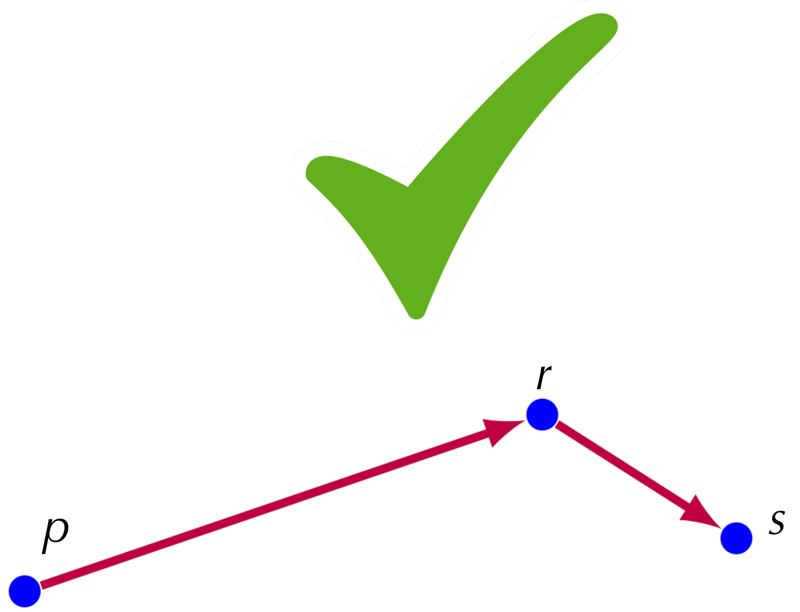
Some axioms discard a significant portion of "ungeometrical configurations",
But require points to be sorted left-to-right.



pqr is clockwise



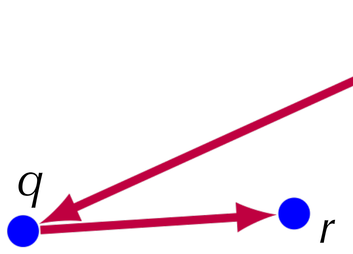
qrs is clockwise



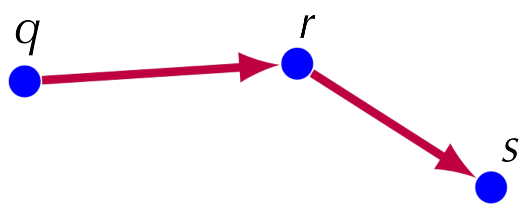
prs is clockwise

Discrete Orientations (~Knuth CC-Axioms)

Some axioms discard a significant portion of “ungeometrical configurations”,
But require points to be sorted left-to-right.



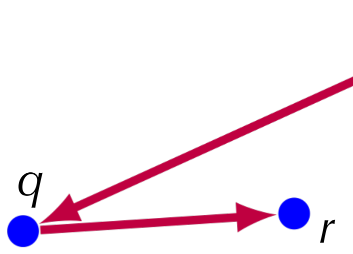
pqr is clockwise



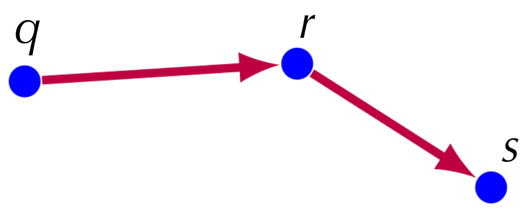
qrs is clockwise

Discrete Orientations (~Knuth CC-Axioms)

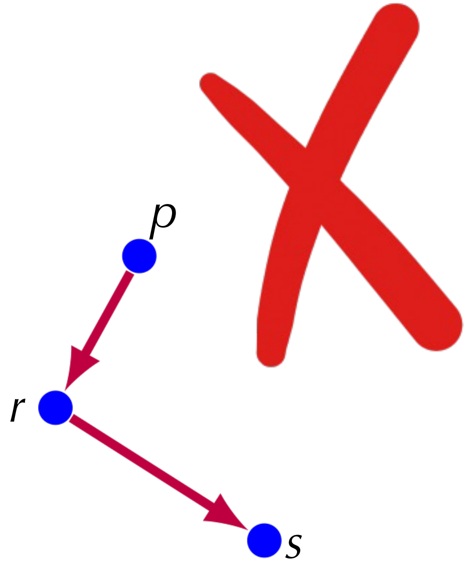
Some axioms discard a significant portion of “ungeometrical configurations”,
But require points to be sorted left-to-right.



pqr is clockwise



qrs is clockwise



prs is **NOT** clockwise !

Erdős-Szekeres Encoding

Goal: n points, without 3 on a line, that do not contain a convex k-gon.

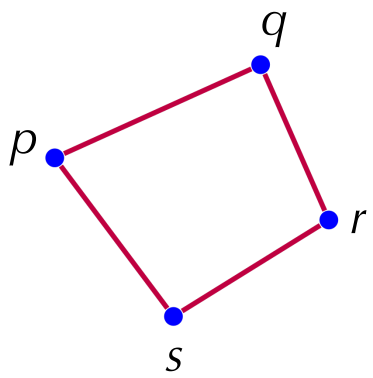
Base variables: $A_{p,q,r}$ (since no collinearities)

Erdős-Szekeres Encoding

Goal: n points, without 3 on a line, that do not contain a convex k-gon.

Base variables: $A_{p,q,r}$ (since no collinearities)

Auxiliary variables: $\mathsf{Conv}_{p,q,r,s}$

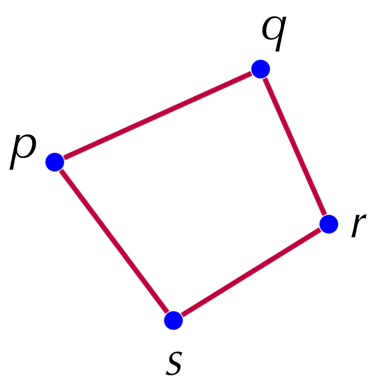


Erdős-Szekeres Encoding

Goal: n points, without 3 on a line, that do not contain a convex k-gon.

Base variables: $A_{p,q,r}$ (since no collinearities)

Auxiliary variables: $\mathsf{Conv}_{p,q,r,s}$



$$\mathsf{Conv}_{p,q,r,s} \leftrightarrow ((A_{p,q,r} \leftrightarrow A_{p,r,s}) \leftrightarrow (A_{p,r,s} \leftrightarrow A_{q,r,s}))$$

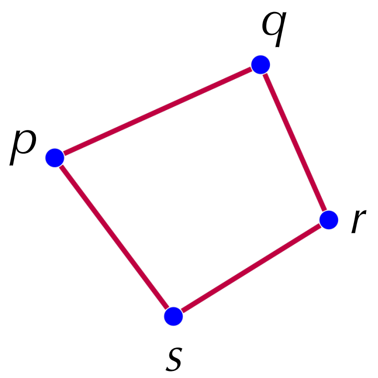
Similar to Scheucher (2020)

Erdős-Szekeres Encoding

Goal: n points, without 3 on a line, that do not contain a convex k-gon.

Base variables: $A_{p,q,r}$ (since no collinearities)

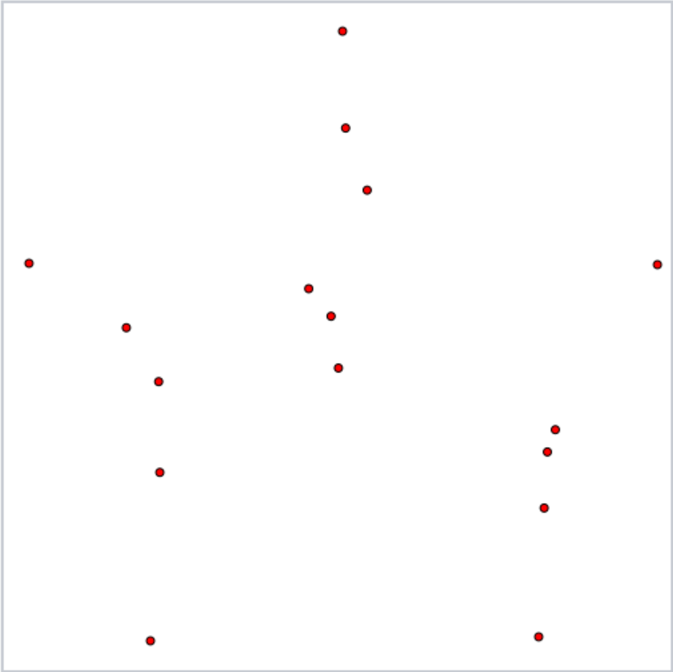
Auxiliary variables: $\mathsf{Conv}_{p,q,r,s}$



Main constraint: for every subset of k-points, at least one of its 4-subsets is not convex.

Symmetry Constraints

We want *s*-fold rotational symmetries

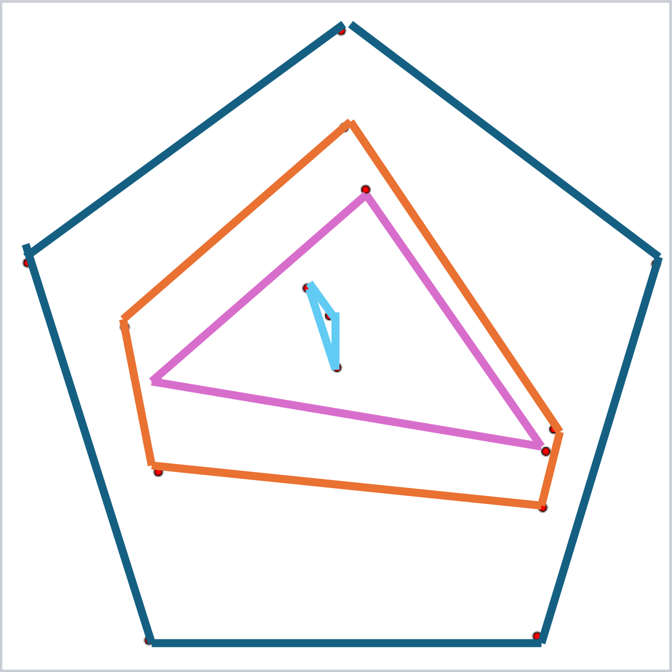


A set of sixteen points in general position with no convex hexagon

To illustrate by example,
remember the Wikipedia picture.

Symmetry Constraints

We want *s*-fold rotational symmetries

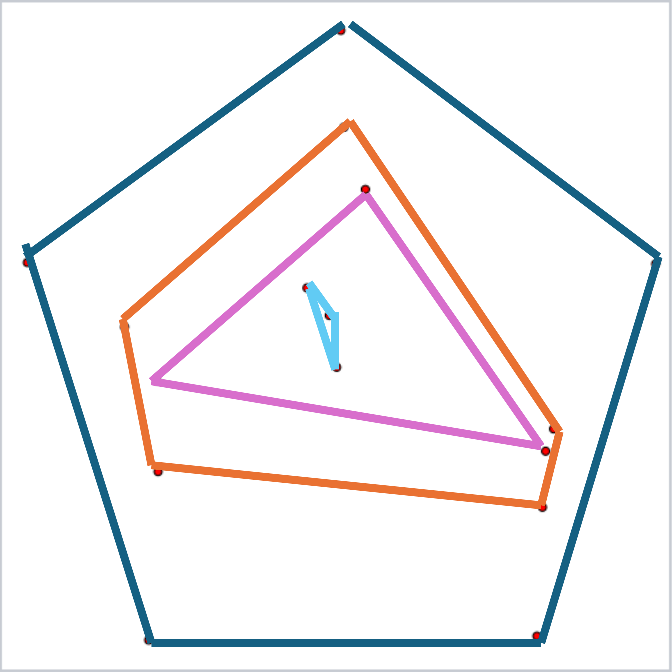


A set of sixteen points in general position with no convex hexagon

Its *convex-hull onion* is 5-5-3-3

Symmetry Constraints

We want *s*-fold rotational symmetries



A set of sixteen points in general position with no convex hexagon

Its *convex-hull onion* is 5-5-3-3

This cannot be made 5-fold symmetric nor 3-fold symmetric

So let's aim for **5-5-5-1**, which can be made 5-fold symmetric.

Symmetry Constraints

We want *s*-fold rotational symmetries

Our symmetric 5-5-5-1 constructions will have orbits:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$$

$$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$$

$$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$$

$$16$$

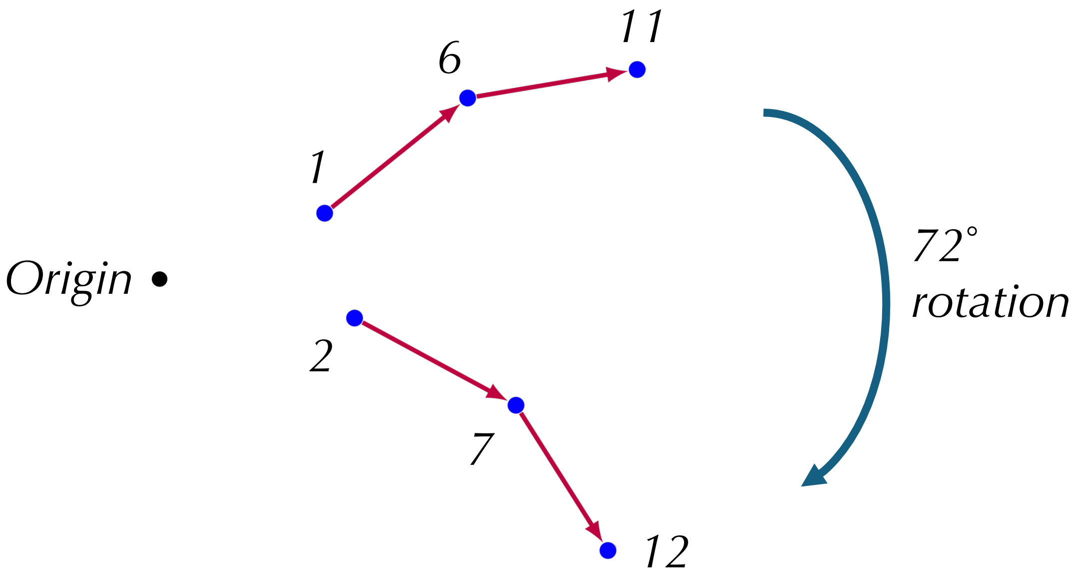
Symmetry Constraints

We want *s*-fold rotational symmetries

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$$
$$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$$
$$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$$
$$16$$

This implies that

$$A_{1,6,11} \leftrightarrow A_{2,7,12} \leftrightarrow \ldots$$



Symmetry Constraints

We want *s*-fold rotational symmetries

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$

$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$

16

View this as cycles of permutation π

$$A_{p,q,r} \leftrightarrow A_{\pi(p),\pi(q),\pi(r)}$$

Remember left-to-right ordering axioms?

Knuth (1992) showed that these reduce search space dramatically:

$$\boxed{2^{O(n^3)} \to 2^{O(n^2)}}$$

If we don't enforce symmetry, we can always relabel points so that they're sorted left-to-right (point 1 is leftmost, point n is rightmost)

But when enforcing a symmetry π, we commit to certain relationships between the labels.

Remember left-to-right ordering axioms?

Knuth (1992) showed that these reduce search space dramatically:

$$\boxed{2^{O(n^3)} \rightarrow 2^{O(n^2)}}$$

If we don't enforce symmetry, we can always relabel points so that they're sorted left-to-right (point 1 is leftmost, point *n* is rightmost)

But when enforcing a symmetry π, we commit to certain relationships between the labels.

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$

$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$

16

Example
point 16 is mapped to itself, which implies its at the origin, and thus cannot be the rightmost

Remember left-to-right ordering axioms?

But when enforcing a symmetry π, we commit to certain relationships between the labels.

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$

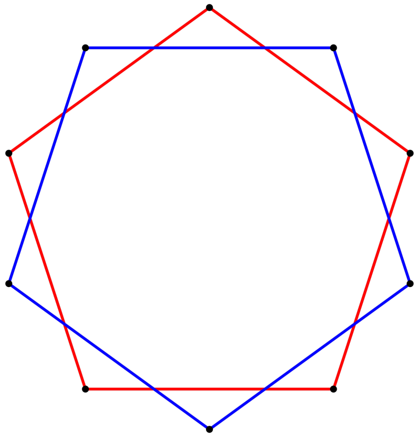
$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$

16

Example 2

Orbits must remain in the same convex-hull layer; thus either (1,2,3,4,5) are all inside (11,12,13,14,15) or viceversa

Or there is a big convex layer of 10 points mixing two orbits.



Remember left-to-right ordering axioms?

But when enforcing a symmetry π, we commit to certain relationships between the labels.

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$

$11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$

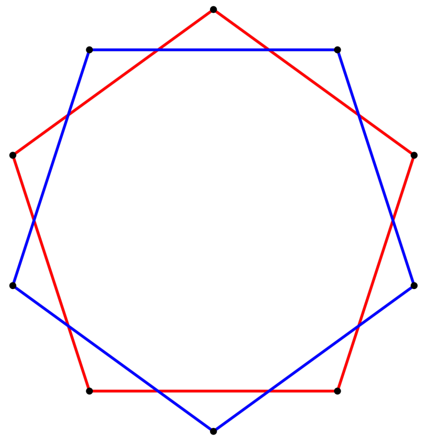
16

Example 2

Orbits must remain in the same convex-hull layer; thus either (1,2,3,4,5) are all inside (11,12,13,14,15) or viceversa

Or there is a big convex layer of 10 points mixing two orbits.

So how can we enforce symmetries while still enforcing the left-to-right axioms that dramatically reduce search space?



Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

Which one though?

You

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

Which one though?

You

I don't know!
We just let the solver figure it out

Me

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

We add variables $\prec_{p,q}$ which the solver will assign

Totality. $(\prec_{p,q} \vee \prec_{q,p})$, for all $1 \leq p \neq q \leq n$.

Asymmetry. $(\prec_{p,q} \leftrightarrow \overline{\prec_{p,q}})$, for all $1 \leq p \neq q \leq n$.

Transitivity. $(\prec_{p,q} \wedge \prec_{q,r} \rightarrow \prec_{p,r})$, for all $1 \leq p \neq q \neq r \leq n$.

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

We add variables $\prec_{p,q}$ which the solver will assign

And then use this ordering to *"guard"* the axioms so that they only apply to ordered tuples

$$(\prec_{p,q} \wedge \prec_{q,r} \wedge \prec_{r,s}) \rightarrow (A_{p,q,r} \vee \overline{A_{p,r,s}} \vee A_{q,r,s})$$

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

Why on earth would any linear ordering be correct?!

You

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

Why on earth would any linear ordering be correct?!

You

I don't know!
I proved this system is equivalent to Knuth's axioms using a SAT solver!

Me

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

In 2024 we verified Knuth's axioms in Lean, as part of a larger project

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

In 2024 we verified Knuth's axioms in Lean, as part of a larger project

My advisor Marijn Heule had the brilliant idea of using arbitrary orderings

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

In 2024 we verified Knuth's axioms in Lean, as part of a larger project

My advisor Marijn Heule had the brilliant idea of using arbitrary orderings

I proved that if this wasn't equivalent to Knuth's axioms, there would be a counterexample with 5 points

Dynamic Ordering Axioms

(most interesting theoretical contribution of our paper, in my opinion)

Instead of enforcing a left-to-right order, we enforce *some* linear ordering $\prec$ between the points

In 2024 we verified Knuth's axioms in Lean, as part of a larger project

My advisor Marijn Heule had the brilliant idea of using arbitrary orderings

I proved that if this wasn't equivalent to Knuth's axioms, there would be a counterexample with 5 points

There's no counterexample with 5 points (proved via SAT)

Realizability Problem

Suppose we solve the formula and obtain:

$$\begin{aligned} A_{1,2,3} &\mapsto \text{true} \\ A_{1,2,4} &\mapsto \text{false} \\ A_{1,3,4} &\mapsto \text{true} \\ A_{1,2,5} &\mapsto \text{false} \\ &\vdots \\ A_{14,15,16} &\mapsto \text{false} \end{aligned}$$

Now what?

It's complete for $\exists\mathbb{R}$ to see if a pointset can actually *realize* these orientations.

Realizability Problem

Previous work (CICM 2024):
Pose a non-linear program to a commercial solver (LocalSolver)

$$
\begin{aligned}
\textbf{maximize} \quad & z \\
\textbf{subject to} \quad & z \leq \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}, \\
& \varepsilon < \sigma^\star(a, b, c) \cdot [(y_c - y_a)(x_b - x_a) - (x_c - x_a)(y_b - y_a)], \\
& 0 \leq x_a \leq K, \\
& 0 \leq y_a \leq K,
\end{aligned}
$$

Realizability Problem

Previous work (CICM 2024):
Pose a non-linear program to a <u>commercial</u> solver (LocalSolver)

$$
\begin{array}{rl}
\textbf{maximize} & z \\
\textbf{subject to} & z \leq \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}, \\
& \varepsilon < \sigma^\star(a,b,c) \cdot [(y_c - y_a)(x_b - x_a) - (x_c - x_a)(y_b - y_a)], \\
& 0 \leq x_a \leq K, \\
& 0 \leq y_a \leq K,
\end{array}
$$

Very slow, handled dozens of points only

Localizer

We introduce a new open-source tool, **Localizer** (local search + realizer)

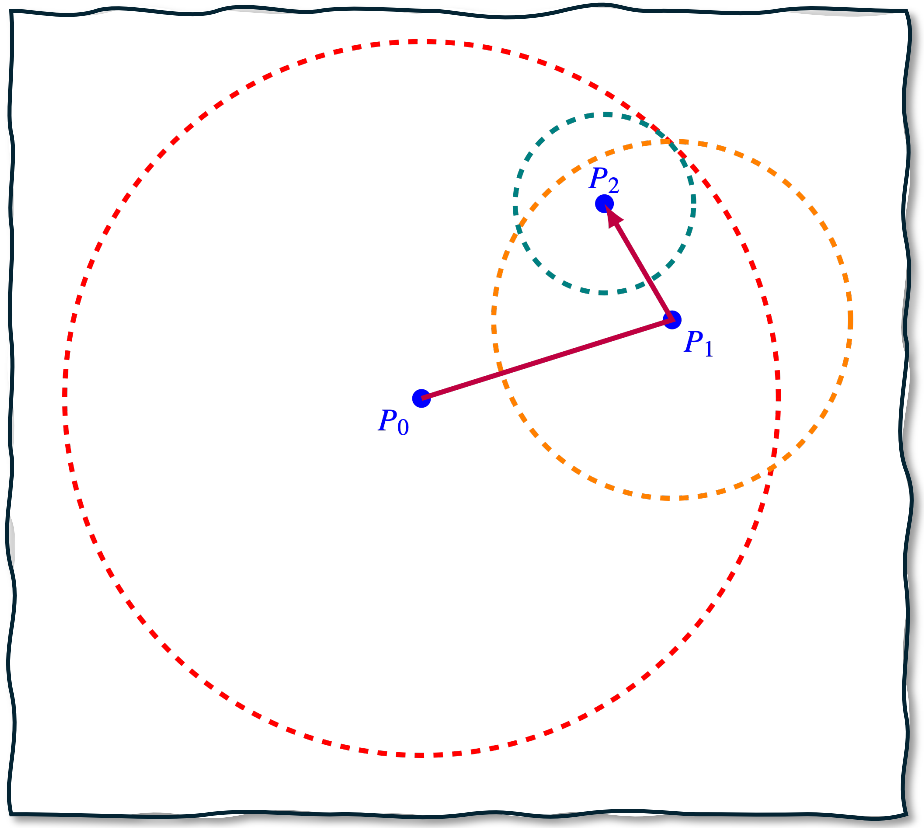
Count violated constraints, and move a point randomly to reduce this count

Localizer

We introduce a new open-source tool, **Localizer** (local search + realizer)

Count violated constraints, and move a point randomly to reduce this count

Start with aggressive moves, and decrease exponentially
(Local version of simulated annealing)



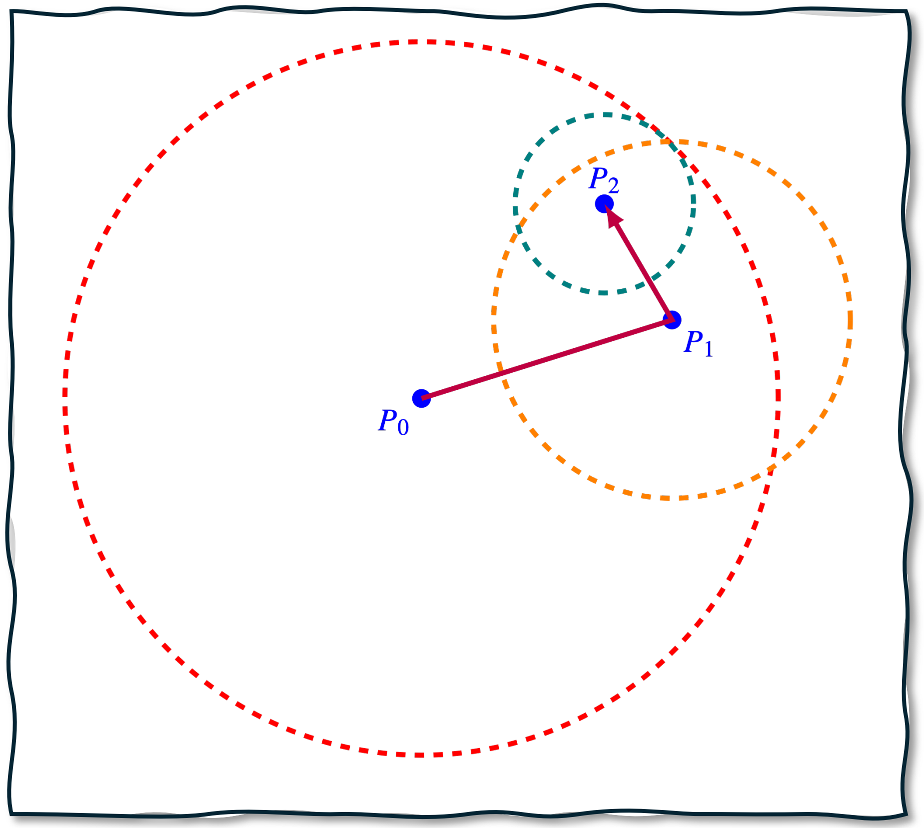
Localizer

We introduce a new open-source tool, **Localizer** (local search + realizer)

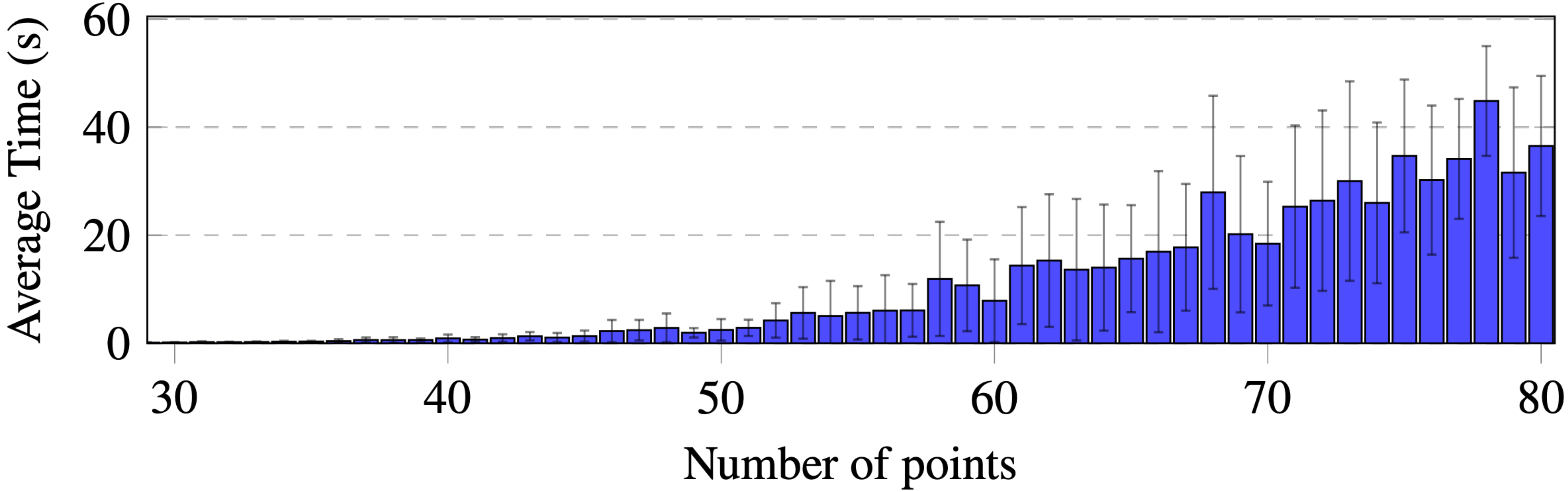
Count violated constraints, and move a point randomly to reduce this count

Start with aggressive moves, and decrease exponentially
(Local version of simulated annealing)

Multi-threaded, written in C for extreme efficiency



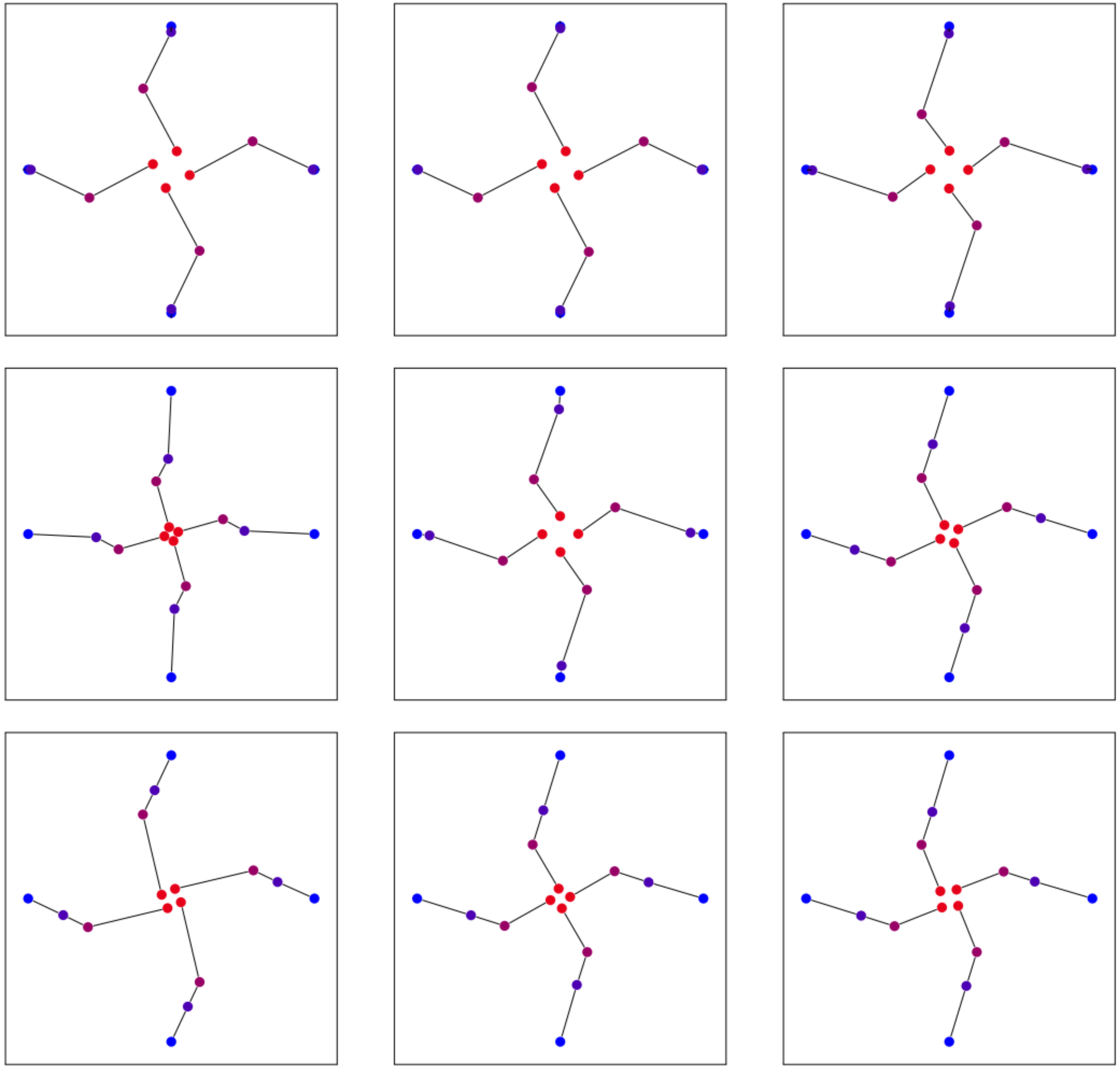
Localizer



Previous approach took hundreds of seconds for under < 20 points,
not even visible here!

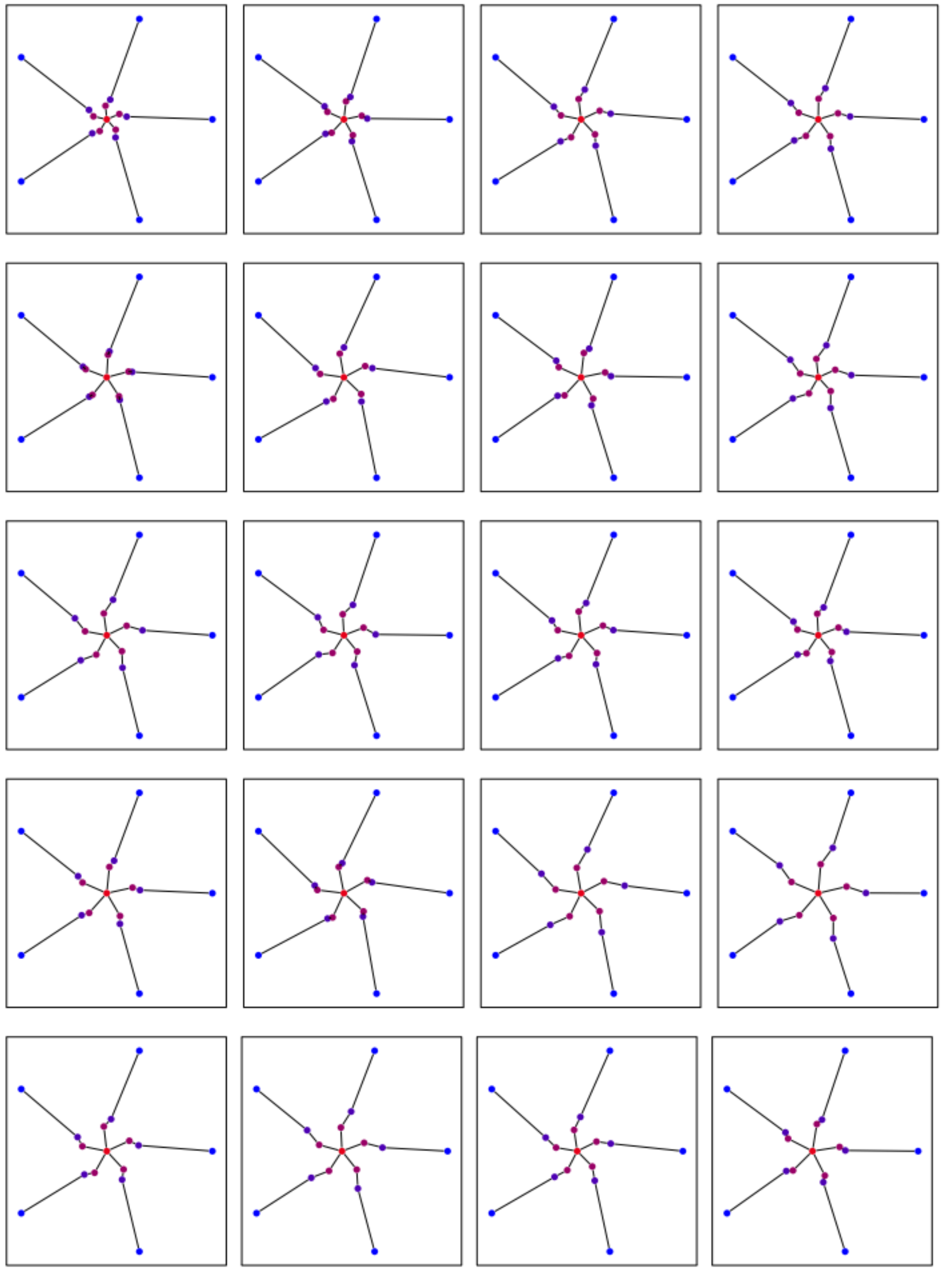
We get a complete enumeration:

9 non-isomorphic solutions with 4-fold symmetry.



We get a complete enumeration:

46 non-isomorphic solutions with 5-fold symmetry.



(etc)

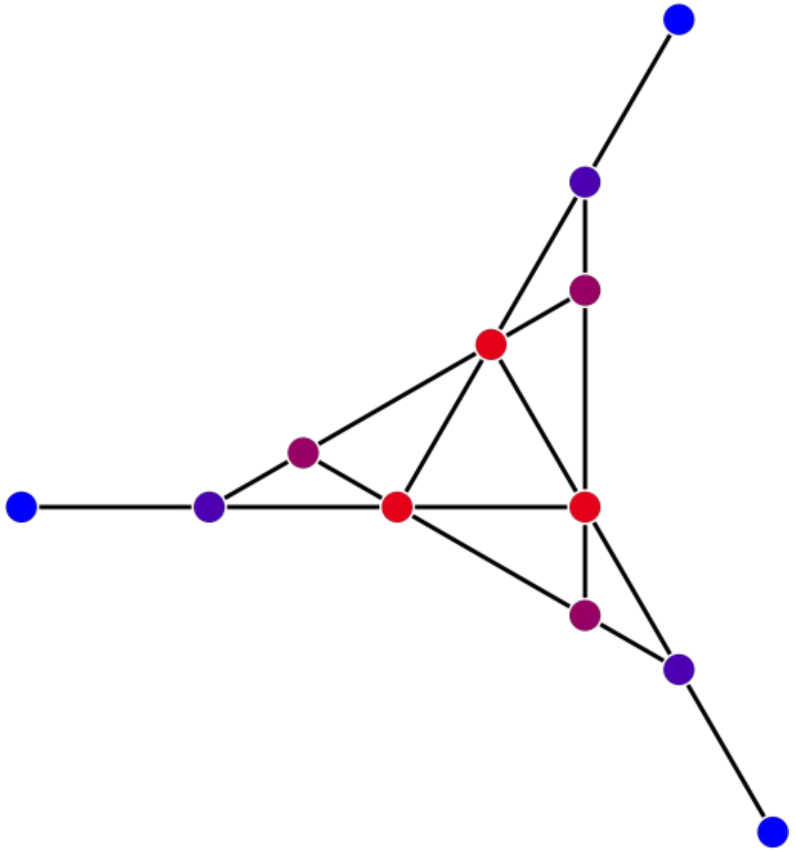
What about 32 points without a convex 7-gon?

Despite millions of solutions, even after filtering for basic isomorphisms, none of them is 4-fold symmetric, which is the expected symmetry for 32 points.

I don't have a good explanation for why this would be the case. But I'd like to know.

Everywhere Unbalanced Points

Alon's construction, circa 2000

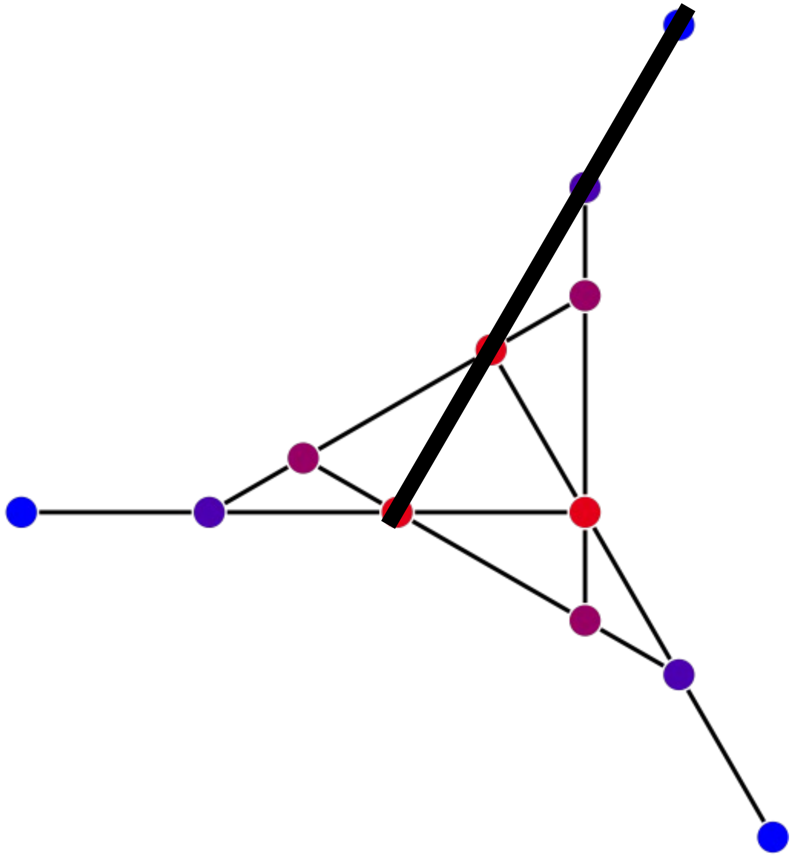


If you think of any line L between two of these 12 points, then

$|\#\text{points above } L - \#\text{points above } L| \geq 2$

Everywhere Unbalanced Points

Alon's construction, circa 2000

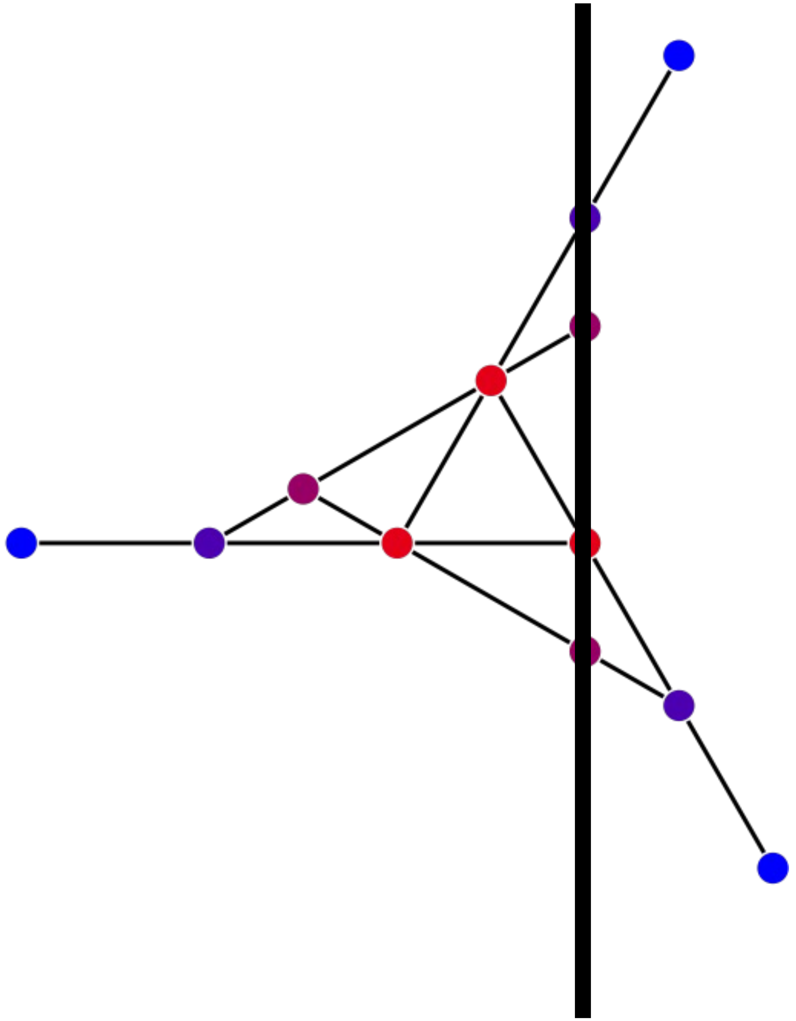


If you think of any line L between two of these 12 points, then

$|\#\text{points above } L - \#\text{points above } L| \geq 2$

Everywhere Unbalanced Points

Alon's construction, circa 2000

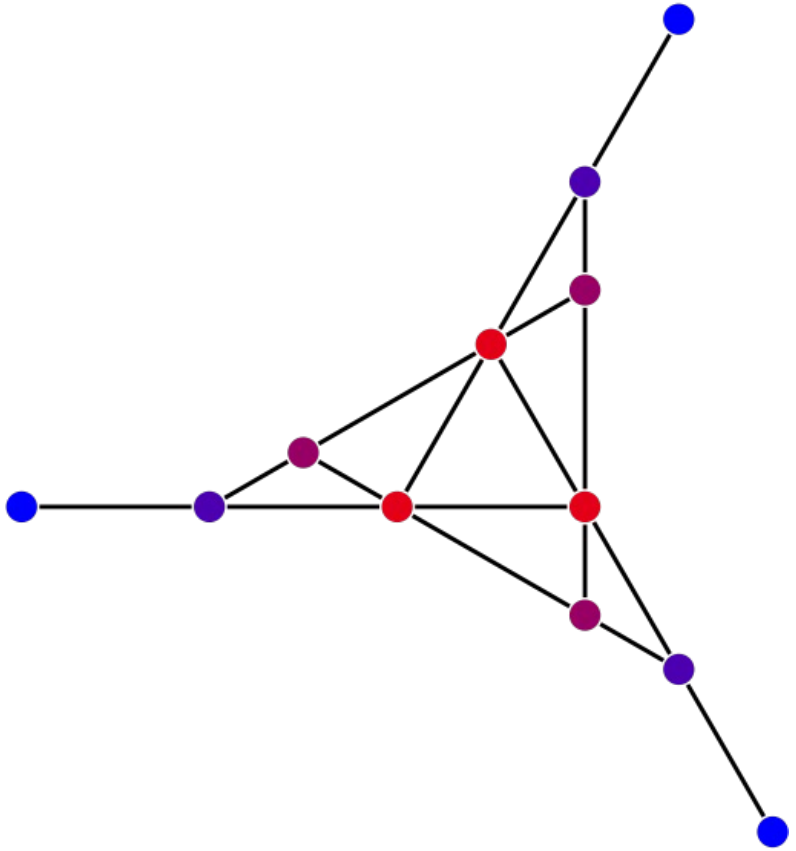


If you think of any line L between two of these 12 points, then

$|\#\text{points above } L - \#\text{points above } L| \geq 2$

Everywhere Unbalanced Points

Alon's construction, circa 2000

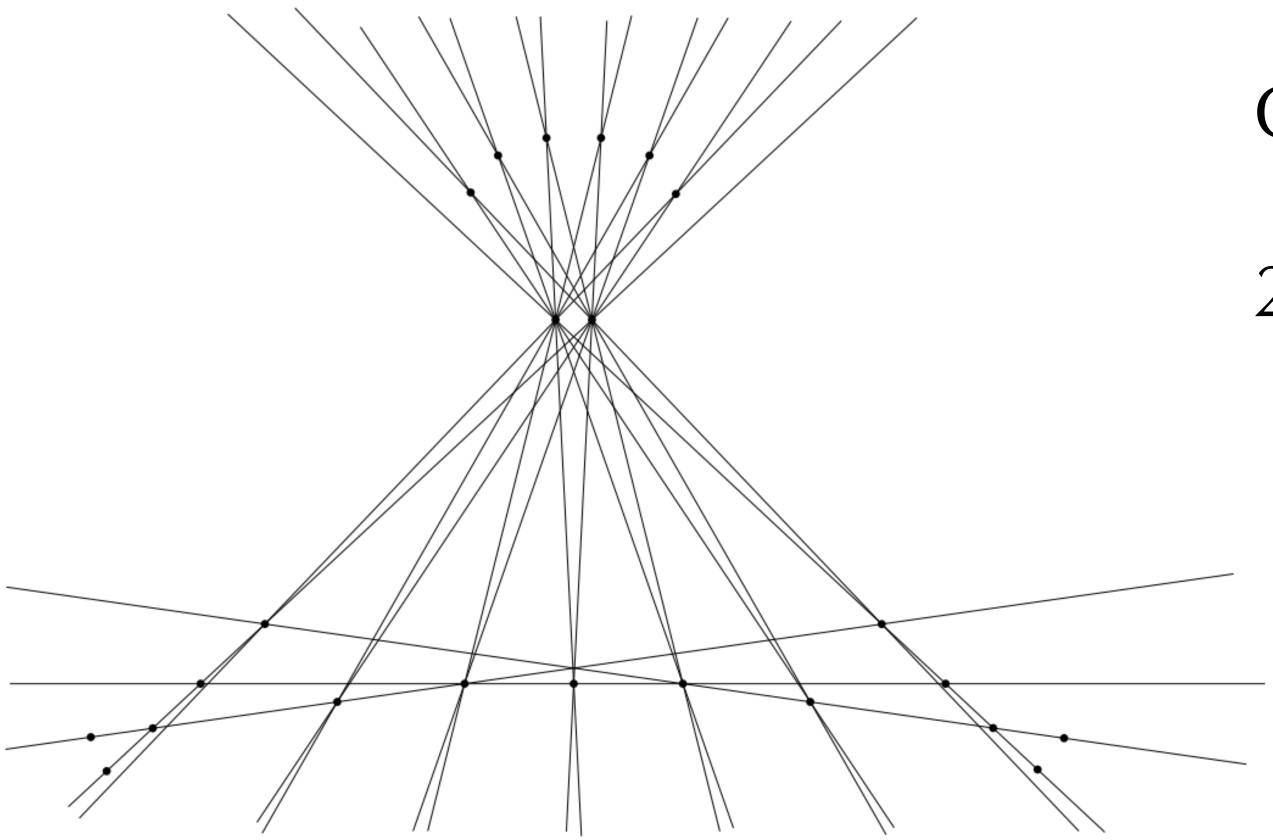


A pointset is k-EU if every line between two of its points has imbalance at least k.

Alon's construction is 2-EU, and we proved it's the smallest.

SAT encoding uses the same axioms, but now *cardinality constraints* to encode the imbalance condition.

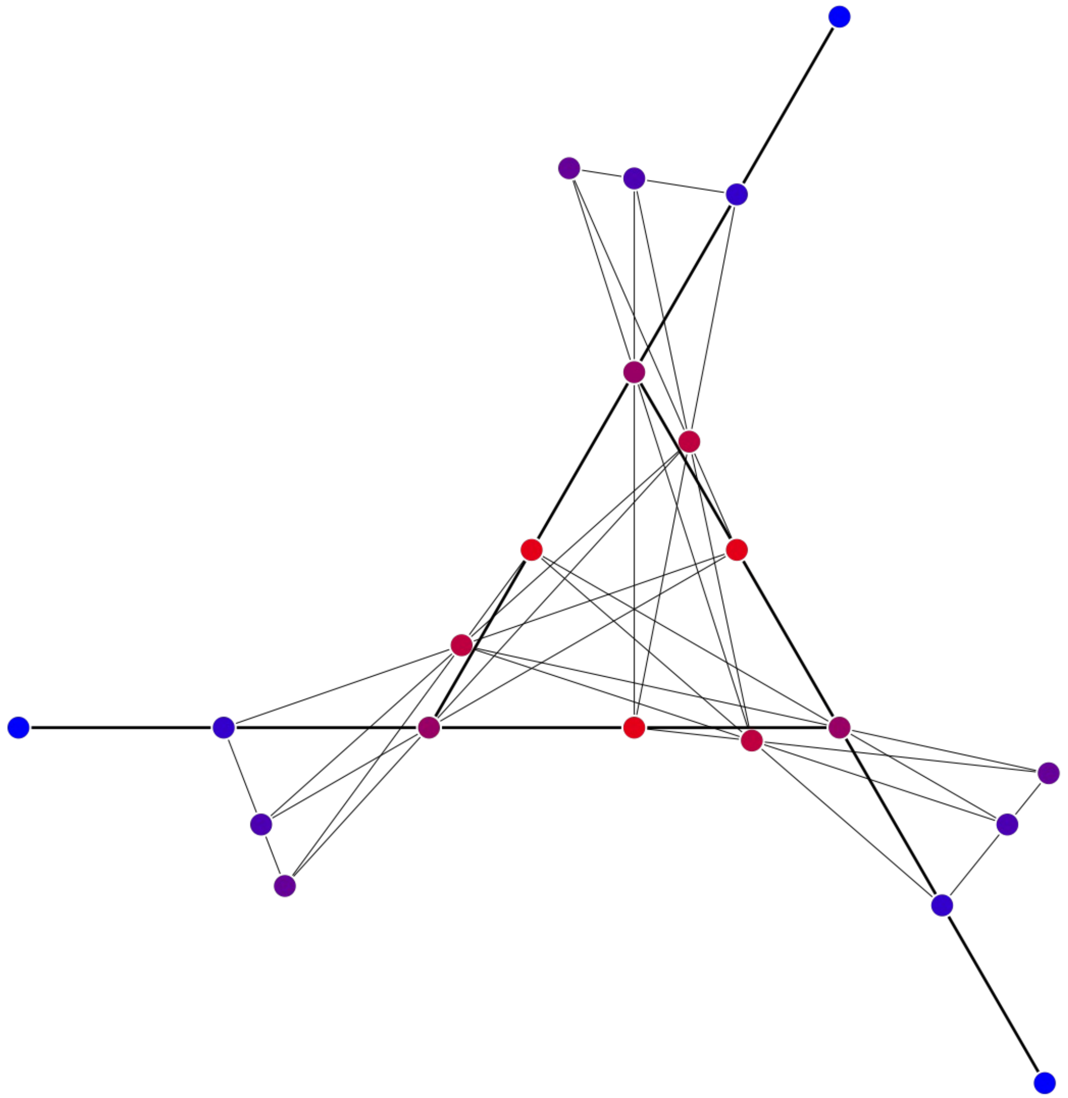
Everywhere Unbalanced Points



Conlon and Lin, 2024

23 points that are 2-EU

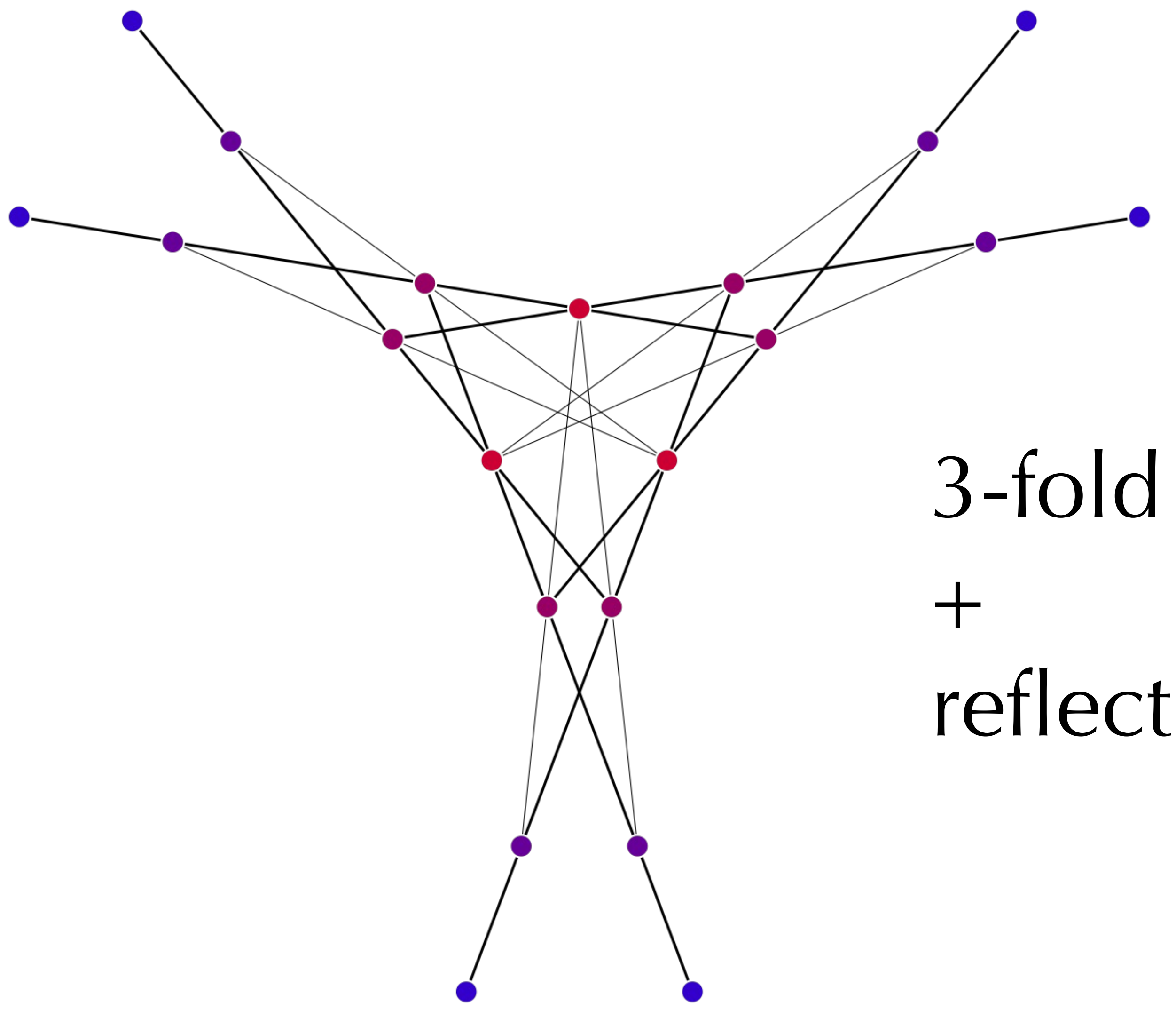
Everywhere Unbalanced Points



We find the minimum odd-size construction, with 21 points, by enforcing 3-fold symmetry.

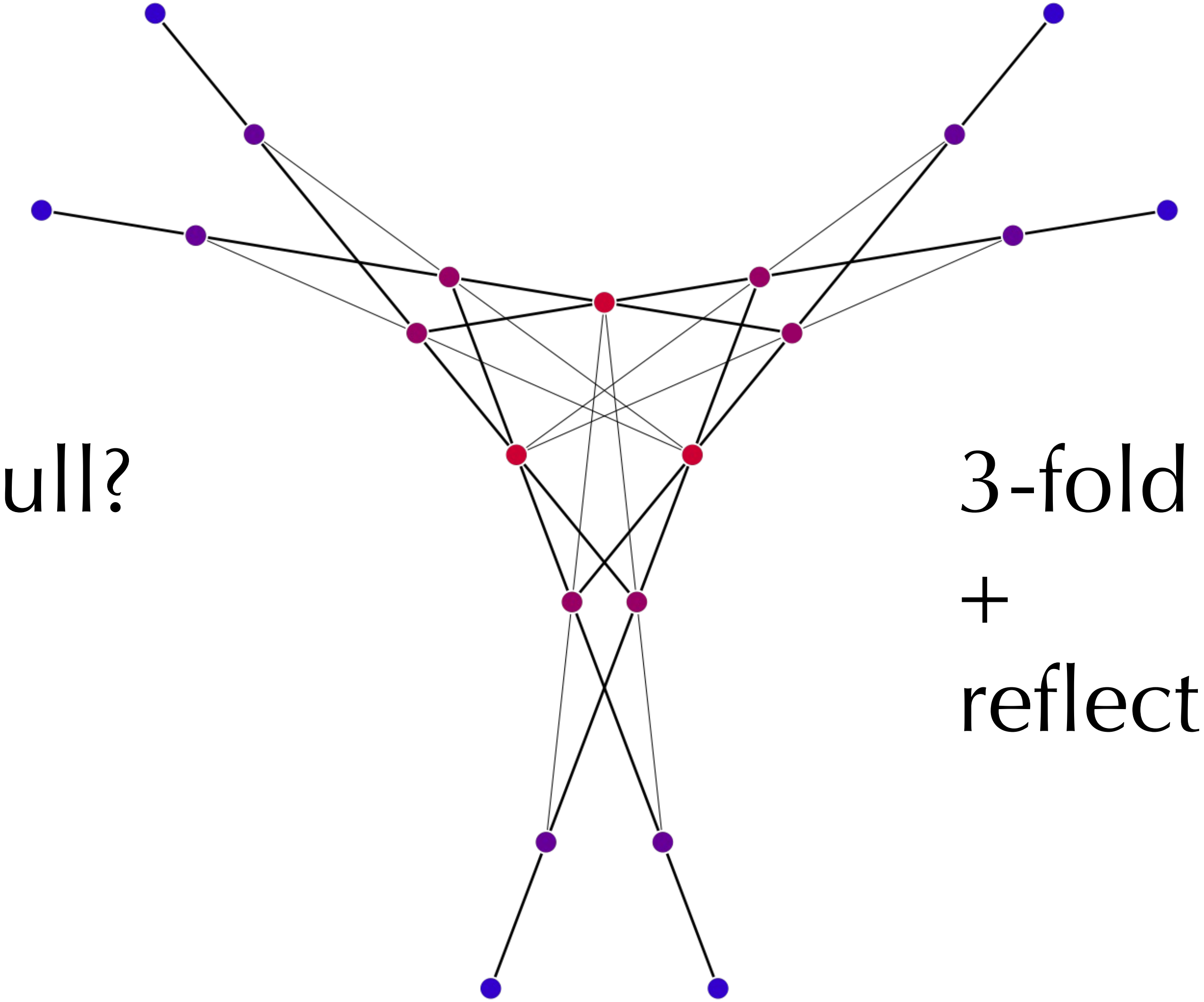
The symmetry dramatically reduces the difficulty of realizability with collinearities!

Everywhere Unbalanced Points



3-fold symmetry
+
reflection

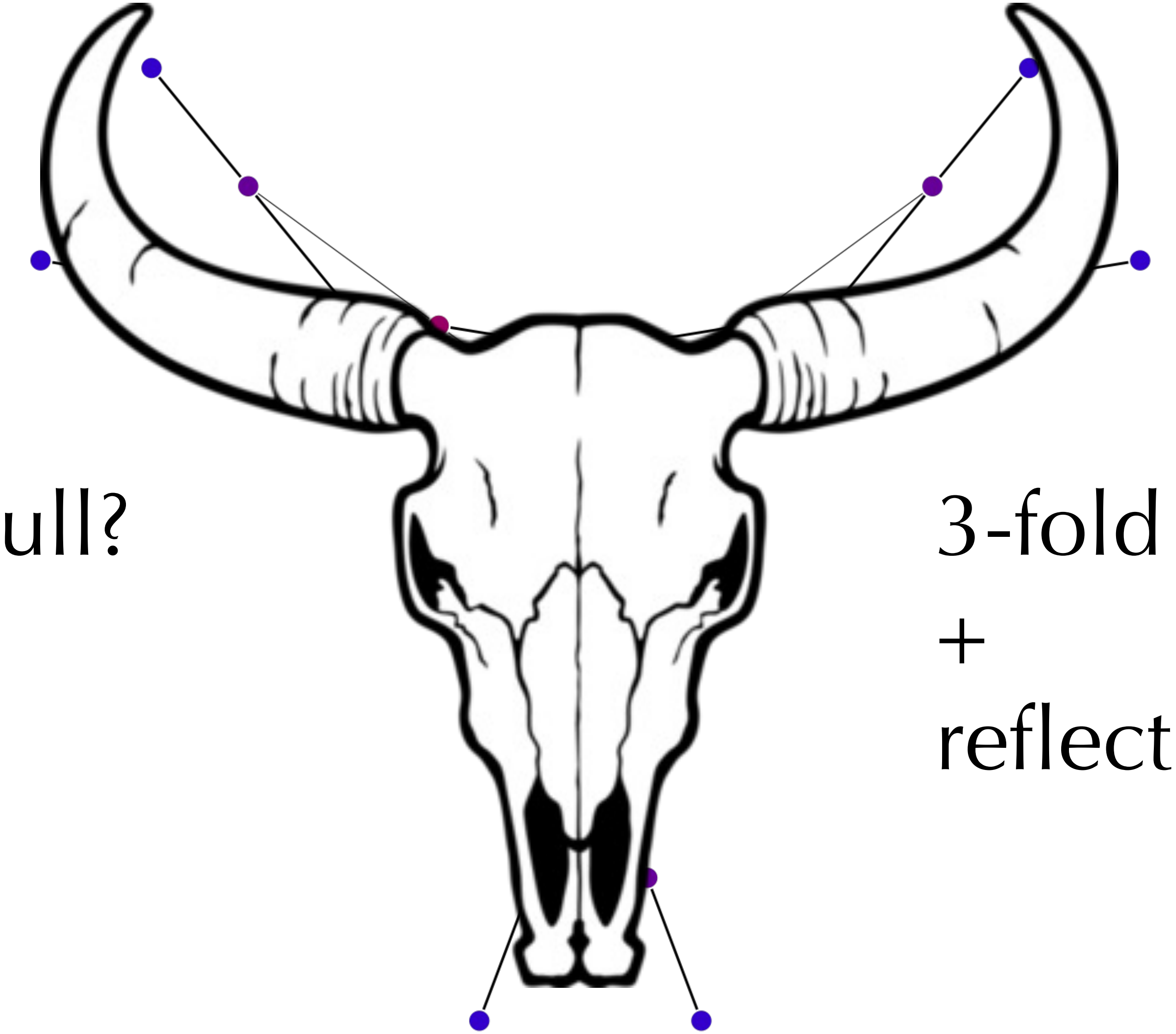
Everywhere Unbalanced Points



Bull's skull?

3-fold symmetry
+
reflection

Everywhere Unbalanced Points



Bull's skull?

3-fold symmetry
+
reflection

Automated Symmetric Constructions in Discrete Geometry

Bernardo Subercaseaux, Ethan Mackey, Long Qian, Marijn J. H. Heule

Carnegie Mellon University