Graph-Theoretic Solutions to Computational Geometry Problems

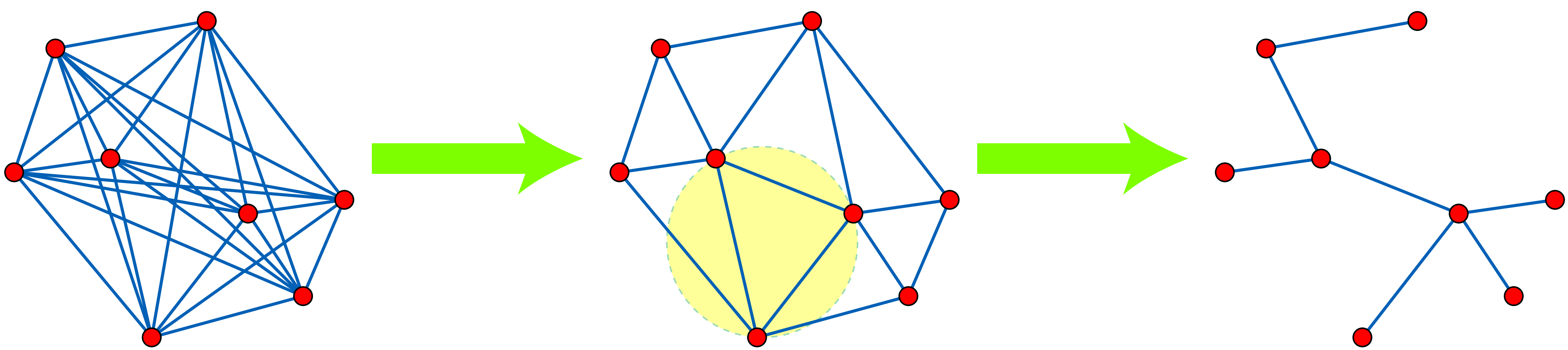
David Eppstein
Univ. of California, Irvine
Computer Science Department

Historically, many connections from graph-theoretic algorithms to computational geometry...

1. Geometric analogues of classical graph algorithm problems

Typical issue: using geometric information
to speed up naive application of graph algorithms

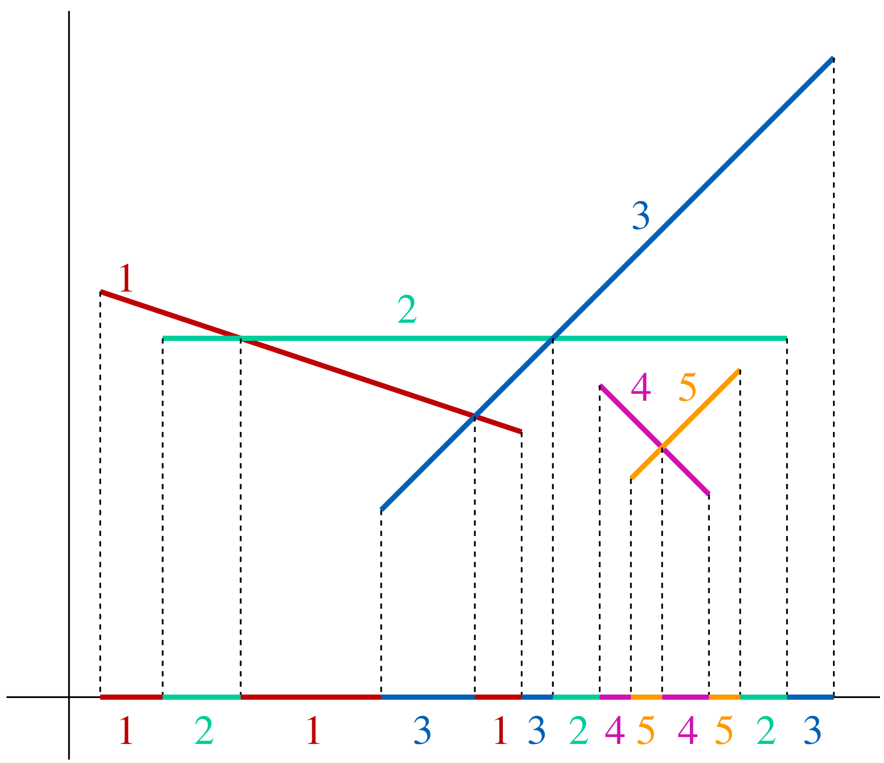
E.g., Euclidean minimum spanning tree
= Spanning tree of complete graph with Euclidean distances
Solved in O(n log n) time by Delaunay triangulation [Shamos 1978]



Historically, many connections from graph-theoretic algorithms to computational geometry...

2. Geometric approaches to graph-theoretic problems

How many different minimum spanning trees
can a graph with linearly varying edge weights form?
$O(m\, n^{1/3})$ via crossing number inequality [Dey, DCG 1998]
$\Omega(m\, \alpha(n))$ via lower envelopes of line segments [E., DCG 1998]



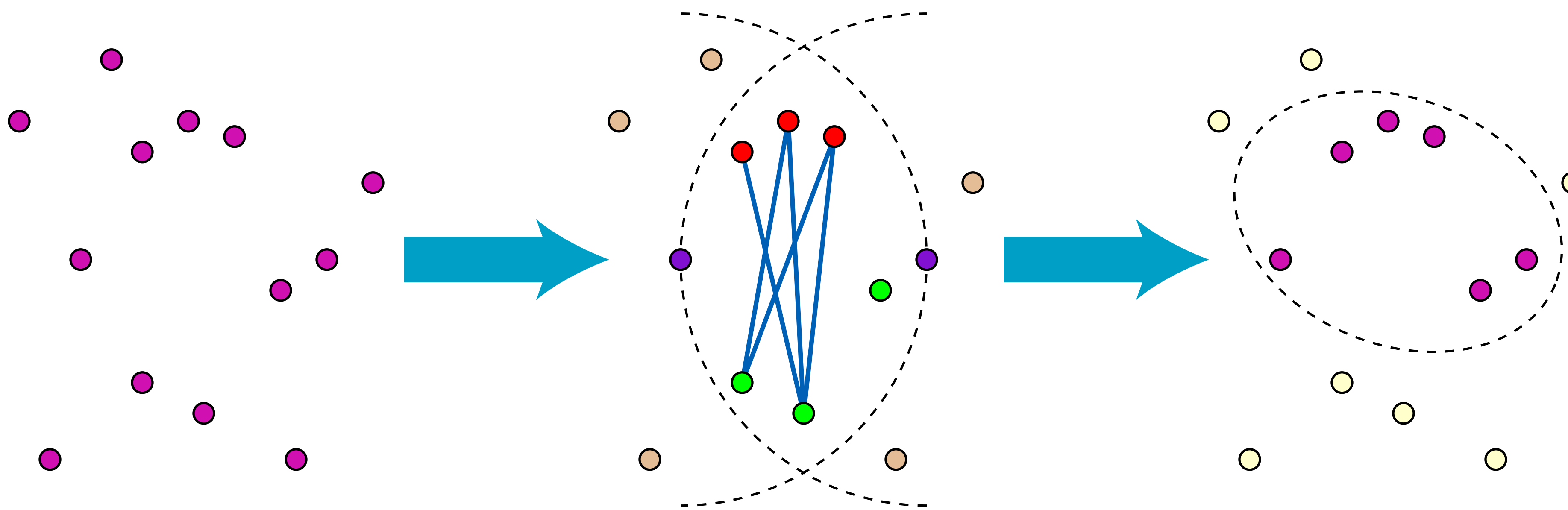
Historically, many connections from graph-theoretic algorithms to computational geometry...

Today: 3. Graph-theoretic approaches to geometric problems

Geometry leads to auxiliary graph

Special properties of auxiliary graph lead to algorithm

Algorithm on auxiliary graph leads to solution



Minimum-diameter clustering via maximum independent sets in bipartite graphs (more detail later in talk)

Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

Mesh stripification

Angle optimization of tilings

Metric embedding into stars

Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

Mesh stripification

Angle optimization of tilings

Metric embedding into stars

The Art Gallery Problem

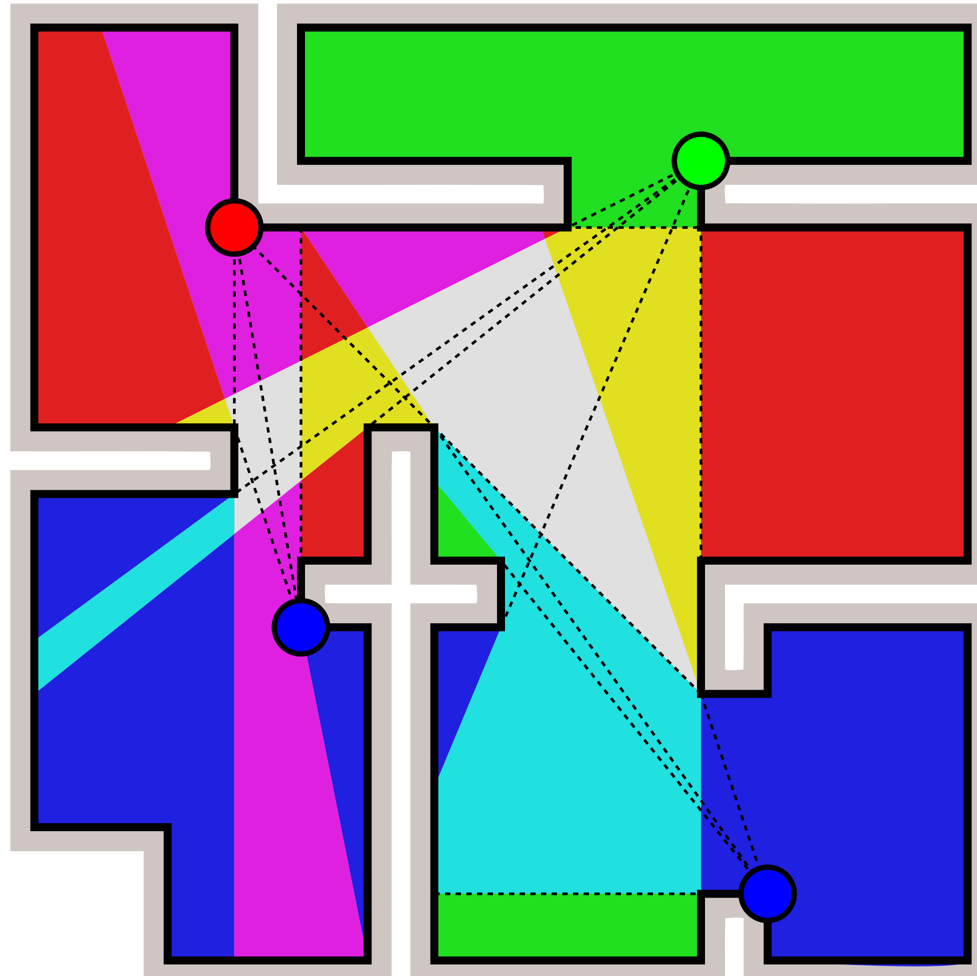
Input: a **simple polygon**
(no holes, no self-crossings),
the floor plan of an art gallery

Output: a **small set of points**
(places for guards to stand)
from which whole gallery visible

Exact optimization is NP-hard

Approximation algorithms known

Today: what is worst-case #guards
as a function of gallery complexity?



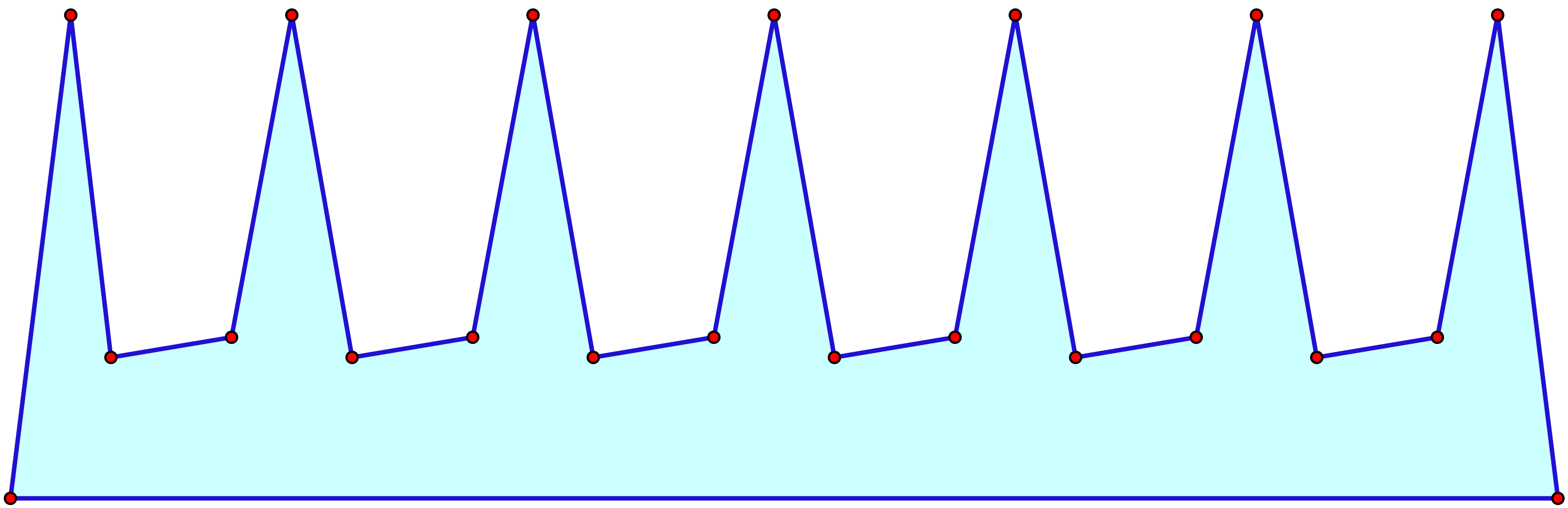
This art gallery can be guarded from four points

Claudio Rocchini, GFDL image on Wikimedia commons,
http://commons.wikimedia.org/wiki/File:Art_gallery_problem.svg

Chvátal's Art Gallery Theorem [Chvátal, JCTB 1975]

Every n-vertex simple polygon requires at most **floor(n/3) guards**

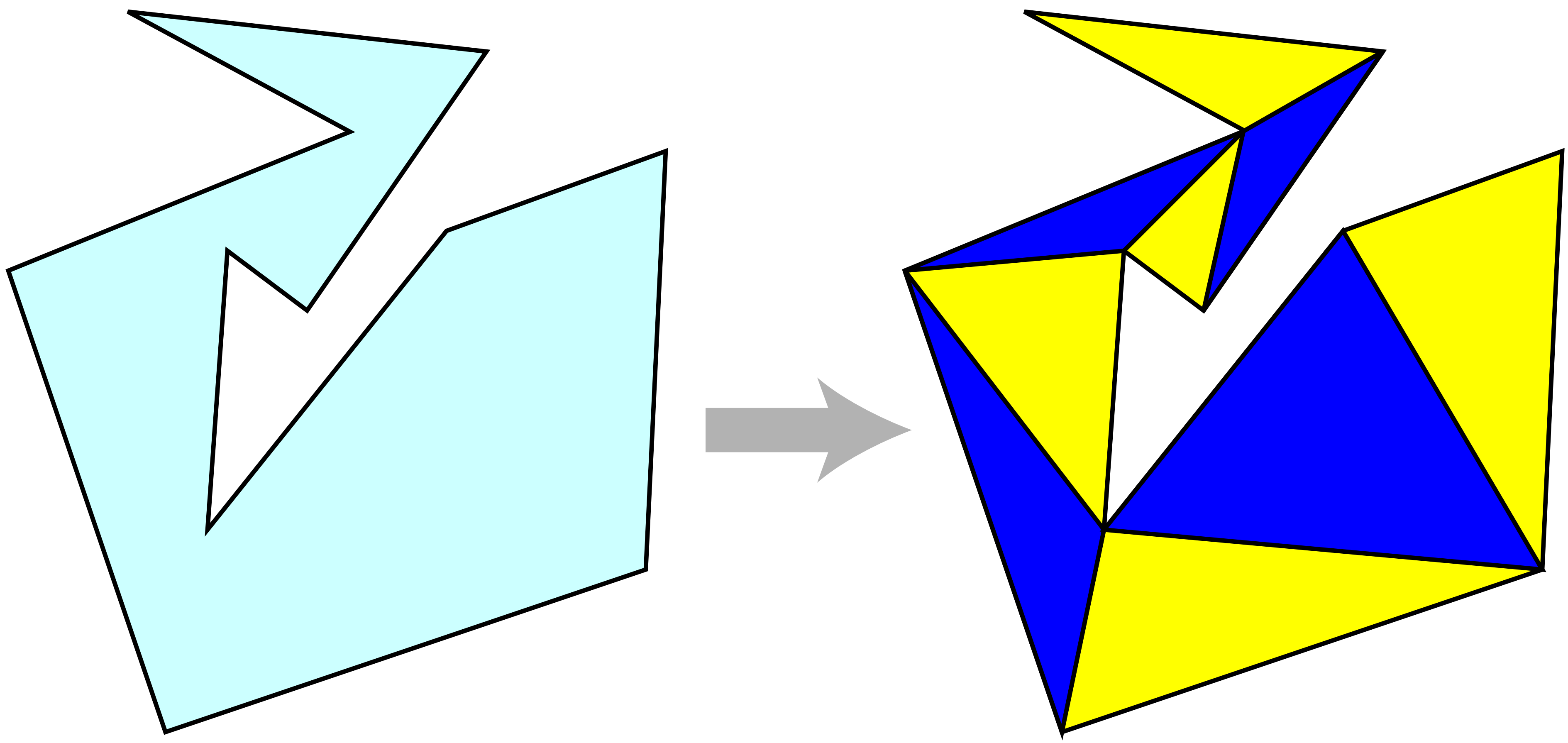
For every n ≥ 3, some simple polygons require exactly floor(n/3) guards



Each guard can see at most one tooth of the comb

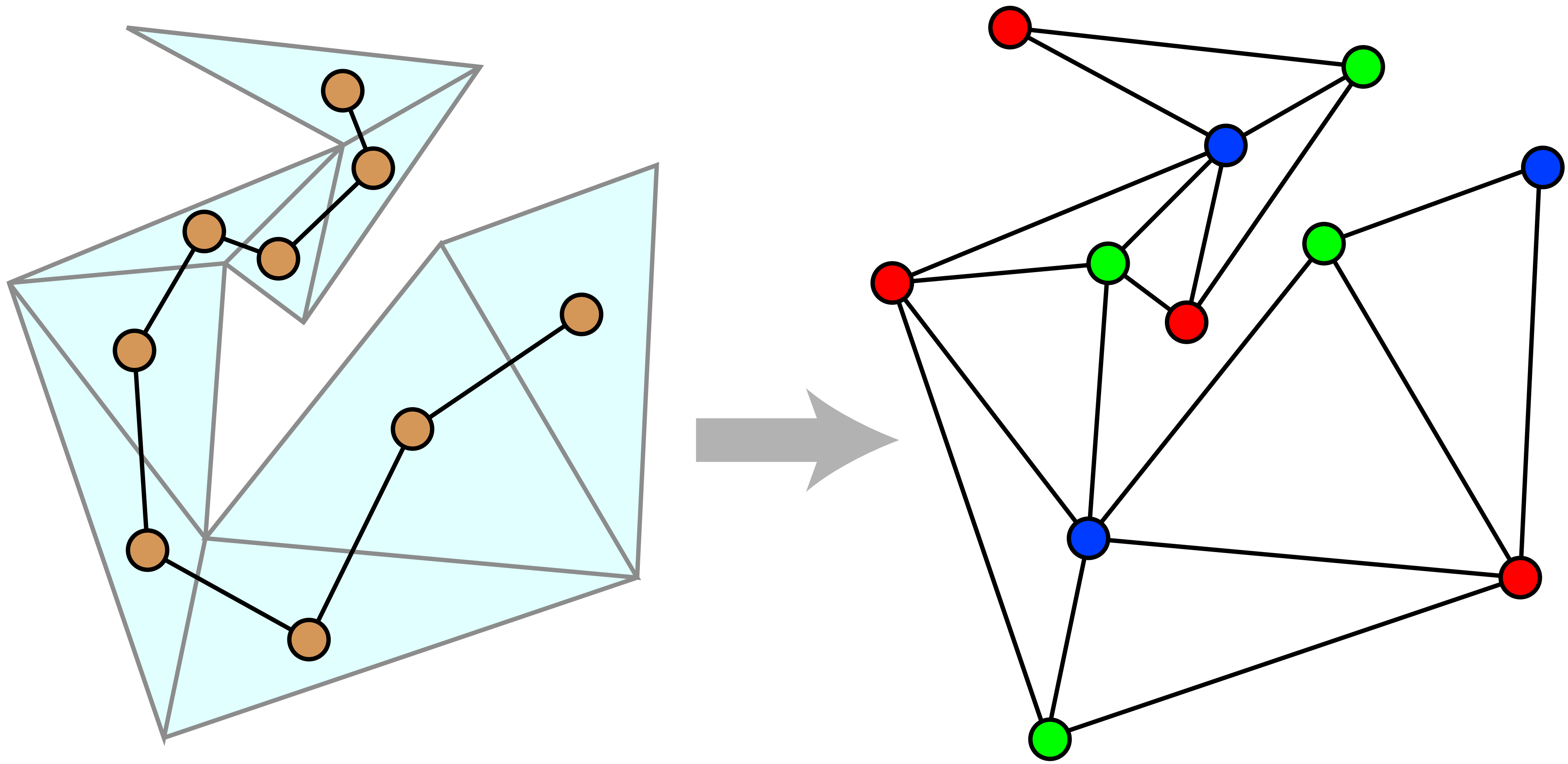
Fisk's Proof of the Art Gallery Theorem [Fisk, JCTB 1978]

I. Triangulate the polygon



Fisk's Proof of the Art Gallery Theorem [Fisk, JCTB 1978]

II. 3-color the (maximal outerplanar) graph of the triangulation

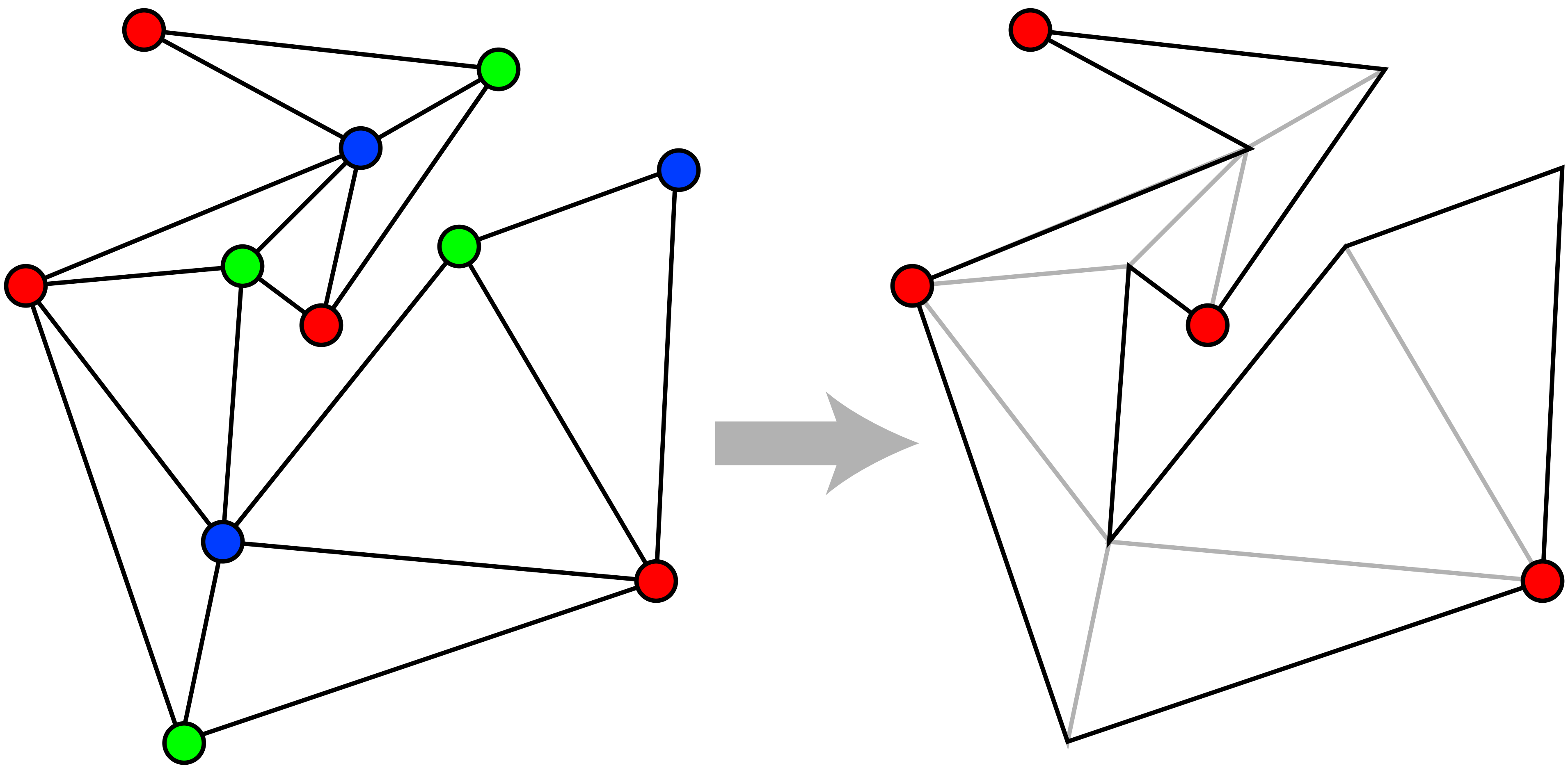


Dual graph is a tree

Remove a leaf (degree-two vertex of triangulation), recurse

Fisk's Proof of the Art Gallery Theorem [Fisk, JCTB 1978]

III. Select the vertices with the least-frequently-used color



The points within any triangle can be seen from the triangle vertex of the selected color

Algorithmic implications

Graph coloring is hard in general, but...

Finding a set of floor(n/3) guards can be performed in **linear time**

Linear time triangulation of simple polygons [Chazelle, DCG 1991]

Linear time optimal coloring of maximal outerplanar graphs

(easy using greedy coloring:
maximal outerplanar graphs are chordal,
chordal graphs are perfectly orderable)

Orthogonal Art Galleries

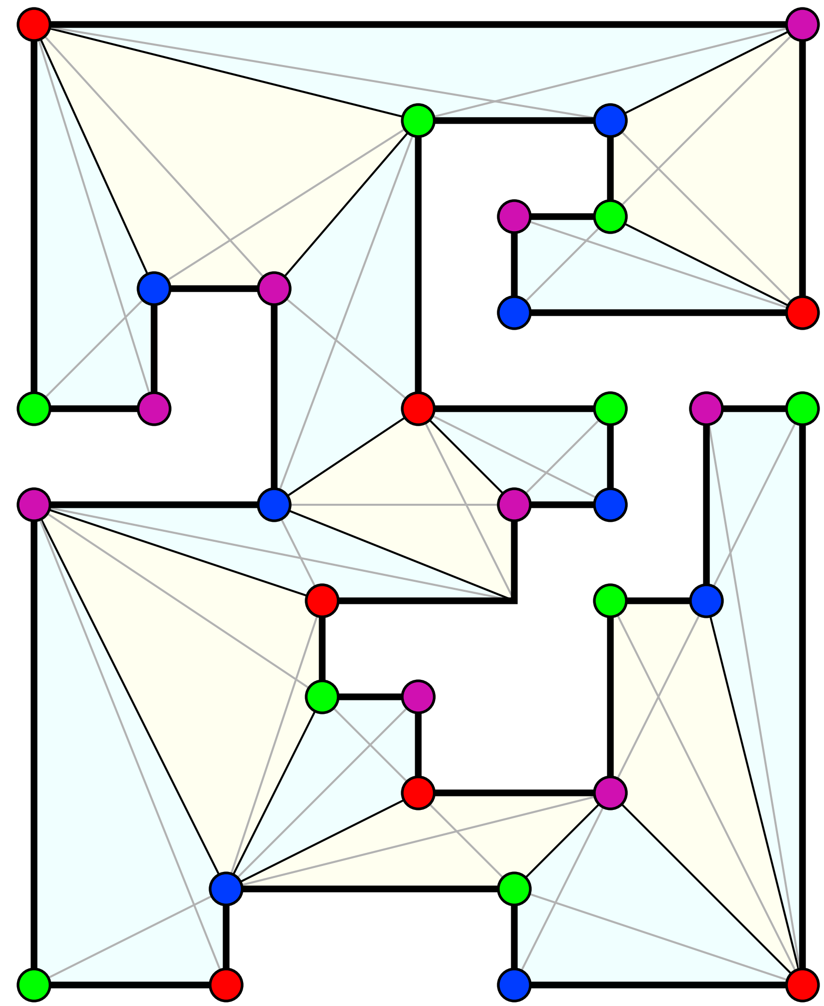
At most floor(n/4) guards needed
[Kahn, Klawe, Kleitman, SIAM ADM 1983]

Partition into convex quadrilaterals
(non-trivial)

Squaregraph: planar graph,
all interior faces quadrilaterals,
all interior vertex degrees ≥ 4
(bipartite, median graph)

Kinggraph: add diagonals of quads
Guaranteed to be 4-chromatic

4-color, use smallest color class



Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

Mesh stripification

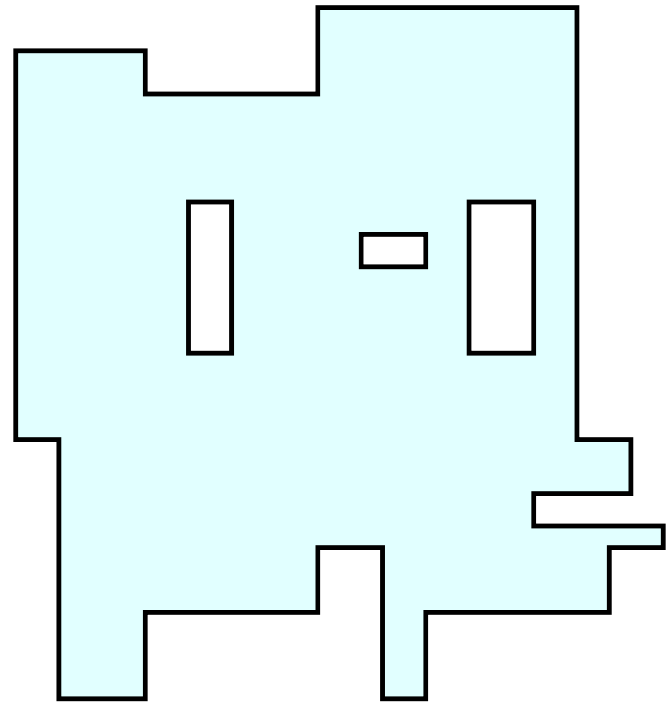
Angle optimization of tilings

Metric embedding into stars

Rectangle partition

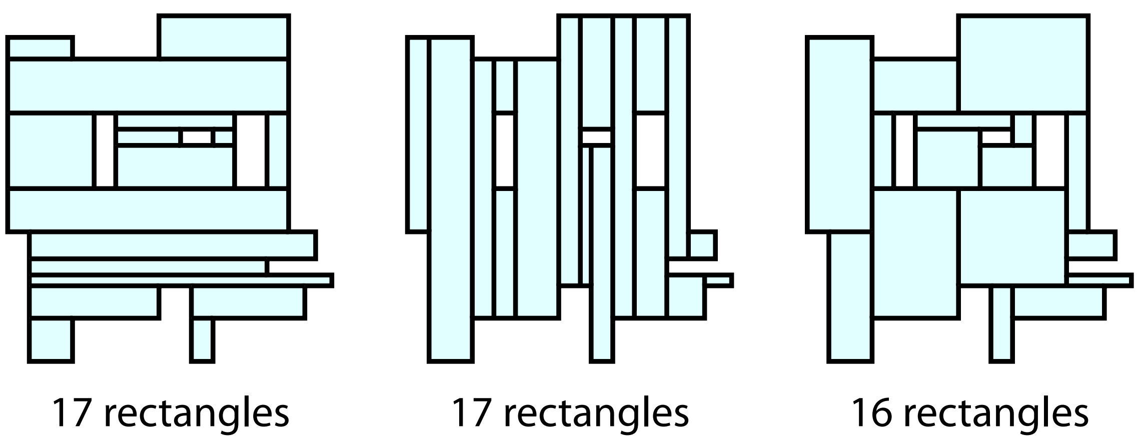
Input: orthogonal polygon
(sides parallel to axes,
possibly with holes)

Output: partition into
minimum # of rectangles



Applications include

- Bitmap data compression
- VLSI mask fabrication
- DNA array design
- Reconnaissance planning



Rectangle partition

Input is partitioned into rectangles

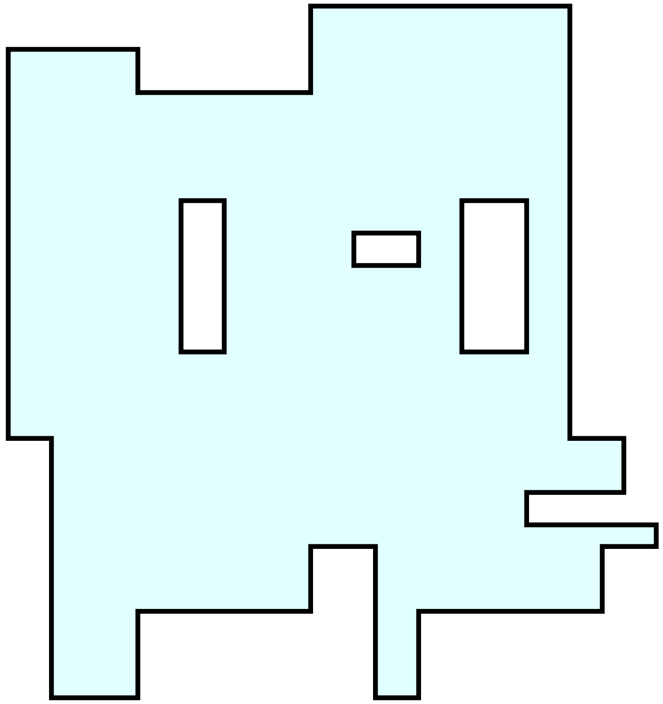
if and only if

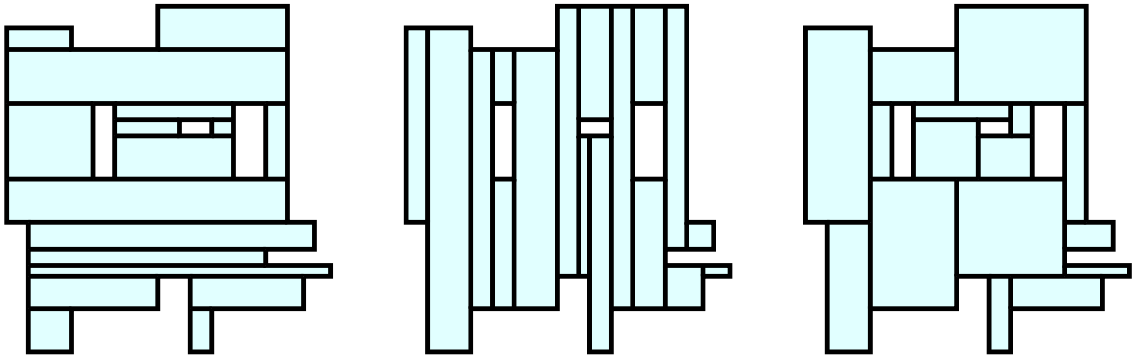
Each concave vertex is adjacent to an internal segment

 # Rectangles ≥ # Segments + 1

(equality when no degree-4 internal vertices)

So # Rectangles = # Concave vertices + 1 regardless of the partition???





17 rectangles

17 rectangles

16 rectangles

Key observation

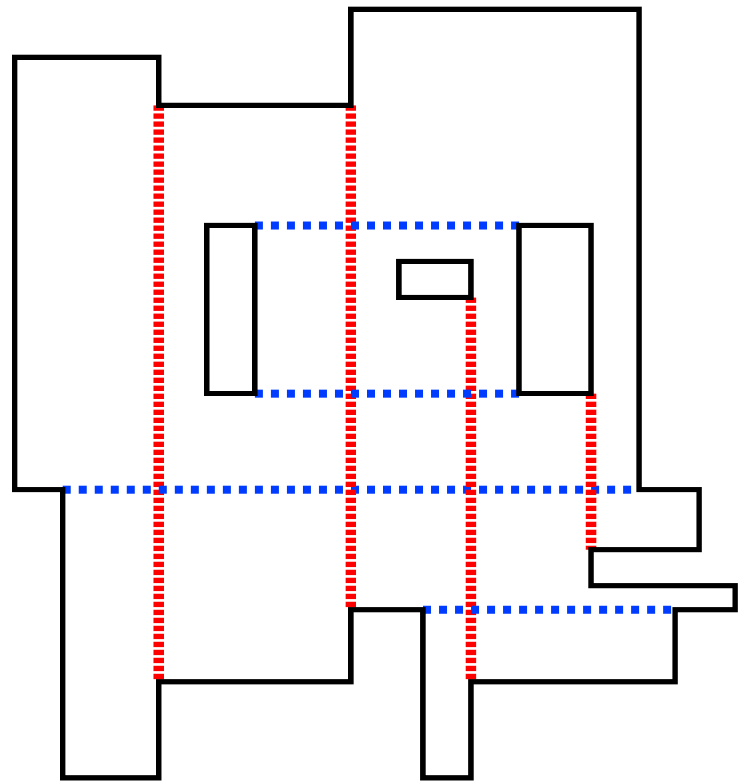
[Lipski et al., Fund. Inf. 1979;
Ohtsuke, ISCAS 1982;
Ferrari et al., CVGIP 1984]

Some segments can cover
two concave vertices at once

 # Rectangles =
 # Concave vertices -
 # Two-vertex segments + 1

Problem becomes one of finding
maximum non-intersecting set
of two-vertex segments

(Shared endpoint counts as intersection)



König's theorem

[D. König, Mat. Fiz. Lapok 1931]

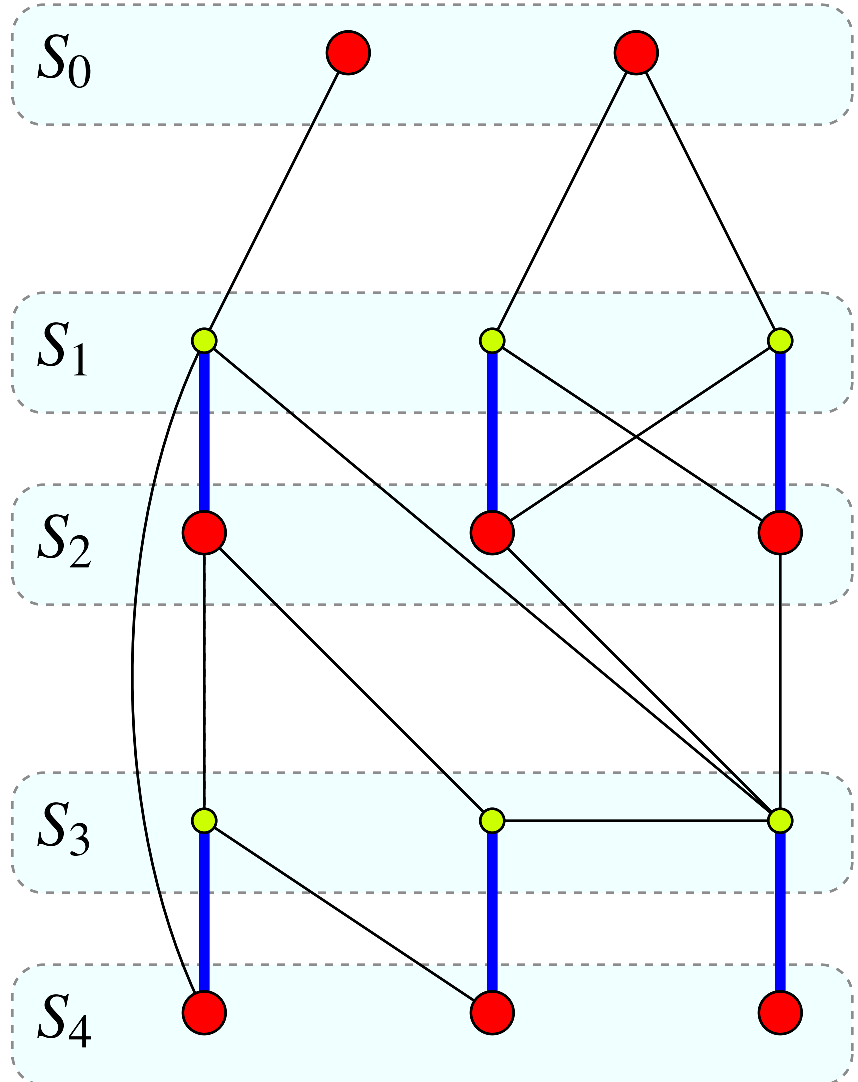
In a bipartite graph,
independence number = n - |M|
where M is a maximum matching

Upper bound:

For arbitrary graphs,
MIS ≤ # vertices – # match
(MIS can only use one vertex
from every matched pair)

Lower bound:

Even levels of alternating path
decomposition starting from
unmatched vertices form a large
independent set



Rectangle partition algorithm

Find line segments that could cover two concave polygon vertices

Form their (bipartite) intersection graph

Use matching algorithms to find a maximum independent set

Add additional line segments to cover the remaining concave vertices

Time (using geometric data structures to speed up matching steps):

$O(n^{3/2} \log n)$

[Lipski, Networks 1983 & IPL 1984; Imai & Asano, SICOMP 1986]

Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

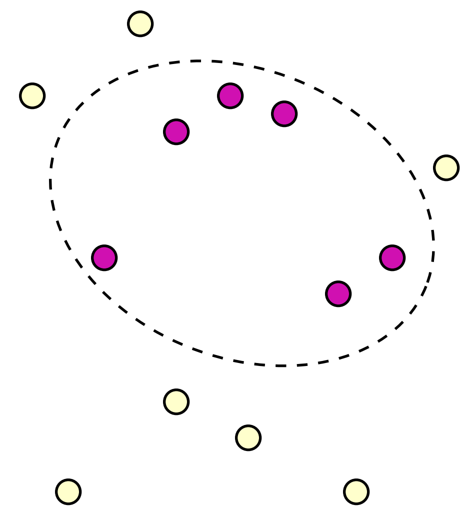
Mesh stripification

Angle optimization of tilings

Metric embedding into stars

Minimum diameter clustering problems

[Aggarwal et al, J. Algorithms 1991; E. & Erickson, DCG 1994]



Diameter = max distance among points

Given n points ...

Find k of them with minimum diameter

Find as many as possible with diameter D

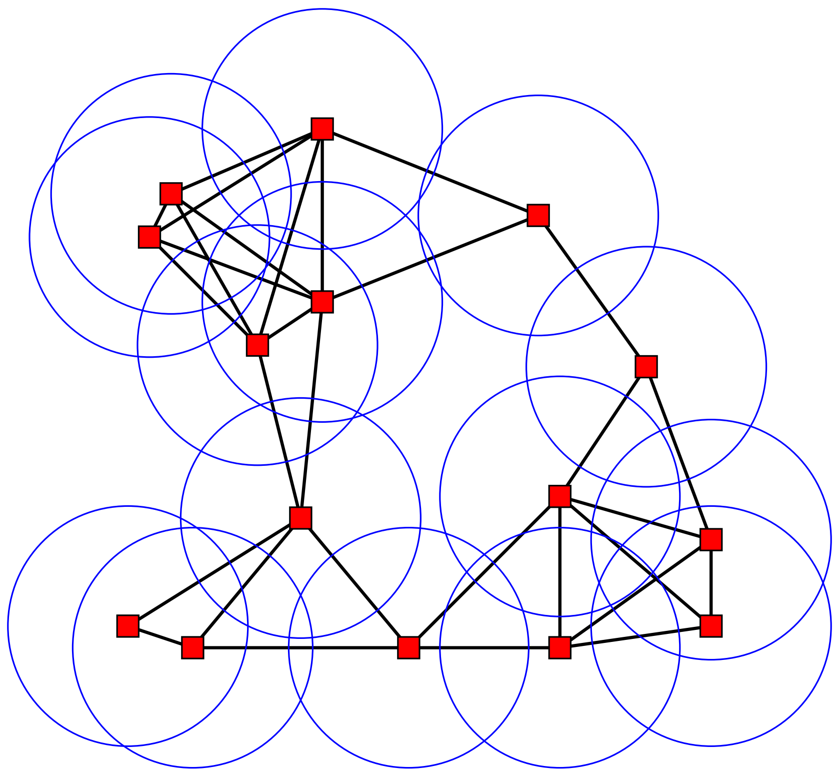
Test whether there exists a subset of k points with diameter D

Since there are $O(n^2)$ possible diameters, these are all equivalent to each other (up to log factors) via binary search

Easy (but unhelpful) reduction to a graph problem

Maximizing #points in a diameter-D cluster
= finding a maximum clique in a two-dimensional unit disk intersection graph

Can be solved in polynomial time
[Clark, Colbourne, & Johnson, Disc. Math. 1990]
but translation into a graph is too direct to provide insight into solution



More insightful reduction

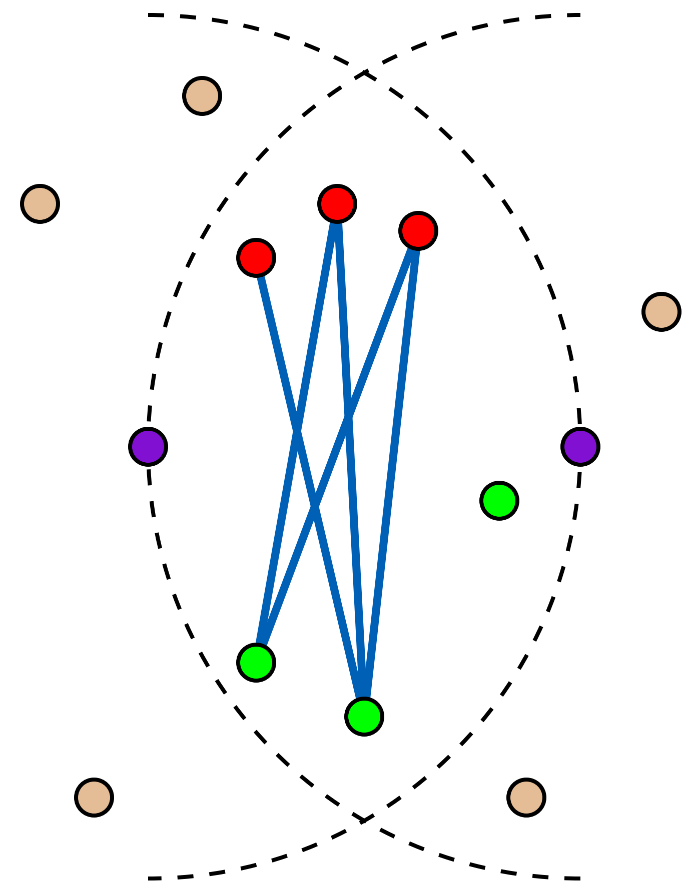
[Clark et al, Aggarwal et al]

For each pair of points p,q that might be a diameter (purple):

The points within dist(p,q) of both p and q form a lune (intersection of two circles)

Any incompatible pair of points (too far apart to be in cluster) belong to opposite half-lunes

Max # points having pq as their diameter = max ind. set of bipartite graph



Dynamic graph algorithm for multiple bipartite MIS's

[E. & Erickson 1994]

To find max cluster size for given diameter D:

For each input point p:
Let q be a point at distance D from p (not necessarily in input),
generate the lune from p and q, and
find maximum independent set in the bipartite graph of lune.

Rotate q continuously around p;
whenever the rotation changes the set of points within the lune:
Do a single alternating path search to update MIS

Time $O(n^3 \log n)$

To find min D given cluster size k:
Use binary search for D among input distances
Limit subproblem size via k-nearest-neighbor graph
Time $O(n \log n + k^2 n \log^2 k)$

Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

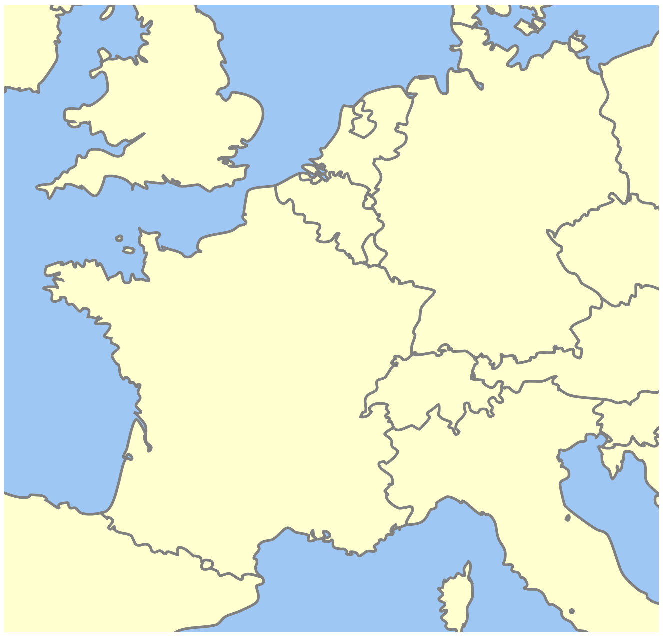
Mesh stripification

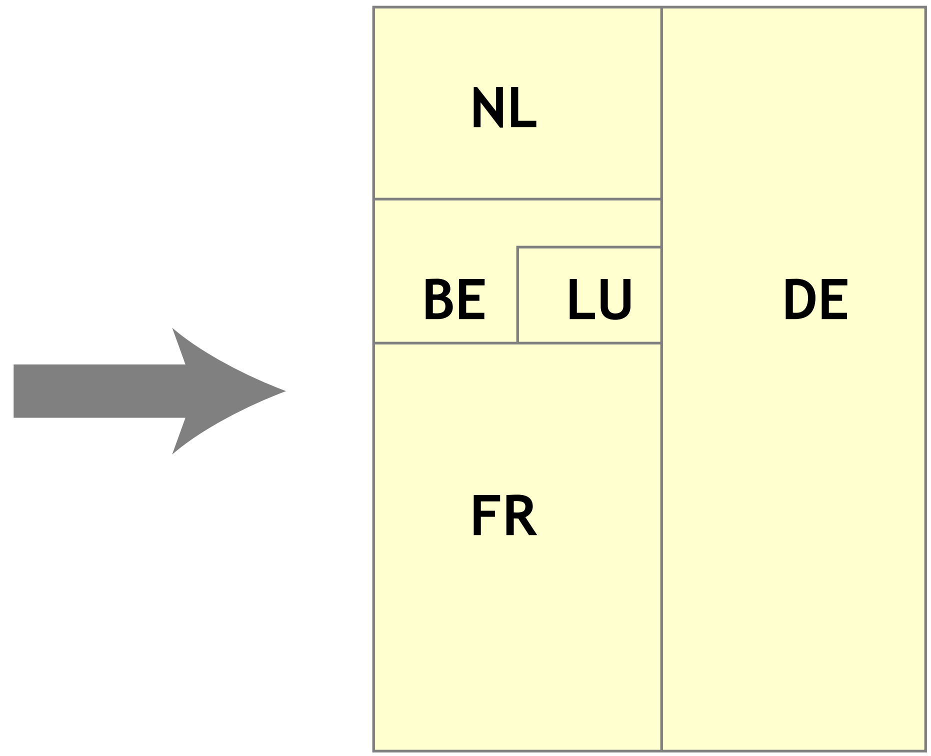
Angle optimization of tilings

Metric embedding into stars

Rectilinear cartograms

Transform map into stylized diagram
in which areas represent non-geographic data about regions





Raisz, Geog. Rev. 1934:
"It should be emphasized that the statistical cartogram is not a map."

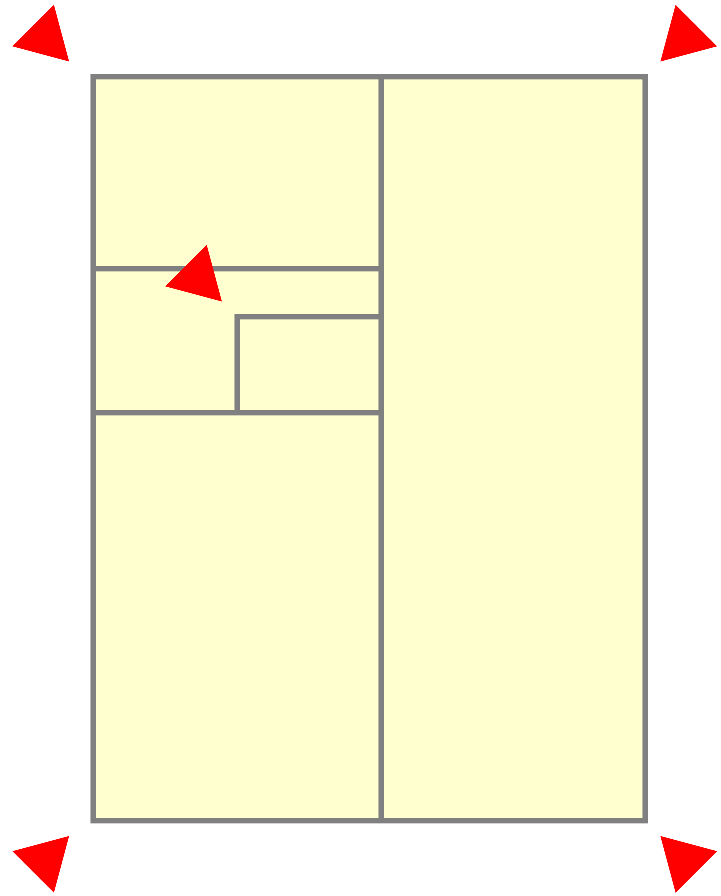
Diagram modifed from CC-BY-SA image by Brianski, Canuckguy, Zaparojdik, Madman2001, Roke, & Ssolbergj,
online at http://commons.wikimedia.org/wiki/File:Blank_Map_of_Europe_-w_boundaries.svg

Algorithmic issues in cartogram construction

Fitting given numerical quantities to areas of regions
[E. et al, SoCG 2009]

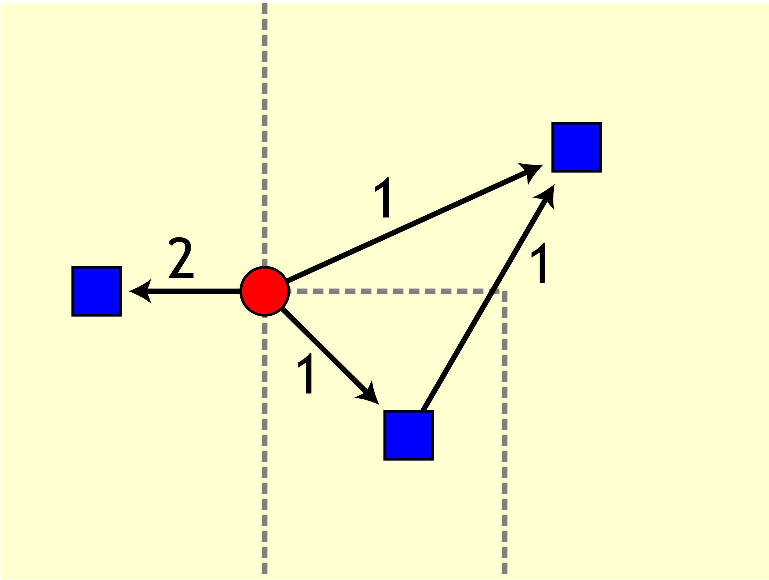
Making region adjacencies match their geographic orientations
[E. & Mumford, WADS 2009]

Today:
Minimizing # bends



Originally studied as a technique for graph drawing
[Tamassia, SICOMP 1987; Tamassia et al., Trans. Sys. Man. Cyb. 1988; Tamassia et al., SPDP 1991; Fößmeier & Kauffman, GD 1995]

Modeling bends by flow



Create a vertex in a graph
for every junction and region

One unit of flow = 90 degrees

Send four units of flow from
junctions to adjacent regions

Send flow across any bend
between two regions

Junction has four outgoing units

Region with k junctions has 4 – 2k outgoing units or 2k – 4 incoming units
(2k + 4 incoming units for the exterior region)

Bend minimization as min-cost flow

Vertex per junction and region
Additional circulation vertex (not shown)

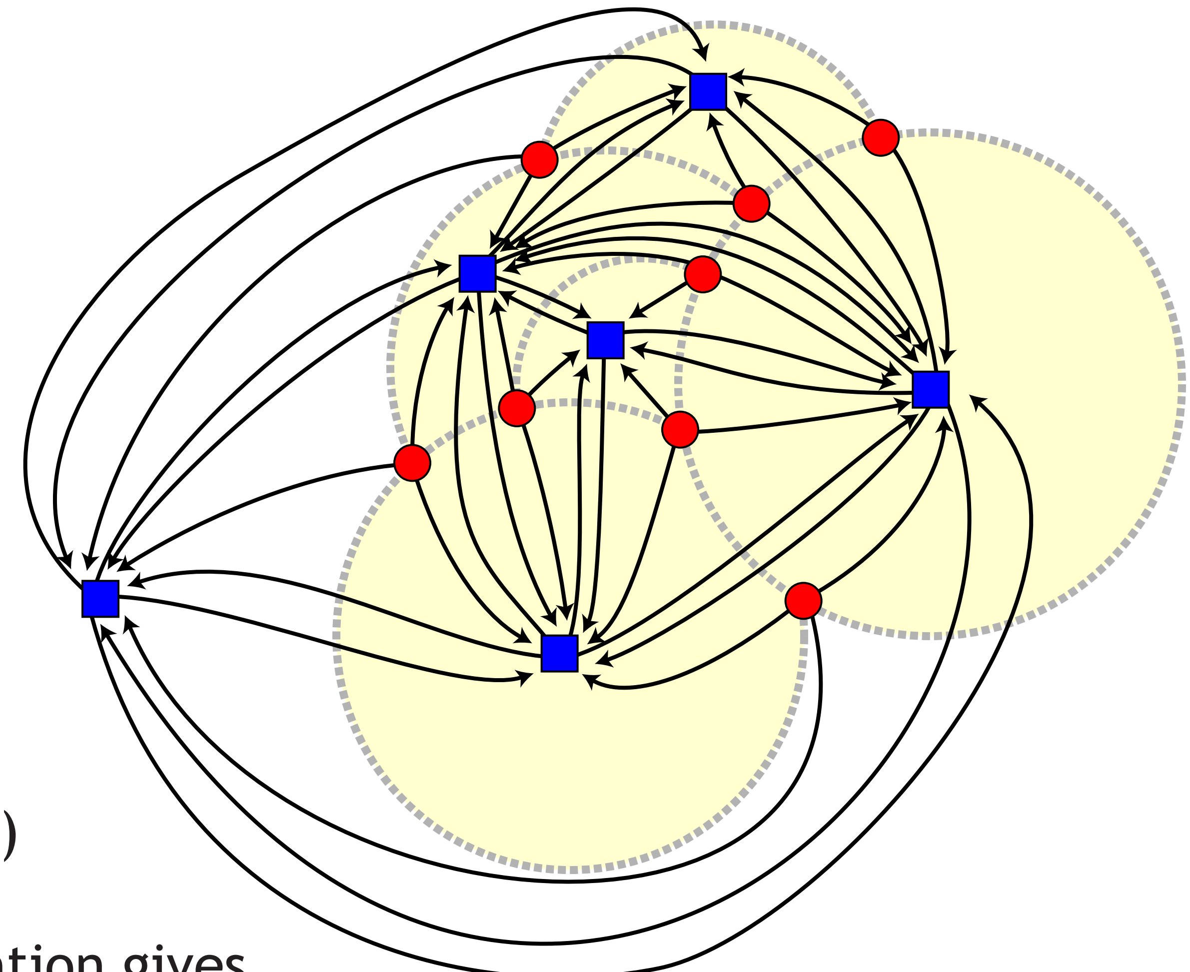
Junction-region edge:
min capacity 1, cost 0

Region-region edge:
min capacity 0, cost 1

Circulation-junction edge:
min&max capacity 4

Region-circulation edge:
min&max capacity 2k – 4
(or 2k + 4 for exterior region)

Minimum cost integer circulation gives
minimum bend layout



Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

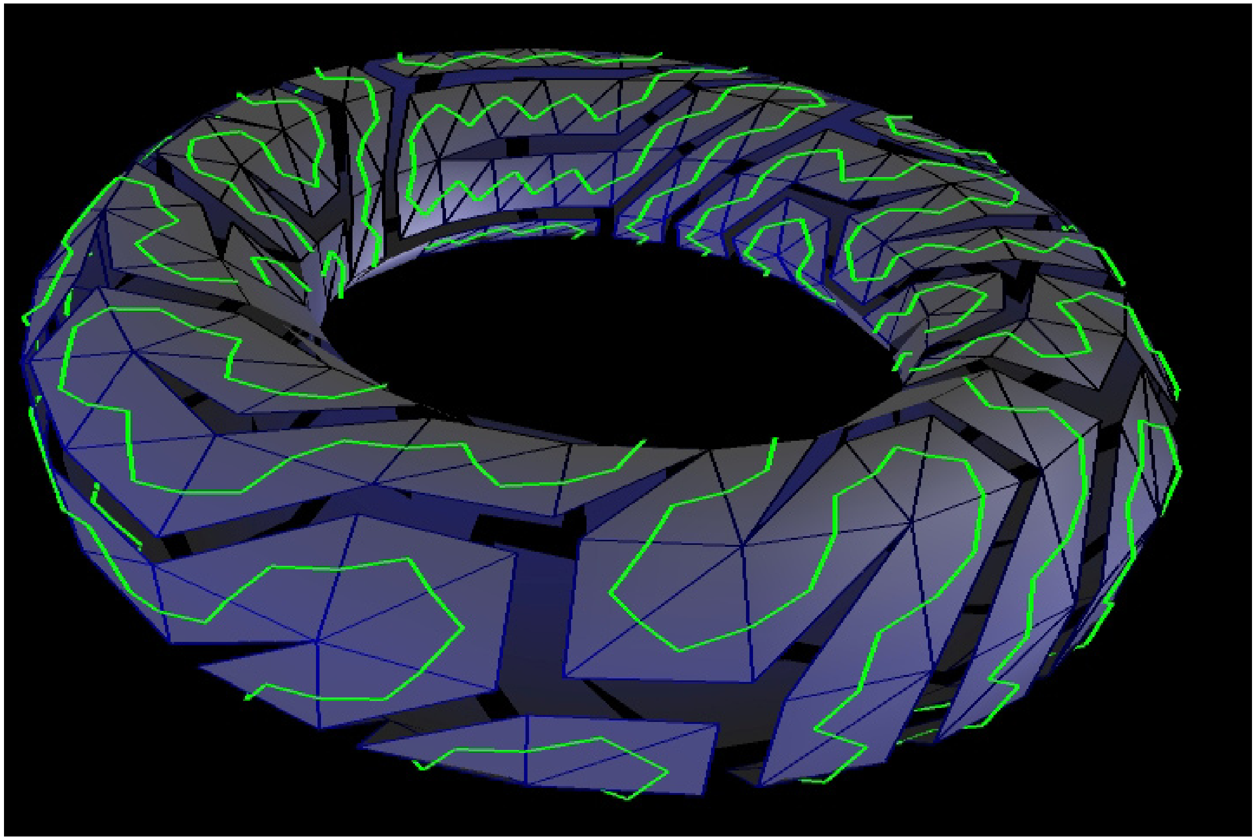
Bend minimization

Mesh stripification

Angle optimization of tilings

Metric embedding into stars

Problem: sort triangles of a geometric model into a single contiguous triangle strip



Communicate only vertex per triangle to graphics hardware instead of three

Part of schemes for data compression of mesh topology & geometry

Space-filling curve from strip useful for dithering, mesh simplification, etc.

[E. & Gopi, Eurographics 2004]

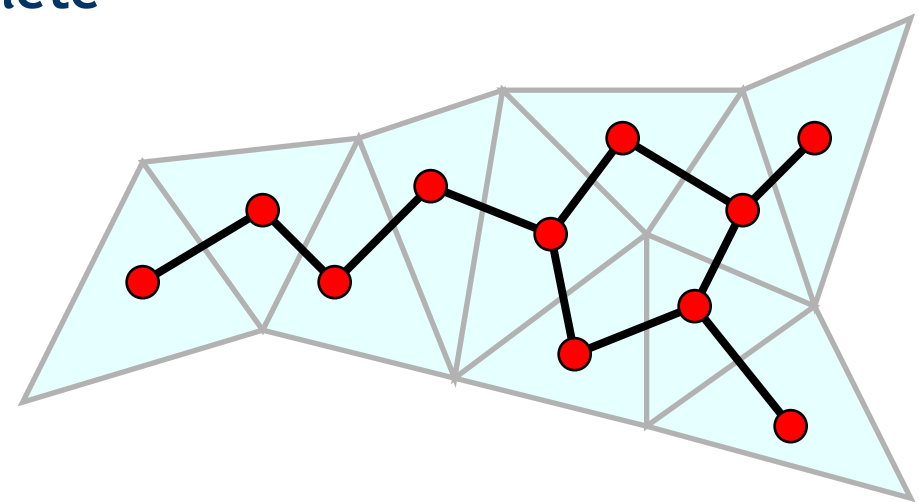
As with min-diameter clustering, a direct reduction to a graph is unhelpful

Form dual graph

- One vertex for every triangle
- Edge connecting two adjacent triangles

Single-loop triangle strip
= **Hamiltonian cycle in dual graph**

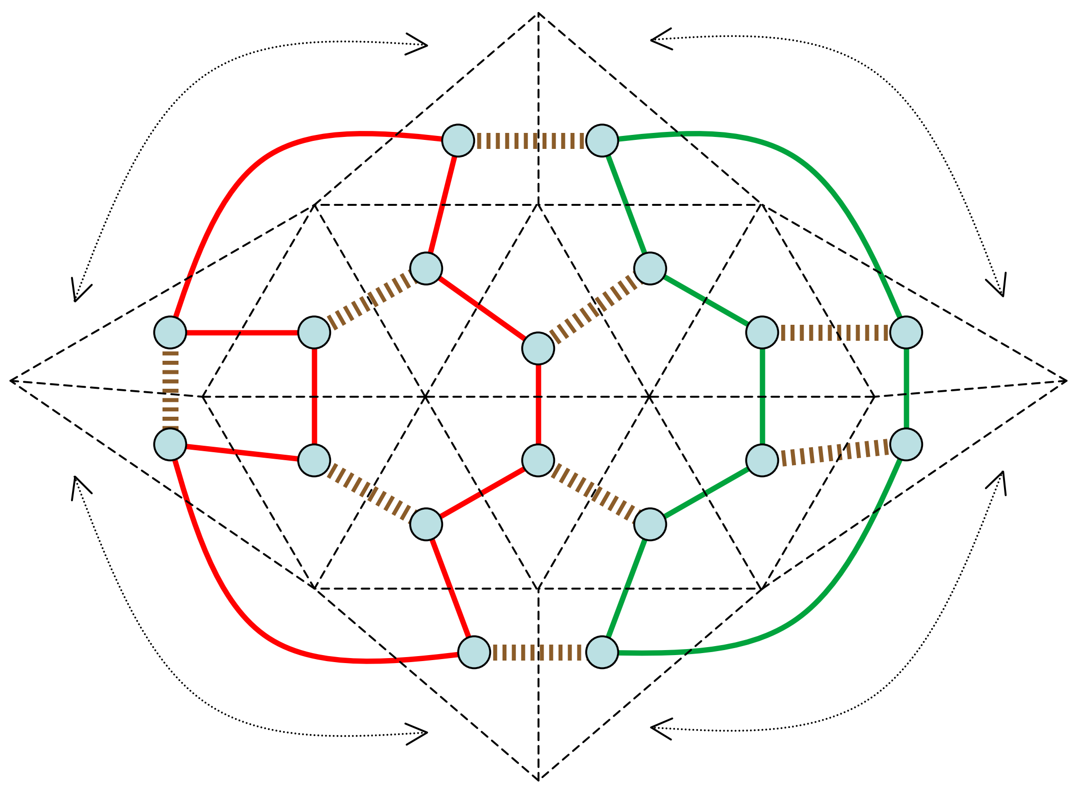
But even in dual graphs of planar triangular meshes,
Hamiltonian cycle is **NP-complete**
and **may not even exist**



Less-direct matching-based approach

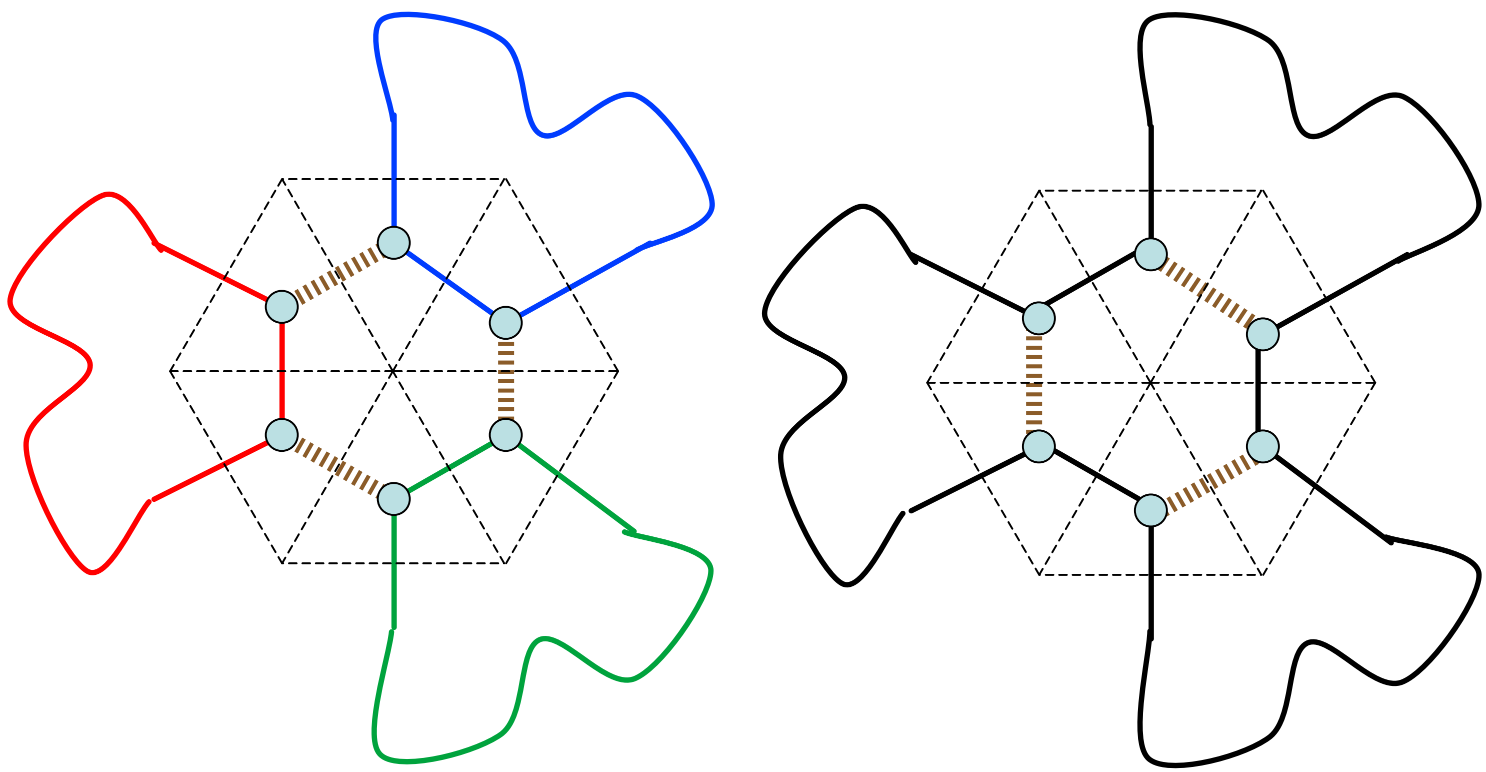
Dual graph is 3-regular and bridgeless; therefore, it has a perfect matching
[Peterson 1891; efficient algorithms due to Biedl et al., J. Algorithms 2001]

Complementary edges to matching form set of cyclic strips



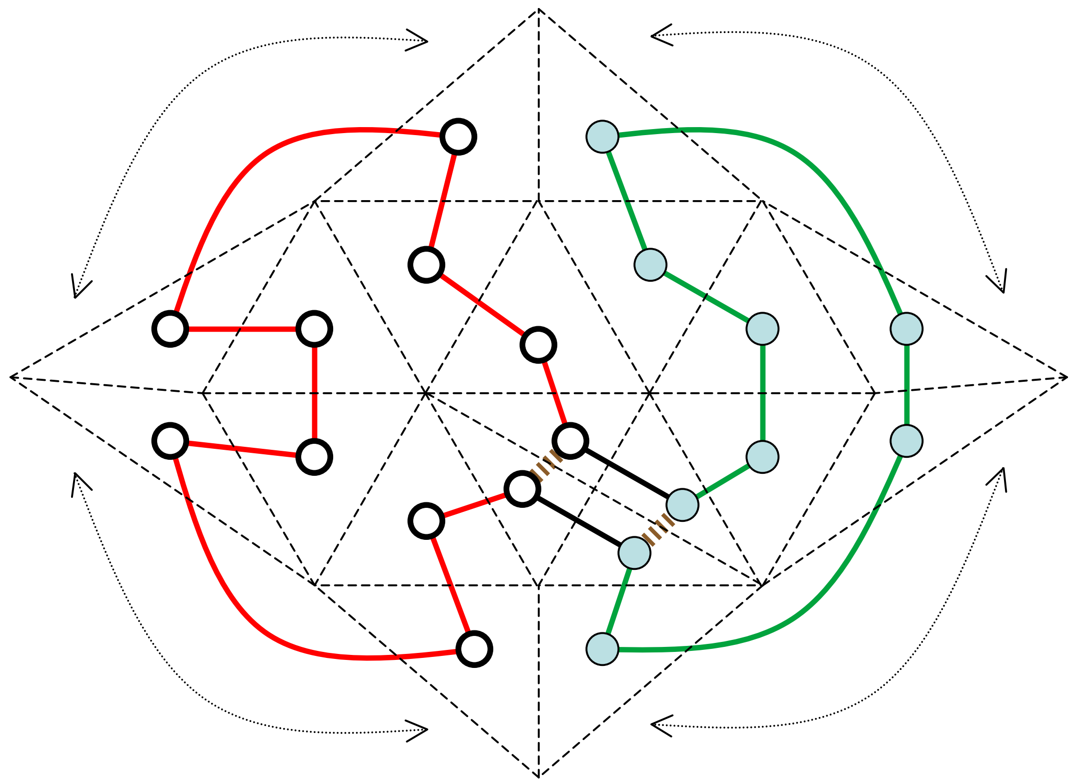
Less-direct matching-based approach (II)

In many cases, local moves allow number of cycles to be reduced



Less-direct matching-based approach (III)

In remaining cases, subdividing two triangles allows local move to merge cycles



Result: single strip with same geometry as original model
at most 3/2 as many triangles as original

In practice, increase only 1-2%; some models require 39/37 factor increase

Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

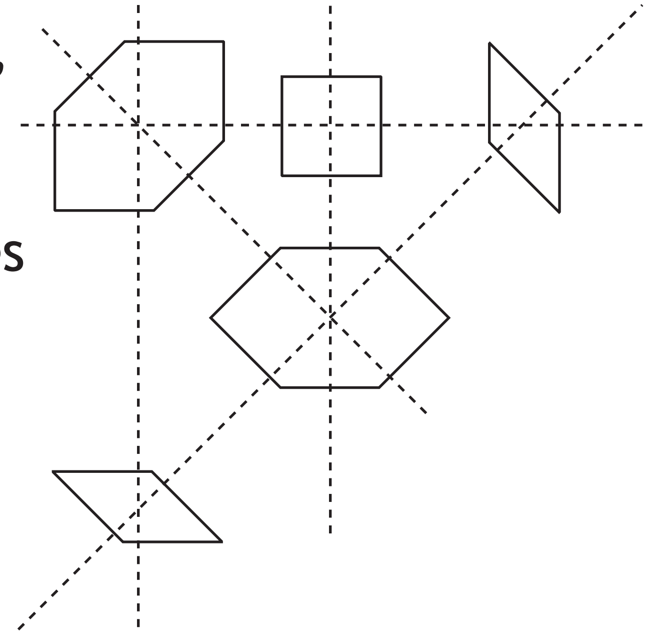
Mesh stripification

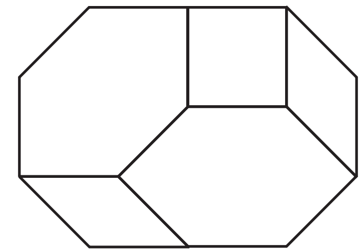
Angle optimization of tilings

Metric embedding into stars

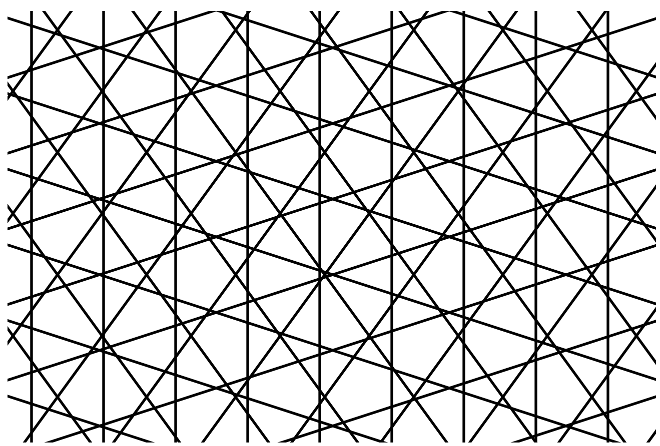
For each vertex of a line arrangement, form convex centrally-symmetric tile

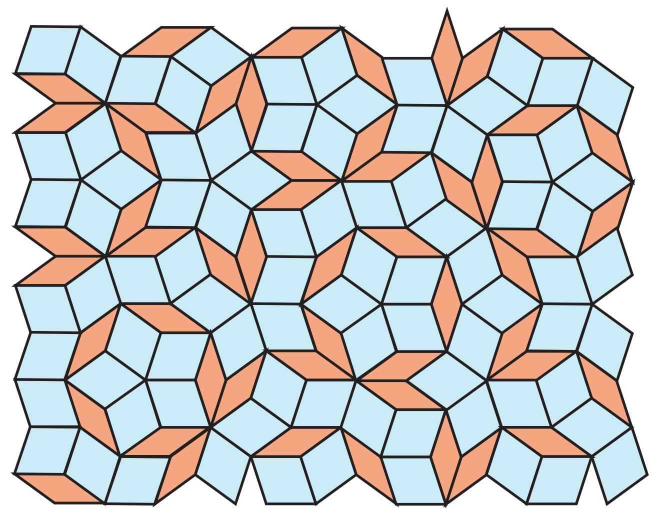
- Unit length sides
- Edges perpendicular to lines





These tiles fit together into a tiling of a convex polygon (or, for infinite arrangements, a tiling of the Euclidean plane)



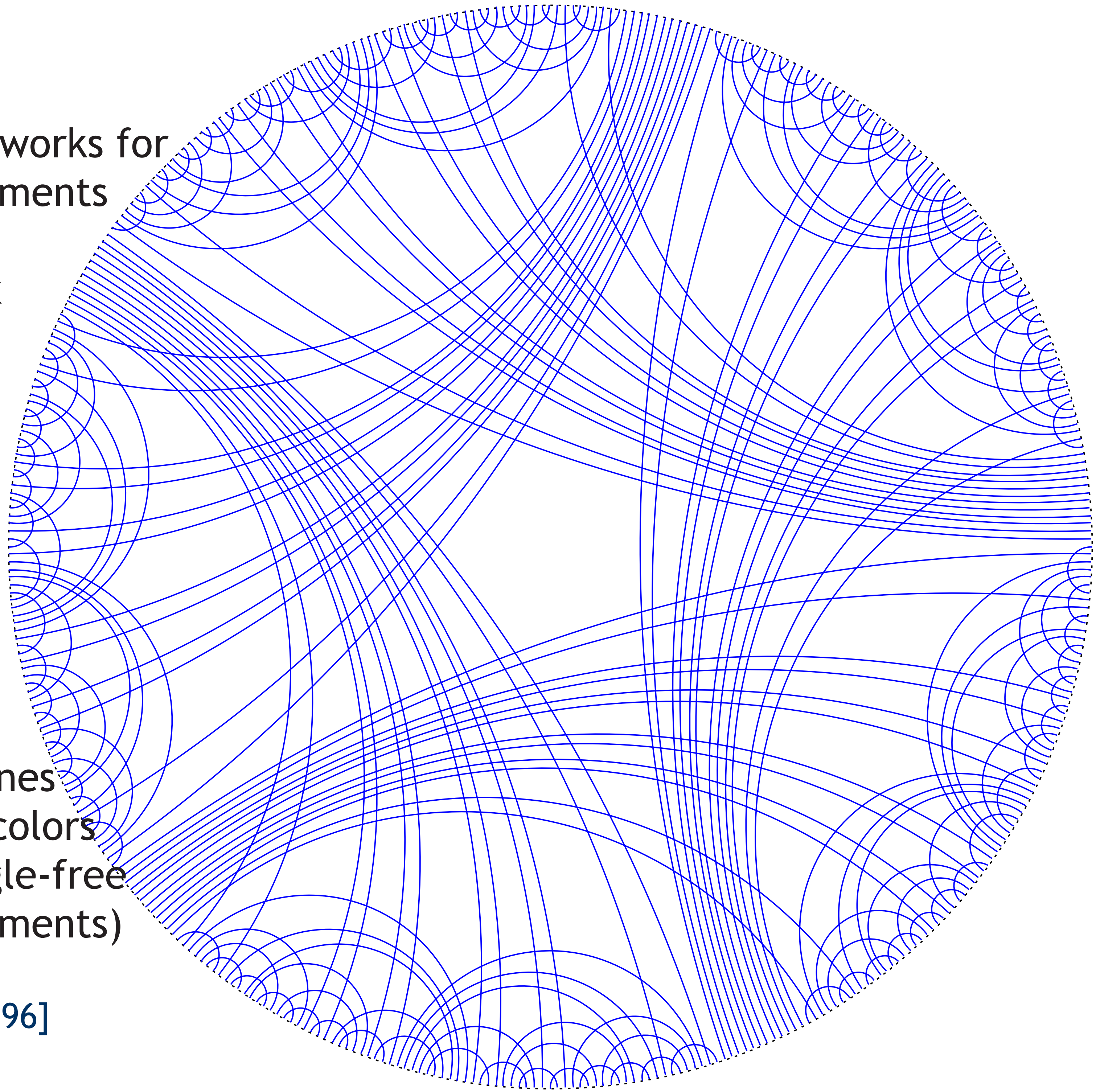


This idea was used by De Bruijn [Indag. Math. 1981] to form a Penrose rhomb tiling from five sets of parallel lines

Same method also works for hyperbolic arrangements

(tiles a non-convex subset of the Euclidean plane)

[E., GD 2004]



This arrangement has no triangles and requires five colors if crossing lines must be different colors (the max for triangle-free hyperbolic arrangements)

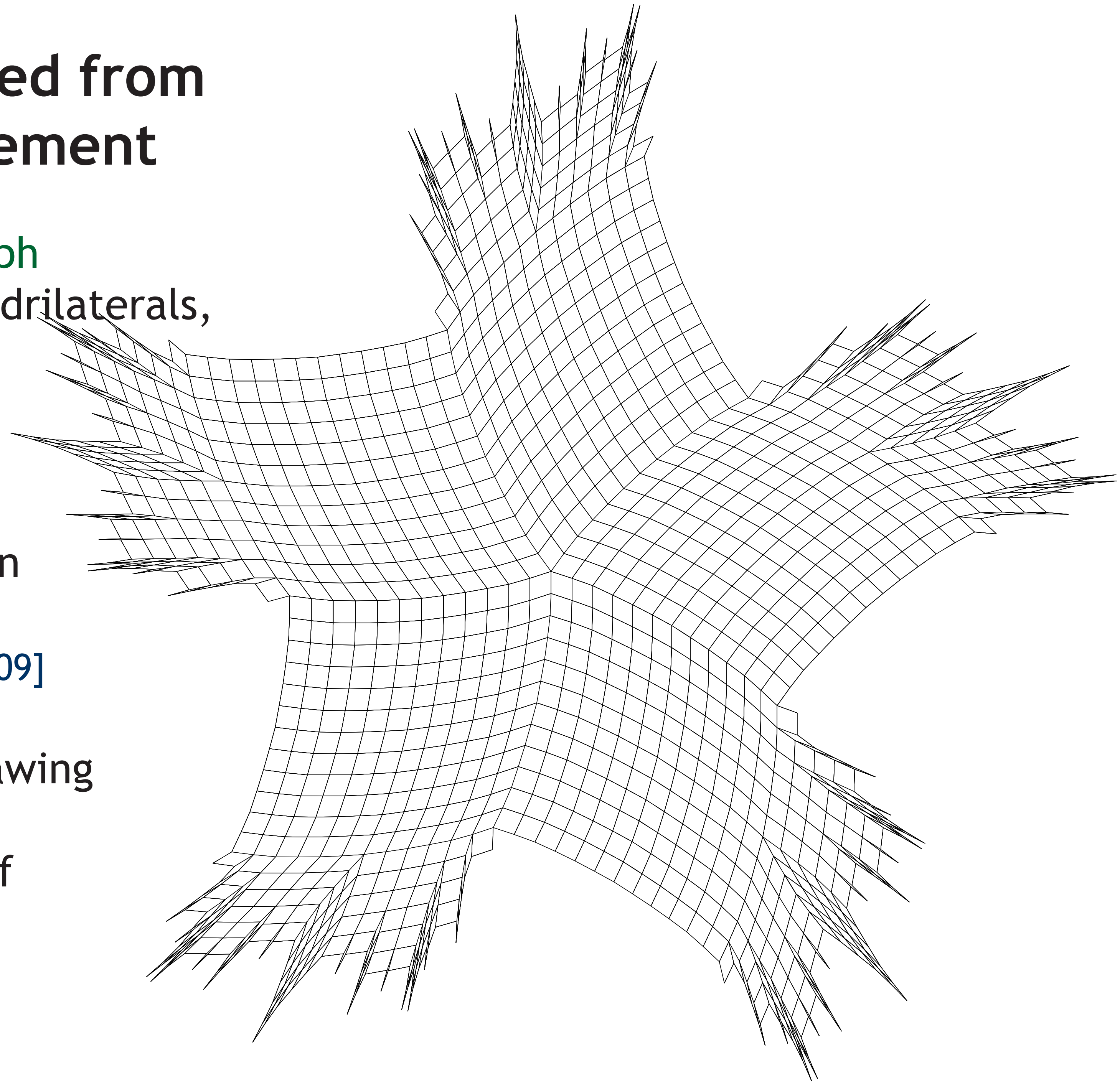
[Ageev, Disc. Math. 1996]

The tiling formed from Ageev's arrangement

Result is a squaregraph
(planar, all faces quadrilaterals,
all interior vertices
have degree ≥ 4)
that cannot be
embedded into the
product of fewer than
five trees

[Bandelt, Chepoi, E., 2009]

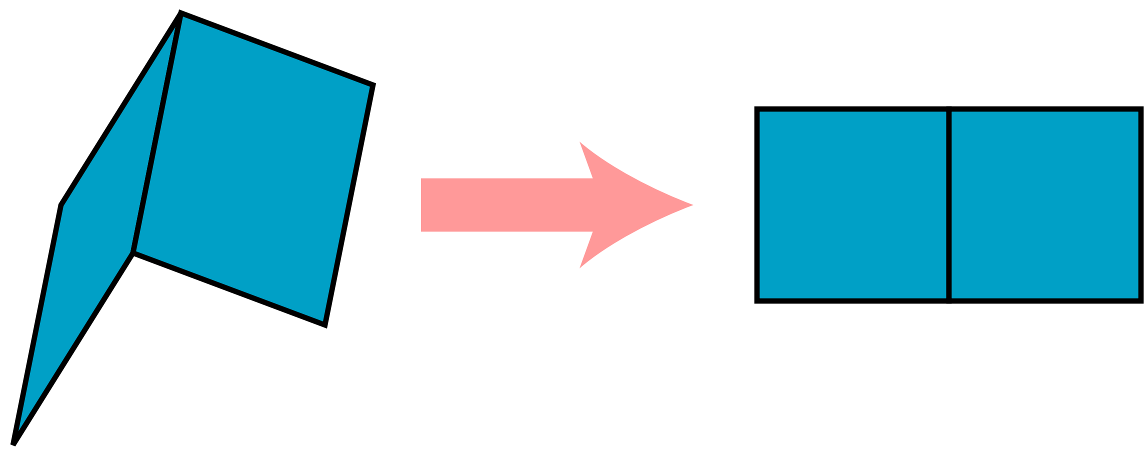
Can we make the drawing
more legible
by adjusting slopes of
sides of tiles?



Optimization of angular resolution [E. & Wortman, 2009]

Angular resolution = sharpest angle of drawing [Malitz & Papakostas, STOC 1992]

Given tiling by symmetric polygons
find combinatorially equivalent tiling with optimal angular resolution



Transformation to parametric shortest path problem

Graph in which edges have as weights linear functions of λ

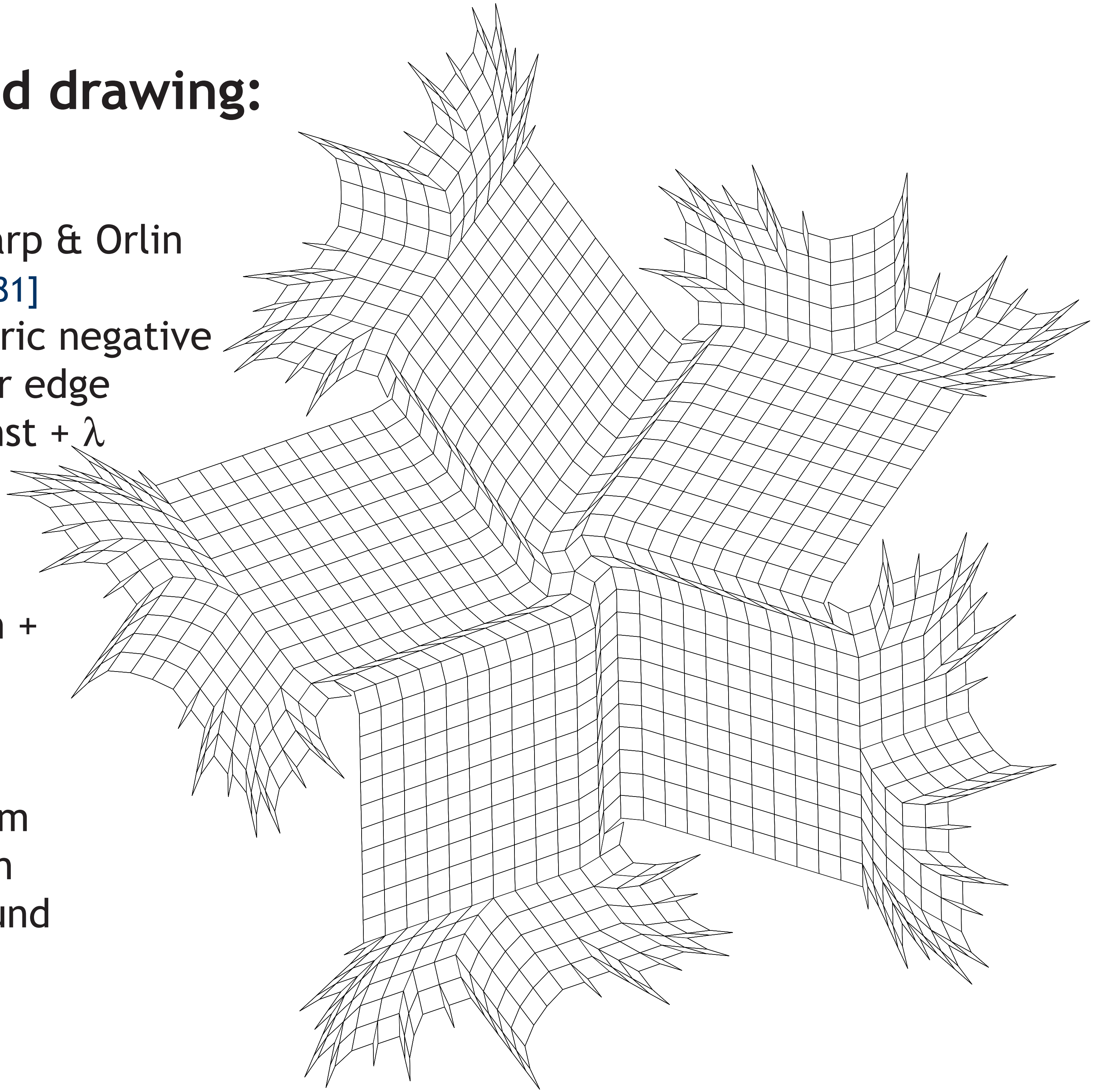
Concept in tiling:	**Corresponding concept in graph:**
"Zone" z_i of parallel line segments	Vertex v_i
Angular resolution α	Parameter value λ
α is a feasible resolution	Graph has no negative cycles for λ
Amount to adjust angle of z_i	Distance from start vertex s to v_i
Angle $z_i z_j$ is at least α	Edge $v_i v_j$ with weight $\theta_i - \theta_j - \lambda$
Interior angles are convex	Edge $v_i v_j$ with weight $\pi + \theta_i - \theta_j$

The optimized drawing:

An algorithm of Karp & Orlin
[Disc. Appl. Math. 1981]
can solve parametric negative cycle detection for edge weights const, const + λ
in time $O(n^3)$

(Implementation used binary search + Bellman–Ford)

Translation to and from graph problem can be done within the same time bound



Outline

Art gallery theorems

Partition into rectangles

Minimum diameter clustering

Bend minimization

Mesh stripification

Angle optimization of tilings

Metric embedding into stars

Metric space

Set X of points, and function d from pairs of points to real numbers satisfying

Symmetry: for all x and y, $d(x,y) = d(y,x)$

Positivity: for all x and y, $d(x,y) \geq 0$, with equality iff $x = y$

Triangle inequality: for all x, y, and z, $d(x,y) + d(y,z) \geq d(x,z)$
(if equal, then y is "between" x and z)

Examples:

Shortest path lengths in weighted undirected graphs

Euclidean distance between points in $\mathbf{R}^d$

Embeddings and distortion

Embedding: 1-1 function from the points of one metric space to another

dilation of a single distance:

$$\frac{d(f(x),f(y))}{d(x,y)}$$

distortion of embedding = worst ratio of dilations

$$\max_{x\neq y}\frac{d(f(x),f(y))}{d(x,y)} \Big/ \min_{u\neq v}\frac{d(f(u),f(v))}{d(u,v)}$$

if embedding scaled so all distances nondecreasing, distortion = max dilation

$$\max_{x\neq y}\frac{d(f(x),f(y))}{d(x,y)}$$

Structured vs unstructured metrics

Unstructured: any metric possible
$O(n^2)$ degrees of freedom in specifying distances

Distance matrix

Graph shortest path distances

n-dimensional $L_{infinity}$ metric

Structured: constrained subset of metrics
$O(n)$ degrees of freedom

Low-dimensional Euclidean or L_p metric

Tree shortest path distances

Metric embedding problem

Given an unstructured metric space
find a low-distortion embedding into a structured space

or, sometimes (not today)
find a random family of embeddings in which
any individual distance has low expected distortion

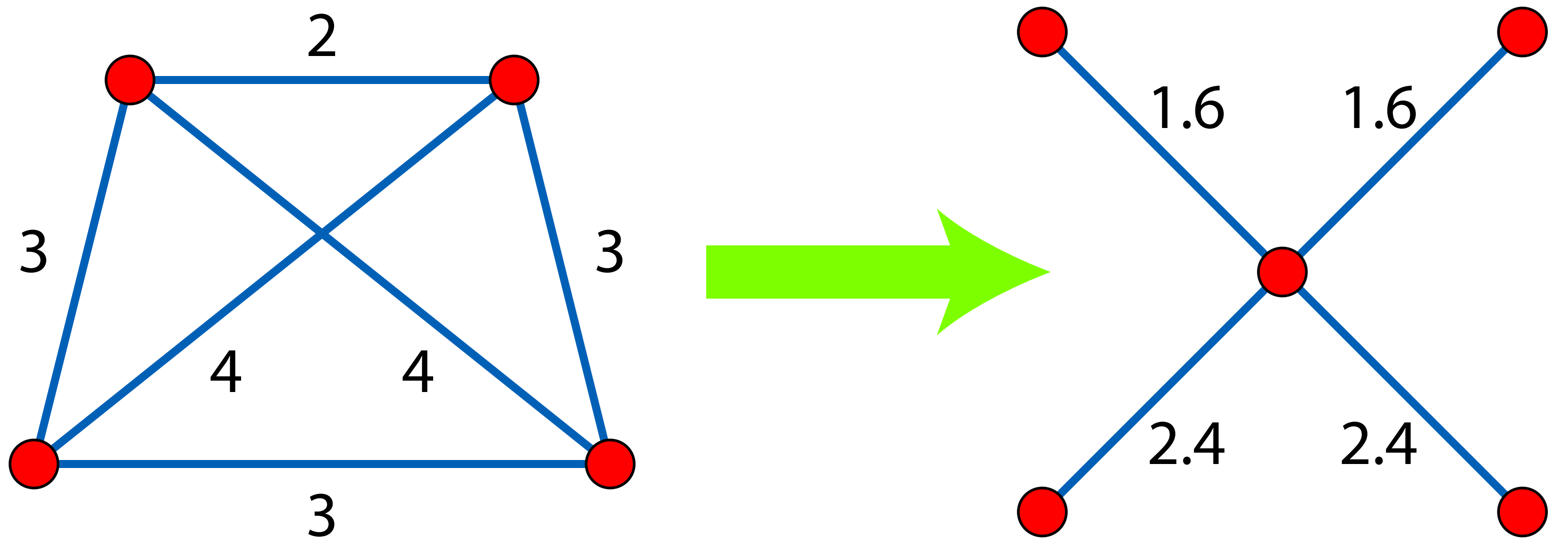
Many applications in which
approximation algorithm designed for structured space
can be extended to arbitrary metric spaces

Minimum dilation star problem [E. & Wortman, WADS 2009]

Find minimum distortion embedding

Target structured space is a star:
There exists a point (not necessarily in the image of the embedding)
that is between every other pair of points

Equivalently, **graph shortest path metric on a tree with one new non-leaf node**

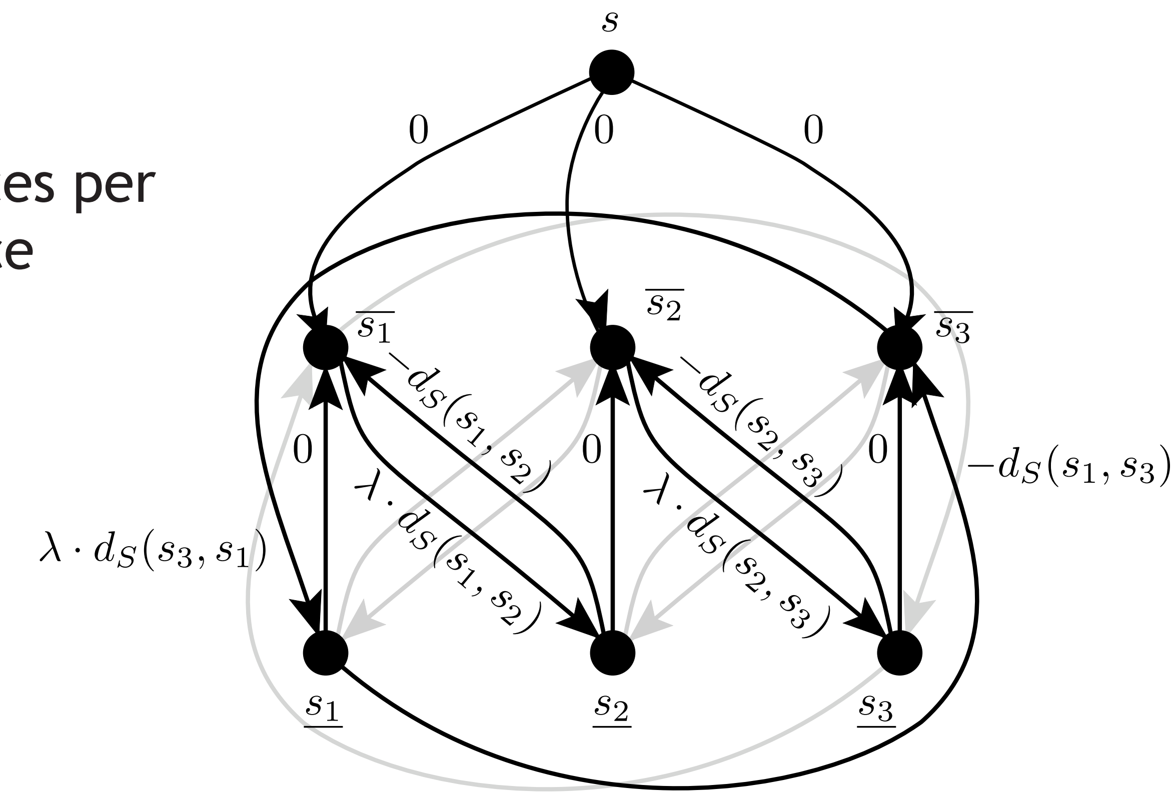


Solution idea

Form a graph, with two vertices per point of the input metric space

Downward edge weight = negative distance

Upward edge weight = param times distance



Optimal distortion =
minimum parameter value s.t. graph has no negative cycles
(no longer Karp–Orlin, but still polynomial)

Distance from star hub to leaf point in embedding =
1/2 (difference between distances from s to two vertices for that point)
always non-negative because of upwards length-0 edge

Conclusions

Graph-theoretic point of view is useful in many non-graph problems

The graph algorithms used for these problems are often classical...

maximum independent set and maximum clique
maximum or perfect matching
maximum or minimum-cost flow
graph shortest paths

...but sometimes with a twist

parametric negative cycle detection

Special classes of graphs and their structure is often important

maximal outerplanar graphs
squaregraphs and kinggraphs
bipartite graphs
intersection graphs of unit circles
planar graphs
bridgeless 3-regular graphs

Much more likely remains to be discovered