Quasiconvex Programming

David Eppstein

Univ. of California, Irvine
School of Information and Computer Science

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

Outline

Minimum enclosing disk example

Quasiconvex programming

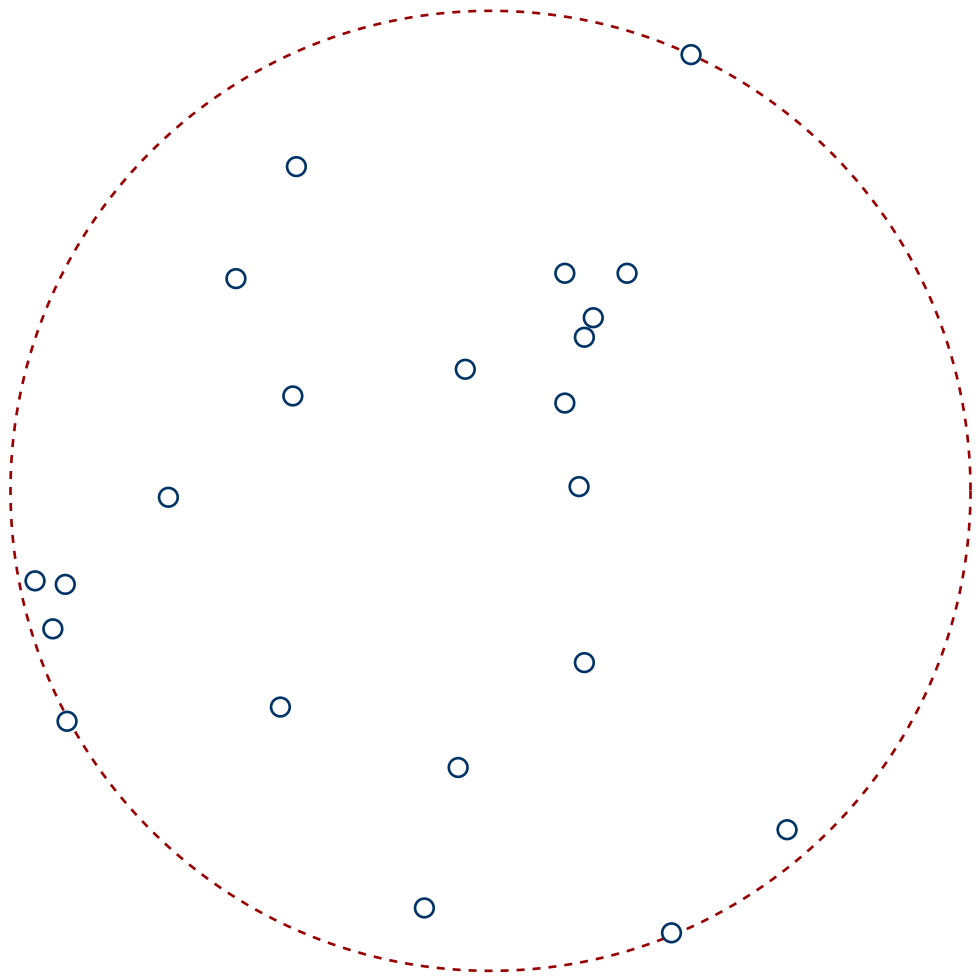
Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

Minimum enclosing disk of *n* planar points



Either diameter circle of two points
or circumcircle of three points forming acute triangle

O(n) time algorithms are known and implemented (e.g. Gärtner)

Minimum enclosing disk as quadratic program

Represent disk with radius r centered at x_0, y_0
as triple (x_0, y_0, R) where $R = r^2 - x_0^2 - y_0^2$

Requirement that disk contains input point (x, y) becomes
linear inequality constraint $(x_0, y_0, R) \bullet (-2x, -2y, 1) \geq x^2 + y^2$

Area minimization criterion becomes
convex quadratic objective function $R + x_0^2 + y_0^2$

Can solve via local improvement
or more sophisticated algorithms e.g. ellipsoid

Minimum enclosing disk as generalized linear program

Function f(S) mapping sets of points to circumradius satisfies axioms:

If $S \subset T$, then $f(S) \leq f(T)$

If $f(S) = f(S \cup \{x\})$ and $f(S) = f(S \cup \{y\})$, then $f(S) = f(S \cup \{x, y\})$

For some constant d, if $|S| > d$, there exists $x \in S$ with $f(S) = f(S \setminus \{x\})$

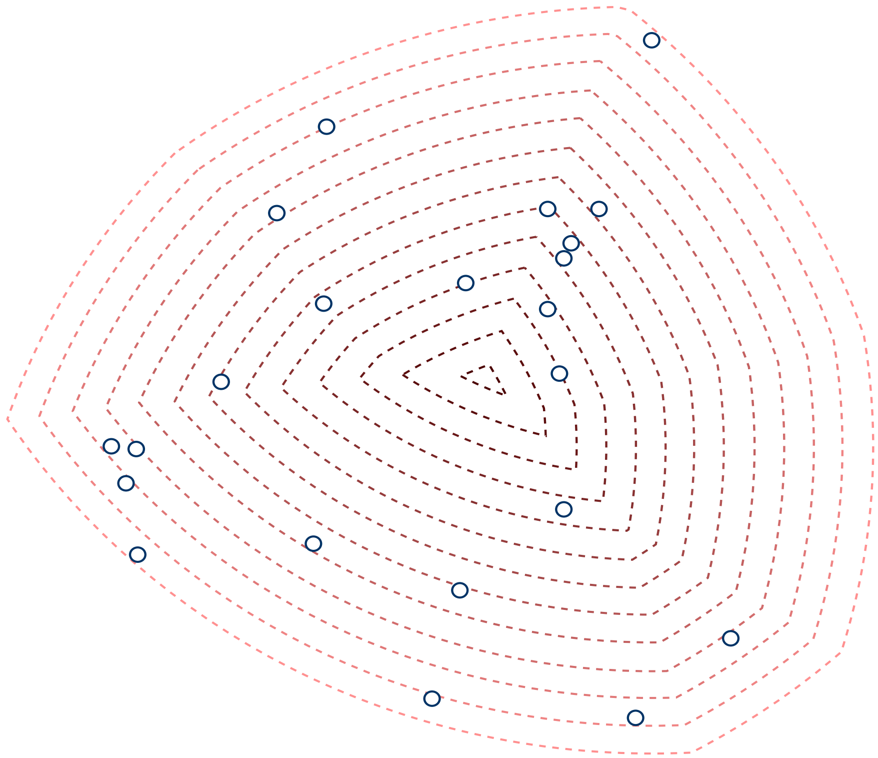
Any such function can be evaluated by randomized dual-simplex algorithms
[Matousek, Sharir, and Welzl, 1992; Amenta, 1994; Gärtner, 1995]

Typically: $O(n)$ feasibility tests (is point inside current circle?)
$o(n)$ basis change operations (find circumradius of constant size subset)

Minimum enclosing disk as quasiconvex program

Distance $d_p(x)$ from site p is *quasiconvex*: level sets $\{x: d_p(x) \leq r\}$ are convex

Want to find x minimizing $\max_p d_p(x)$



Level of abstraction between convex programs and generalized LP

Includes problems that do not form convex programs
Quasiconvexity may be easier to prove than GLP axioms
Unlike GLP, still permits numerical solution techniques

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

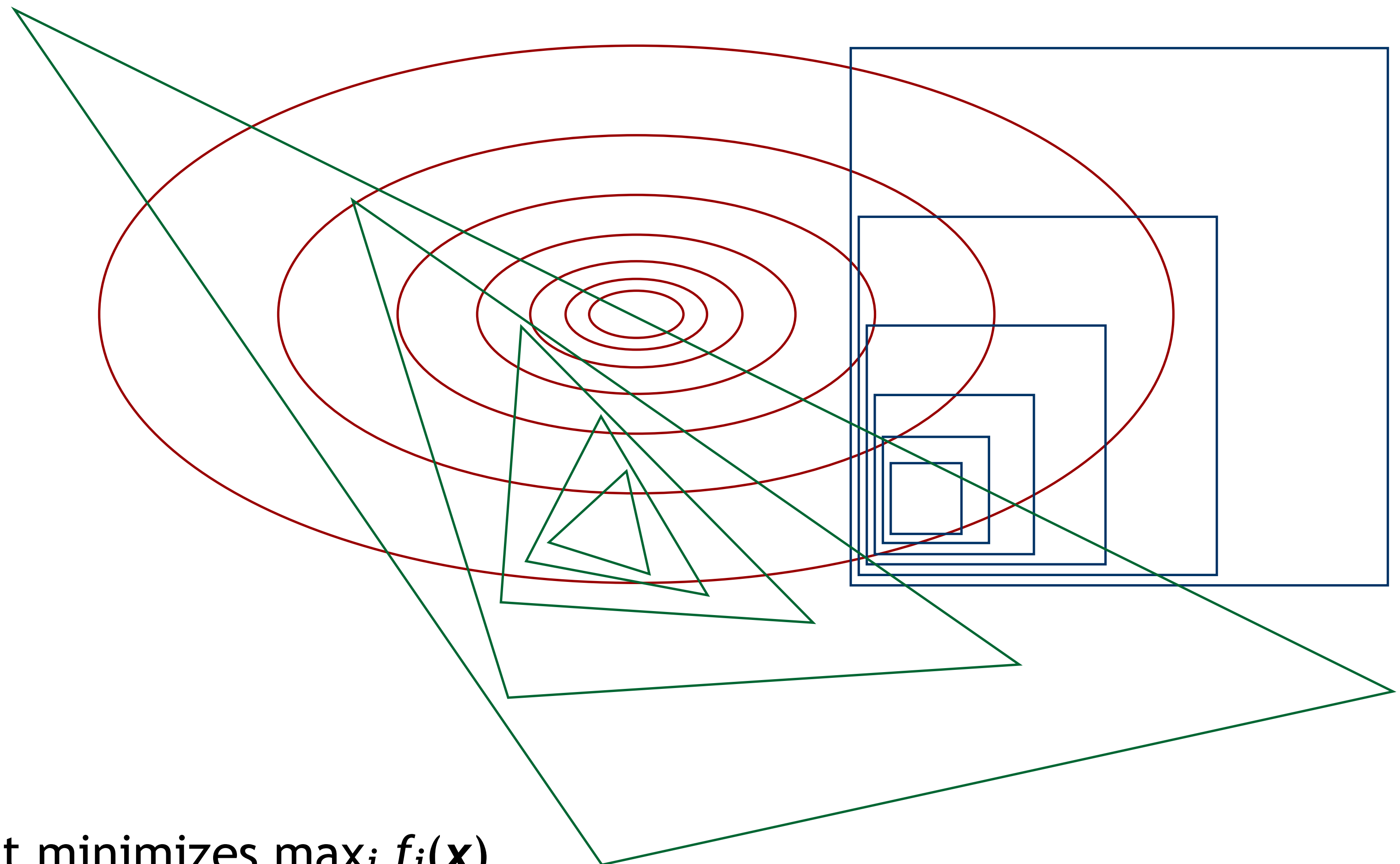
Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

Quasiconvex program:

Input: family of quasiconvex functions $f_i(\boldsymbol{x})$, $\boldsymbol{x}$ in $\mathbf{R}^d$

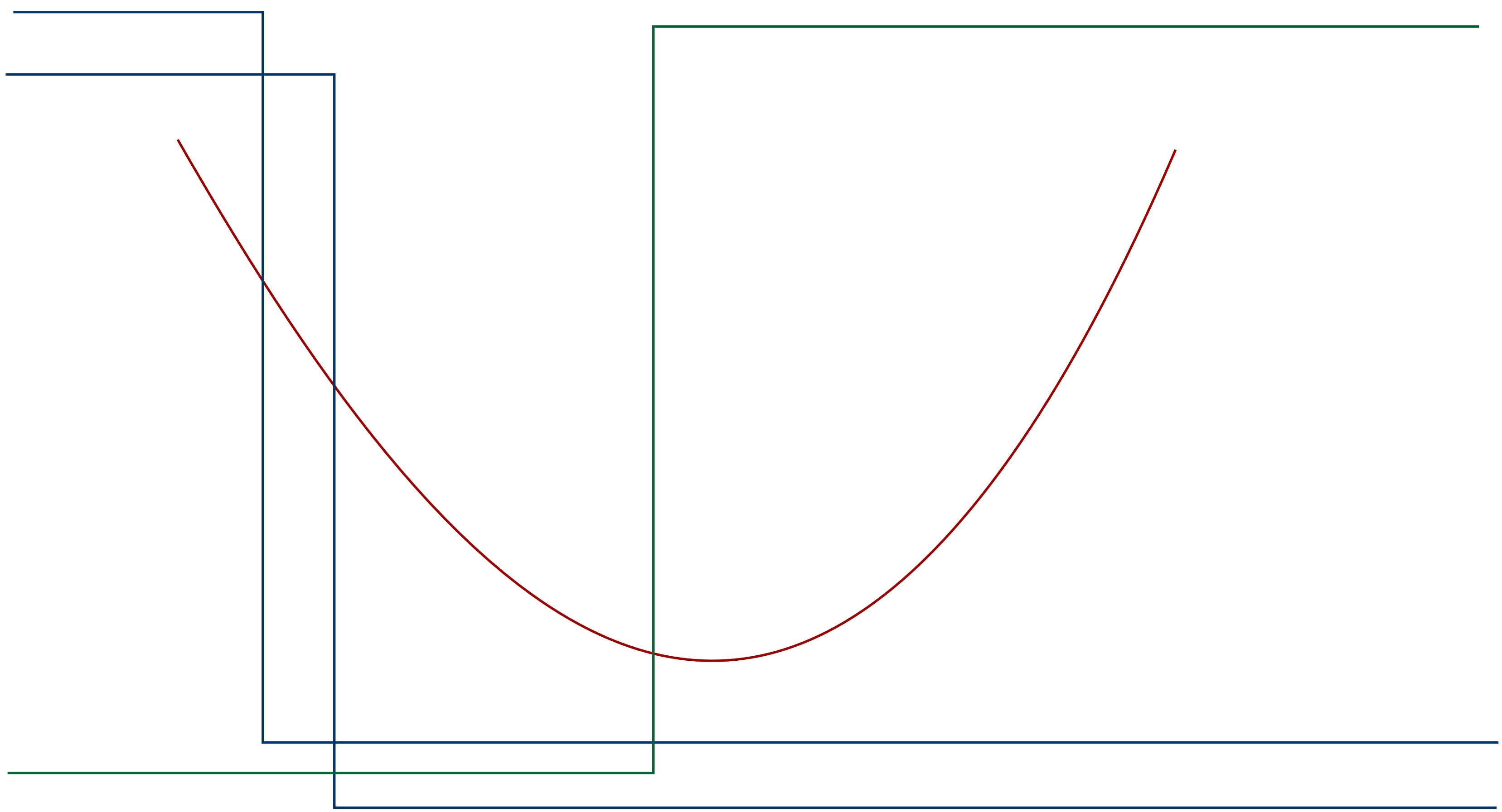


Output: $\boldsymbol{x}$ that minimizes $\max_i f_i(\boldsymbol{x})$

Convex programs are quasiconvex programs

Convex objective function is quasiconvex

Replace each linear inequality constraint by a step function
Very high value where constraint violated
Very small value where constraint satisfied



Quasiconvex programs are generalized linear programs

Define $f(S) = \max_{i \in S} f_i(\boldsymbol{x})$

f satisfies all the GLP axioms

Helly's theorem gives bounds on GLP dimension (cardinality of basis):

at most 2d+1 for arbitrary quasiconvex program

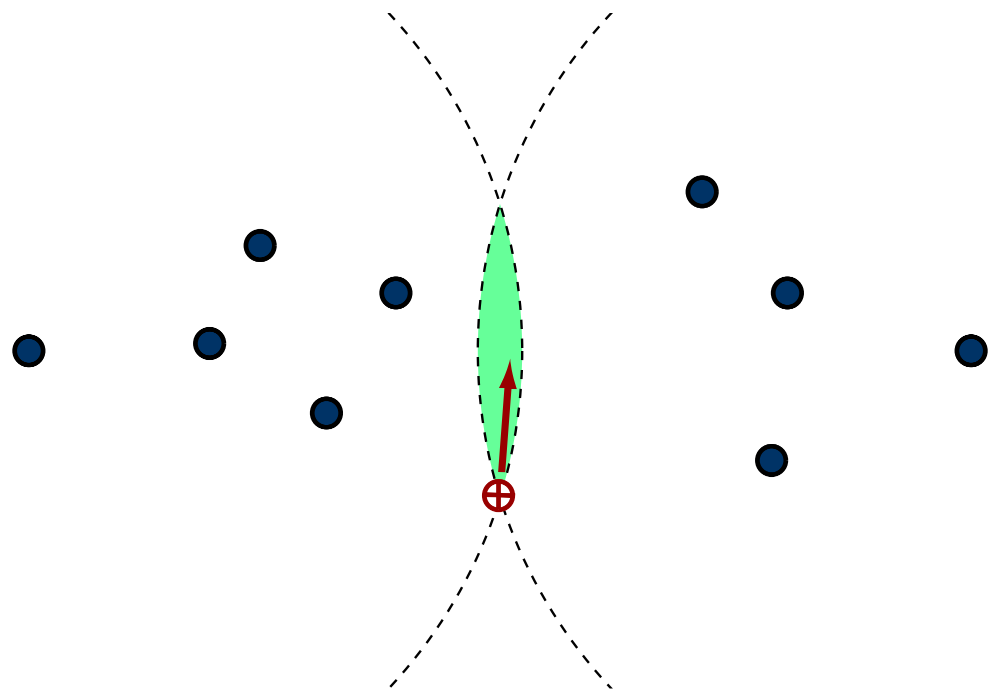
at most d+1 for well-behaved quasiconvex functions
(level sets strictly nested, not constant on any open set)

...so can use GLP algorithms if basis change operation can be implemented

Numerical search for quasiconvex program value

Objective function $\max_i f_i(\boldsymbol{x})$ is itself quasiconvex

No local optima to get stuck in, so
local improvement techniques will reach global optimum



How to find improvement direction?

May get trapped in sharp corner
e.g. for minimum enclosing disk, equidistant from diameter points

Smooth quasiconvex programming (multi-gradient descent)

Suppose level sets have unique tangents at all boundary points
e.g. differentiable functions, step functions of smooth convex sets
(in 2d, use left & right tangents without smoothness assumption)

Then can find gradient $\boldsymbol{v}$ s.t. $\boldsymbol{w}$ is improvement direction iff $\boldsymbol{v} \bullet \boldsymbol{w} < 0$

Repeat:

Find gradients of functions within numerical tolerance of current max

Find simultaneous improvement direction $\boldsymbol{w}$ for all gradients
(lower dimensional minimum enclosing disk of few points)
If not found, algorithm has converged to solution

Replace $\boldsymbol{x}$ by $\boldsymbol{x} + \Delta\, \boldsymbol{w}$ for sufficiently small Δ

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

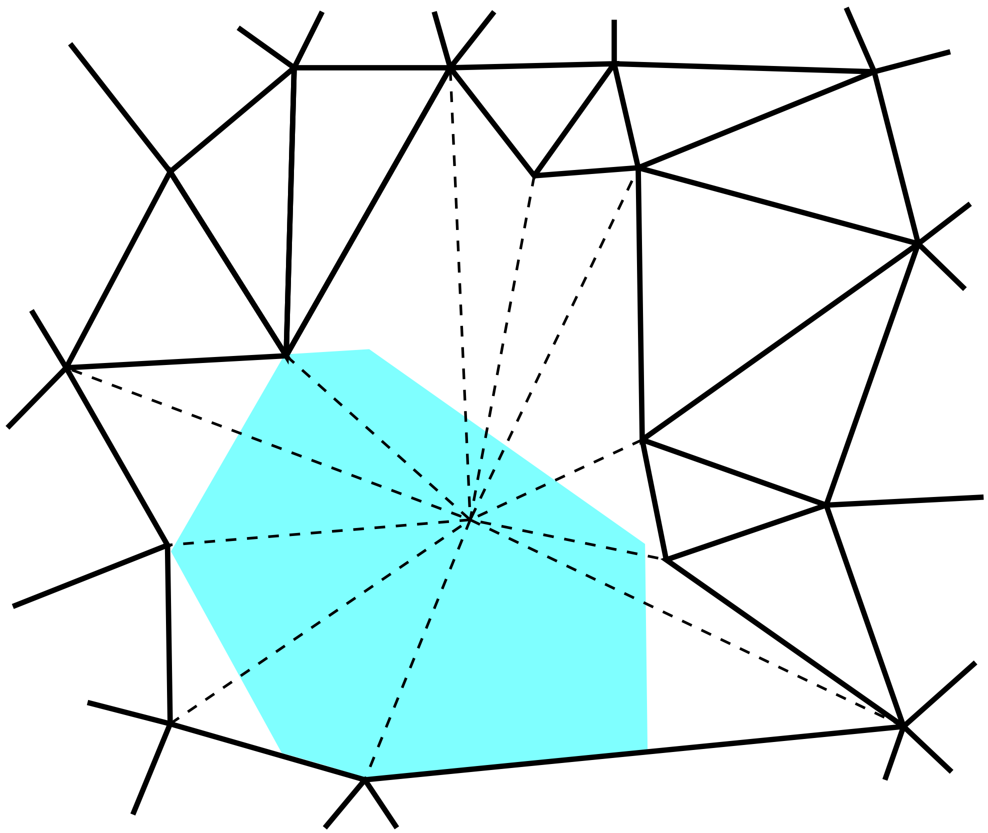
Color gamut optimization

Analysis of backtracking algorithms

Mesh improvement problem

Finite element meshes produced by standard methods (e.g. quadtree)
are insufficiently regular (many 45-45-90 triangles, prefer equilateral)

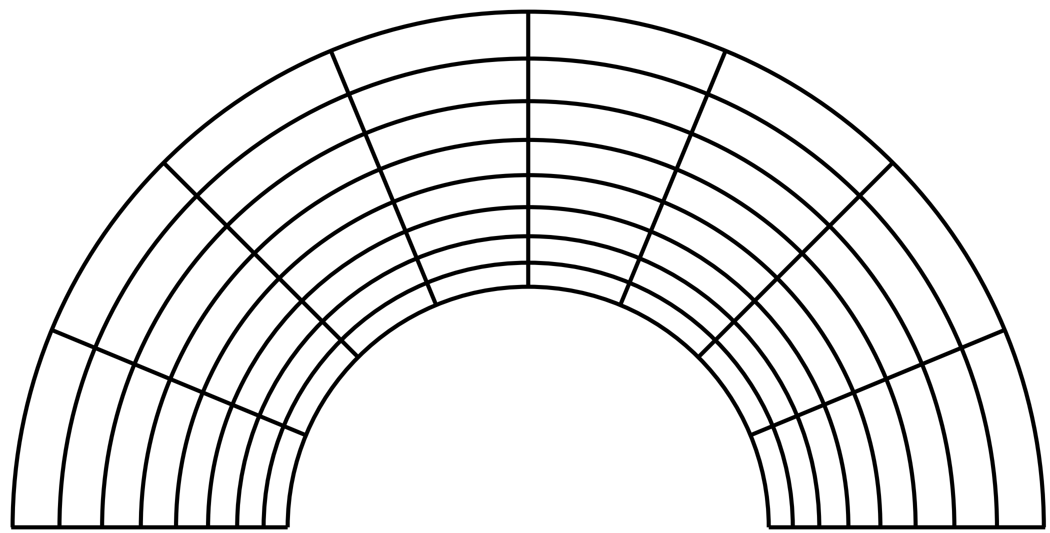
Solution: make local improvements to mesh to improve shape
e.g. move vertices one at a time to better location



Standard approach: Lagrangian smoothing

Move vertex to average of neighbors' positions

Unclear what it optimizes, can lead to malformed meshes



after too many iterations, arch center starts to droop...

Optimization based smoothing

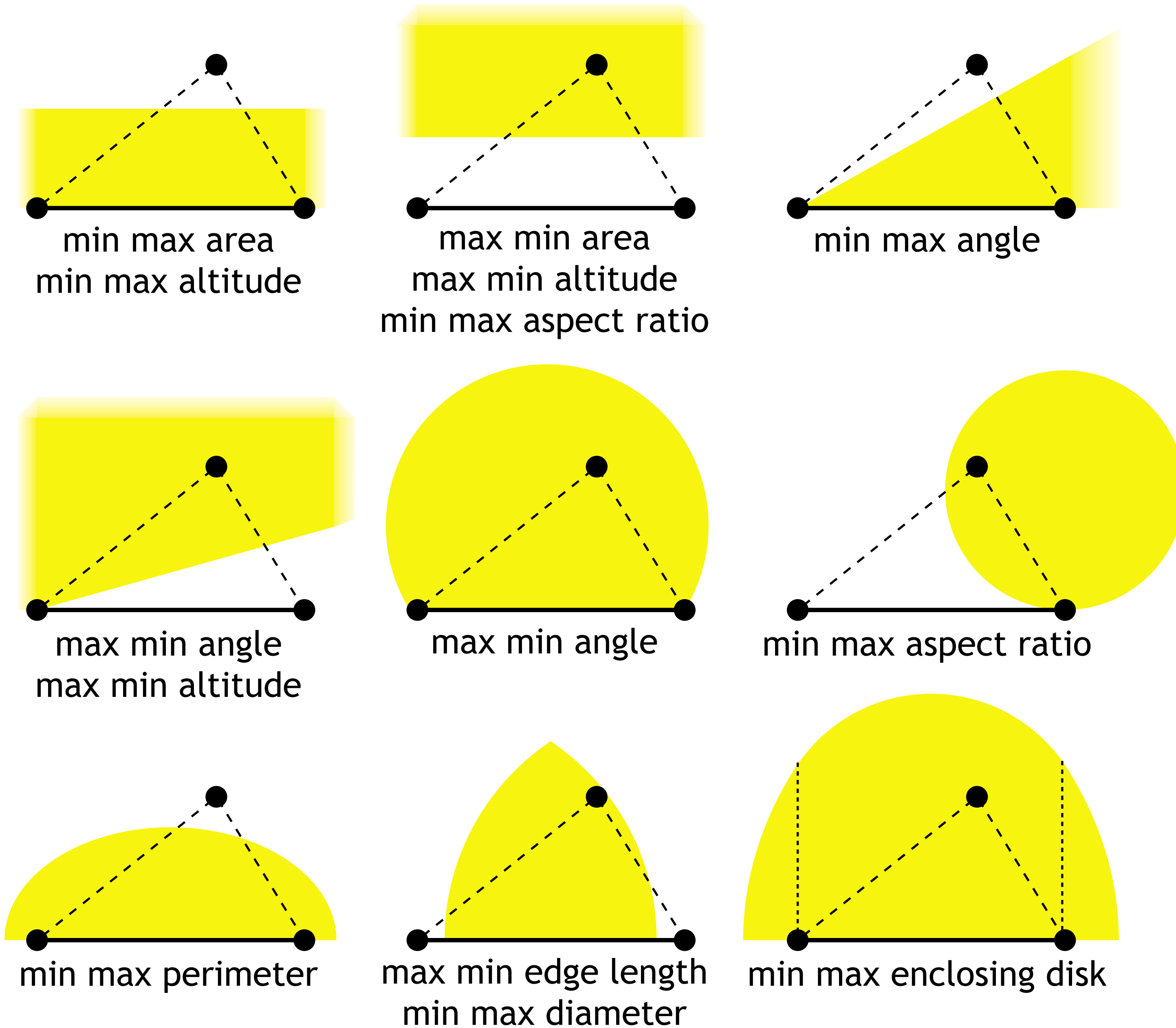
Compute new vertex location to optimize shapes of nearby elements

Many possible choices for mesh quality measure

Typical practice: simple hill-climbing

Why should this work?

Element quality measures and level set shapes



Similar but more complicated in three dimensions...

Mesh smoothing conclusions

Hill climbing avoids local optima: many quality measures are quasiconvex
Can ignore smoothing efficiency when choosing among these measures

Number of inputs is typically small, precise answers unimportant
so numerical improvement may be more appropriate than GLP

May be appropriate to switch from naïve hill-climbing
techniques to multi-gradient descent

Mesh topology changes also important (especially in 3d)
unclear how to mix with smoothing

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

What are Möbius transformations?

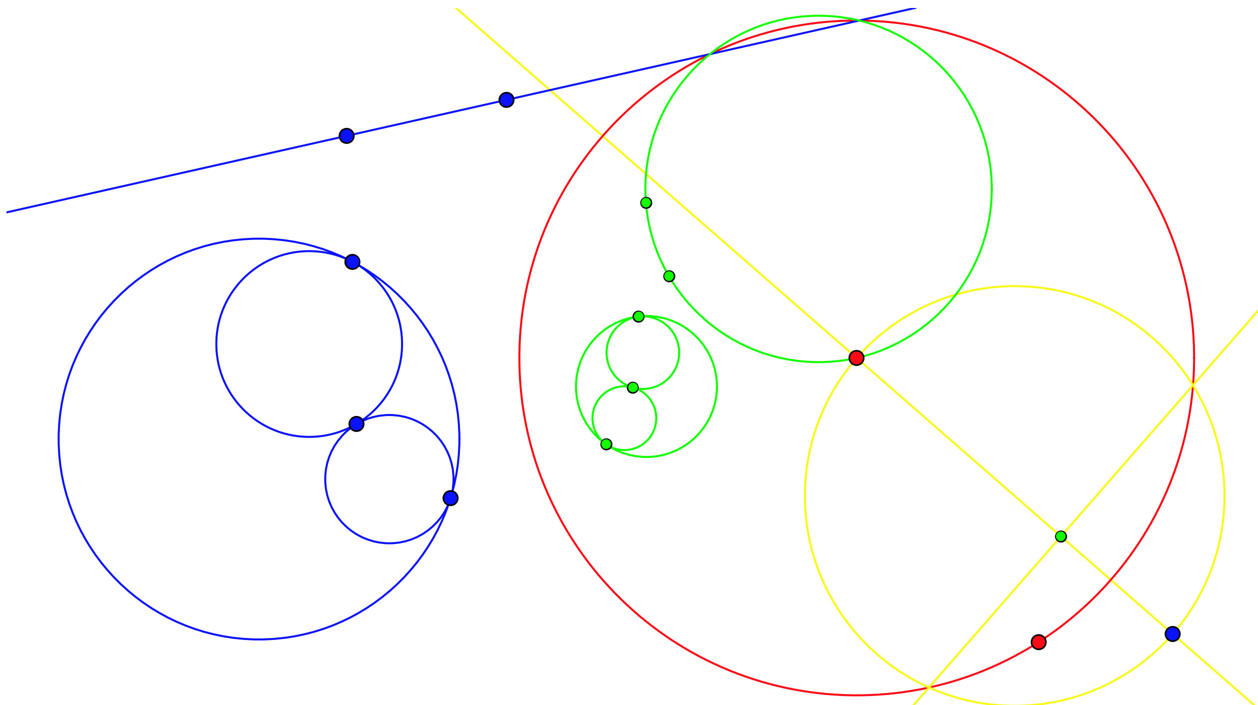
Fractional linear transformations of complex numbers:

$$z \rightarrow (a\,z + b)\ /\ (c\,z + d)$$

But what does it mean geometrically?
How to generalize to higher dimensions?
What is it good for?

Inversion

Given any circle (red below)
map any point to another point on same ray from center
product of two distances from center = radius^2



Circles ⇔ circles
(lines = circles through point at infinity)

Conformal (preserves angles between curves)

Möbius transformations = products of inversions

(or sometimes orientation-preserving products)

Forms group of geometric transformations

Contains all circle-preserving transformations

In higher dimensions (but not 2d) contains all conformal transformations

Optimal Möbius transformation:

Given a planar (or higher dimensional) input configuration

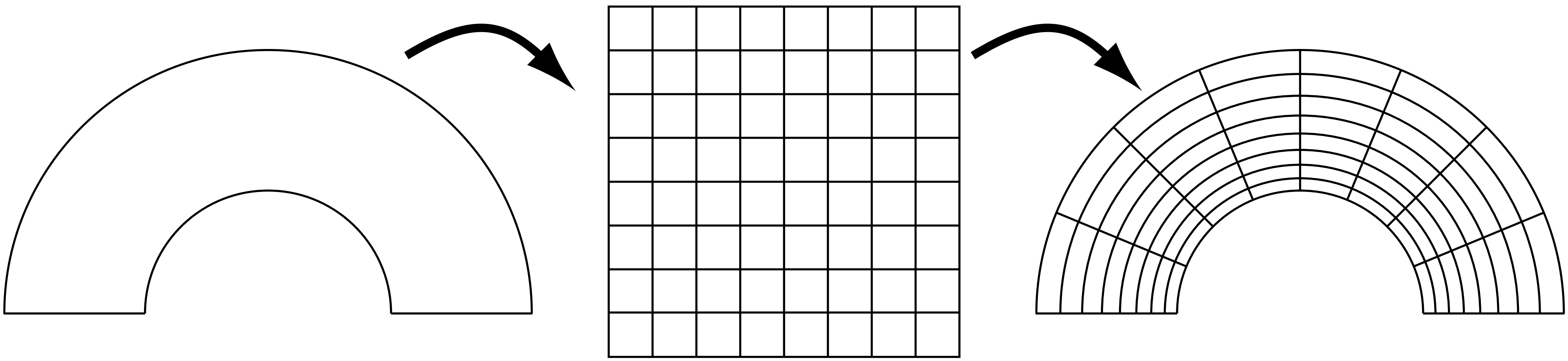
Select a Möbius transformation
from the (six-dimensional or higher) space of all Möbius transformations

That optimizes the shape of the transformed input

Typically min-max or max-min problems:
maximize min(set of functions describing transformed shape quality)

Application: conformal mesh generation

Given simply-connected planar domain to be meshed
Map to square, use regular mesh, invert map to give mesh in original domain



Different points of domain may have different requirements for element size
Want to map regions requiring small size to large areas of square

Conformal map is unique up to Möbius transformation

Optimization Problem:

Find conformal map maximizing min(size requirement * local expansion factor)
to minimize overall number of elements produced

Application: brain flat mapping [Hurdal et al. 1999]

Problem: visualize the human brain
Complicated folded 2d surface

Approach: find quasi-conformal mapping brain → plane
Avoids distorting angles but areas can be greatly distorted

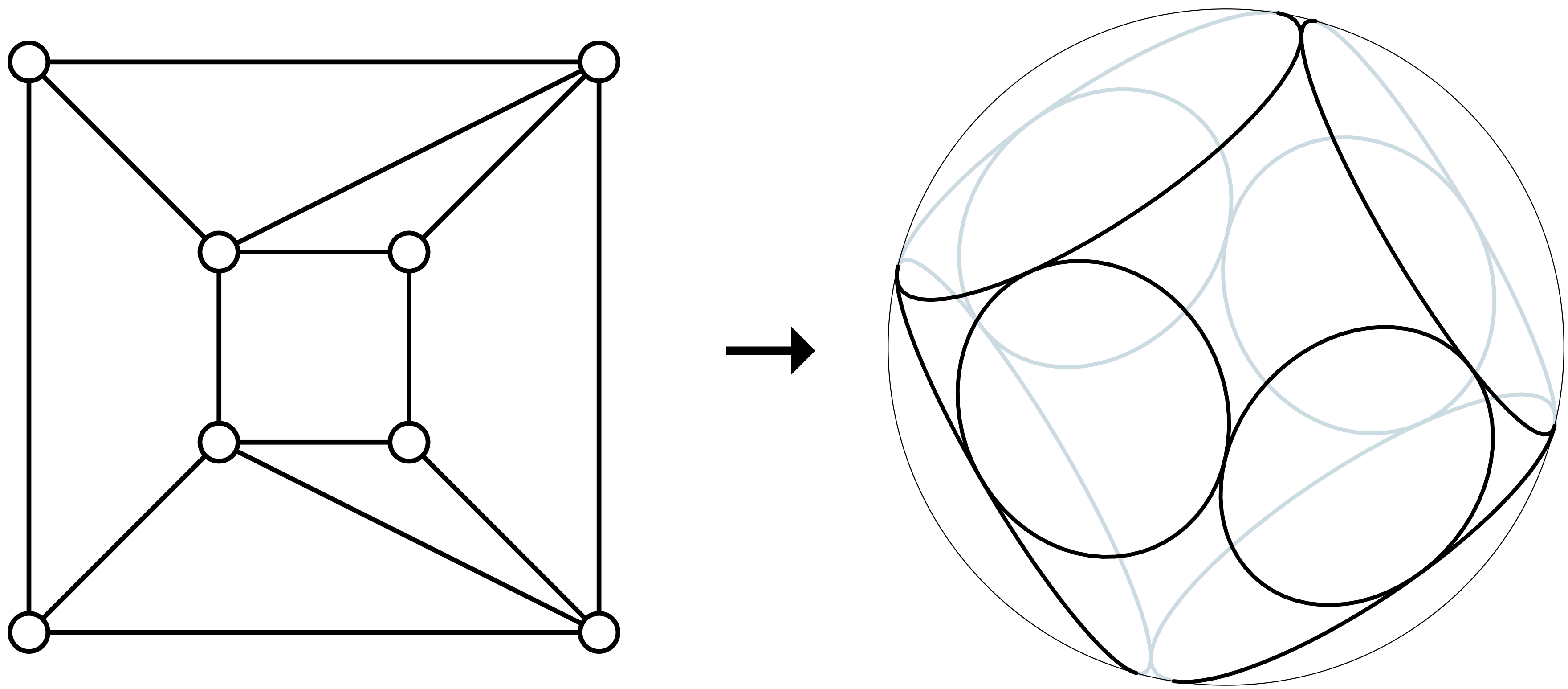
As in mesh gen. problem, mapping unique up to Möbius transformation

Optimization problem:

Given map 3d triangulated surface → plane,
find Möbius transformation minimizing max(area distortion of triangle)

Application: coin graph representation

Koebe-Andreev-Thurston Theorem:
vertices and edges of any planar graph can be represented
by disjoint disks and their tangencies on a sphere

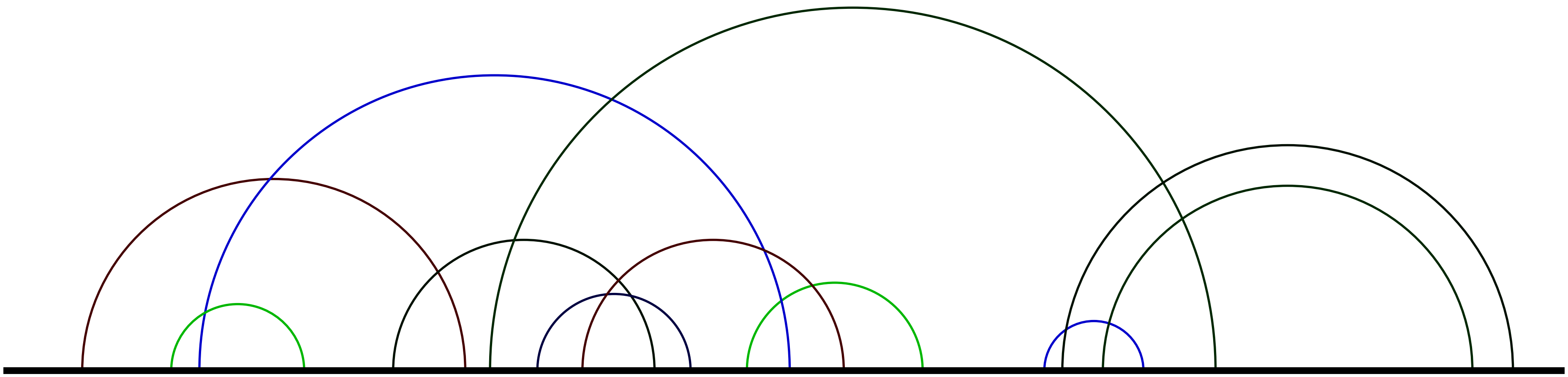


For maximal planar graphs, representation unique up to Möbius transformation

Problem: transform disks to maximize size of smallest disk
Uniqueness of optimal solution leads to display of graph symmetries

Hyperbolic interpretation of Möbius transformations

View d-dimensional space as boundary of Poincaré (halfspace or unit disk) model of hyperbolic $(d + 1)$-dimensional space



Möbius transformations of d-space $\leftrightarrow$ hyperbolic isometries

Simplify: optimal transformation → optimal location

View Möbius transformation as choice of
Poincaré model for hyperbolic space

Factor transformations into
choice of center point in hyperbolic model (affects shape)
Euclidean rotation around center point (doesn't affect shape)

Select optimal center point by quasiconvex programming

Klein model of hyperbolic geometry preserves convexity
so quasiconvex programming works equally well in hyperbolic space

Represent quality of Möbius transformation as max of quasiconvex functions
where function argument is hyperbolic center point location

Hard part: proving that our objective functions are quasiconvex

Result: can use QCP to find optimal Möbius transformations

Unclear: how to represent basis change operations for GLP algebraically?

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

Tiled projector systems

Problem: display system for collaborative workspaces
high resolution (≥ 6Mp), large scale (conference room wall)

Solution: combine output from multiple LCD projectors

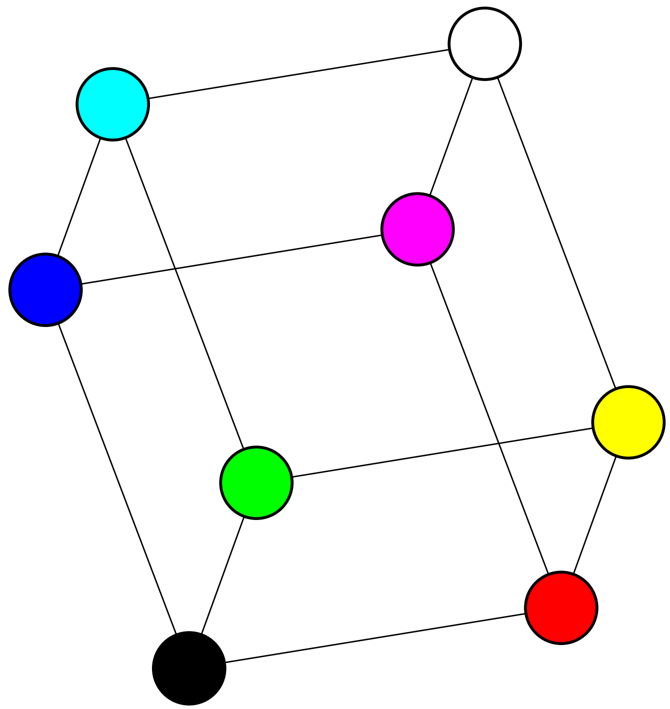
but...

Different projectors (even of same model)
have visible differences in gamut (set of available colors)

Want to construct a common gamut
of colors producable by all projectors in the system

Additive color

Gamuts of additive color devices (such as projectors)
form parallelepipeds in 3d color space



Want to find similarly shaped gamut within gamuts of all projectors

Problem: find large parallelepiped inside intersection of parallelepipeds

Volume-based approaches

Space of parallelepipeds is 12-dimensional (coordinates of four vertices)

$48n$ halfspace constraints (vertex of output within facet of input)

Max volume gamut: nonconvex, test each face of halfspace intersection: $O(n^6)$

Separate luminance (dark-light) and chrominance (color)
Find black and white points with same color value, max luminance ratio
Optimize volume in remaining 6-dimensional subspace: $O(n^3)$

Good enough to work with small numbers of projectors
But what if we want an input gamut for each pixel of each projector?

Quasiconvex approach (speculative)

Find eight 3d quasiconvex functions f_K, f_R, f_G, f_B, f_C, f_M, f_Y, f_W
measuring quality of each gamut corner location
Lift each function to 12*d* function of gamut location (still quasiconvex)

Add 48*n* halfspace constraints (quasiconvex step functions)

Quasiconvex program value = gamut optimizing worst color corner

Scales linearly with number of input gamuts
Can treat some colors (black, white) as more important than others

More colorimetric expertise needed: what qcf's to use?

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

A scary recurrence

$$T(n,h) \le \max \begin{cases} 2\,T(n+1,h+1) + T(n+2,h+1), \quad 2\,T(n+1,h+4) + 3\,T(n+2,h+2) + 3\,T(n+2,h+3), \\ 6\,T(n+2,h+2), \quad 2\,T(n+2,h) + T(n+3,h+1) + T(n+4,h) + T(n+4,h+1), \\ T(n+2,h) + T(n+3,h-2) + T(n+3,h-1), \quad 2\,T(n+2,h) + 2\,T(n+2,h+3), \\ T(n+3,h-2) + T(n+3,h-1) + T(n+5,h-3) + T(n+5,h-2) + T(n+7,h-3), \\ T(n+1,h-1) + T(n+4,h-1), \quad T(n+1,h) + T(n+3,h-1) + 3\,T(n+3,h+3), \\ T(n+3,h-2) + T(n+3,h-1) + T(n+4,h-2), \quad T(n+3,h-2) + 6\,T(n+3,h+2), \\ T(n+3,h-2) + T(n+3,h-1) + 4\,T(n+5,h-1), \quad T(n+1,h-1) + T(n+4,h-1), \\ T(n+1,h) + T(n+1,h+1), \quad T(n+2,h+2) + 5\,T(n+3,h+1) + T(n+4,h), \\ 2\,T(n+3,h-1) + 2\,T(n+3,h) + 2\,T(n+3,h+3), \quad T(n,h+1) + T(n+4,h-3), \\ T(n+3,h-2) + 2\,T(n+3,h-1) + T(n+6,h-3), \quad 3\,T(n+1,h+2) + 2\,T(n+1,h+5), \\ T(n+1,h), \quad 2\,T(n+3,h) + 2\,T(n+3,h+3) + T(n+4,h-3) + T(n+4,h-2), \\ T(n+1,h+2) + T(n+3,h-2) + T(n+3,h-1), \quad T(n+2,h-1) + 2\,T(n+3,h-1), \\ T(n+1,h) + T(n+3,h-1) + T(n+3,h+3) + T(n+5,h) + T(n+6,h-1), \quad T(n-1,h+2), \\ T(n+3,h-2) + T(n+3,h-1) + T(n+5,h-3) + T(n+6,h-3) + T(n+7,h-4), \\ 2\,T(n+3,h-1) + T(n+3,h+2) + T(n+5,h-2) + T(n+5,h-1) + T(n+5,h) + 2\,T(n+7,h-3), \\ T(n+2,h-1) + T(n+2,h) + T(n+4,h-2), \quad T(n+2,h-1) + T(n+3,h-1) + T(n+4,h-2), \\ T(n+3,h-2) + 2\,T(n+3,h-1), \quad 4\,T(n+2,h+3) + 3\,T(n+4,h) + 3\,T(n+4,h+1), \\ 8\,T(n+1,h+4), \quad 2\,T(n+2,h) + T(n+3,h) + T(n+4,h) + T(n+5,h-1), \\ T(n+2,h) + T(n+3,h-2) + T(n+3,h-1), \quad T(n+3,h-2) + 2\,T(n+4,h-2) + T(n+5,h-3), \\ T(n,h+1), \quad T(n+1,h-1), \quad 2\,T(n+2,h) + T(n+2,h+3) + T(n+3,h) + T(n+3,h+2), \\ 2\,T(n+3,h-1) + T(n+3,h+2) + T(n+5,h-2) + T(n+5,h-1) + T(n+5,h) + T(n+6,h-2) + T(n+7,h-2), \\ 9\,T(n+9,h-5) + 9\,T(n+9,h-4), \quad 2\,T(n+2,h+1) + 3\,T(n+2,h+3) + 2\,T(n+2,h+4), \\ T(n+2,h-1) + T(n+2,h), \quad T(n+3,h-2) + T(n+3,h-1) + T(n+5,h-2) + 2\,T(n+6,h-3), \\ T(n+4,h-3) + 2\,T(n+4,h-2) + T(n+7,h-4), \quad T(n+3,h-2) + T(n+3,h-1) + 2\,T(n+5,h-2), \\ 2\,T(n+2,h-1), \quad 2\,T(n,h+2), \quad T(n+2,h+2) + 2\,T(n+3,h) + T(n+3,h+1) + 3\,T(n+4,h), \\ T(n+2,h-1) + T(n+2,h) + T(n+5,h-2), \quad T(n+1,h-1) + T(n+2,h+2), \\ T(n+2,h) + T(n+3,h), \quad 10\,T(n+3,h+2), \quad 5\,T(n+2,h+2) + 2\,T(n+2,h+3), \\ T(n+3,h-2) + T(n+3,h-1) + T(n+4,h-2) + T(n+6,h-3), \quad 3\,T(n,h+3), \\ 6\,T(n+3,h+1), \quad 9\,T(n+2,h+3), \quad T(n,h+1) + T(n+2,h), \quad 2\,T(n+5,h-3) + 5\,T(n+5,h-2) \end{cases}$$

from unpublished joint work with J. Byskov on graph coloring algorithms

Where does this recurrence come from?

Backtracking algorithms for NP-hard problems such as graph coloring or SAT

Repeat:
- Find a decision to be made
- Split into subproblems
- Solve each subproblem recursively

E.g. for listing all independent subsets of a path:
either exclude the path endpoint (one fewer vertex)
or include the endpoint and exclude its neighbor (two fewer vertices)
$T(n) = T(n - 1) + T(n - 2)$ = Fibonacci numbers

Why is the recurrence so complicated?

Intricate case analysis to find decision leading to small subproblems

Each case leads to term like $T(n - 1) + T(n - 2)$
Worst case analysis means we have to take max of terms

Multiple measures of subproblem instance size
lead to recurrences in more than one variable

E.g. modify independent set problem to list independent sets of $\leq k$ vertices
Parameters are number of vertices (n), target set size (k)

Graph coloring: count numbers of vertices with different available colors

Traveling salesman problem: vertices, forced edges, more complex features

What do we want to find out?

Upper bounds: $T(n, h) = O(1.7780544^{n + 0.660703h})$

Lower bounds: upper bound is within polynomial factor of tight when $h = 0$

Sensitivity analysis: solution is dominated by two terms

$T(n - 2, h) + T(n, h - 1)$

and

$2\,T(n - 3, h + 1) + T(n - 3, h + 2) + T(n - 6, h + 3)$

Exploratory research: need fast solution, numerical approximation ok

Published worst case bounds: correctness critical (exact real arithmetic)

Upper bound technique

Given recurrence $T(\boldsymbol{x})$, $\boldsymbol{x}$ in $\mathbf{Z}^d$, and test vector $\boldsymbol{x}_0$
we want asymptotic behavior of $T(n\,\boldsymbol{x}_0)$ for large n

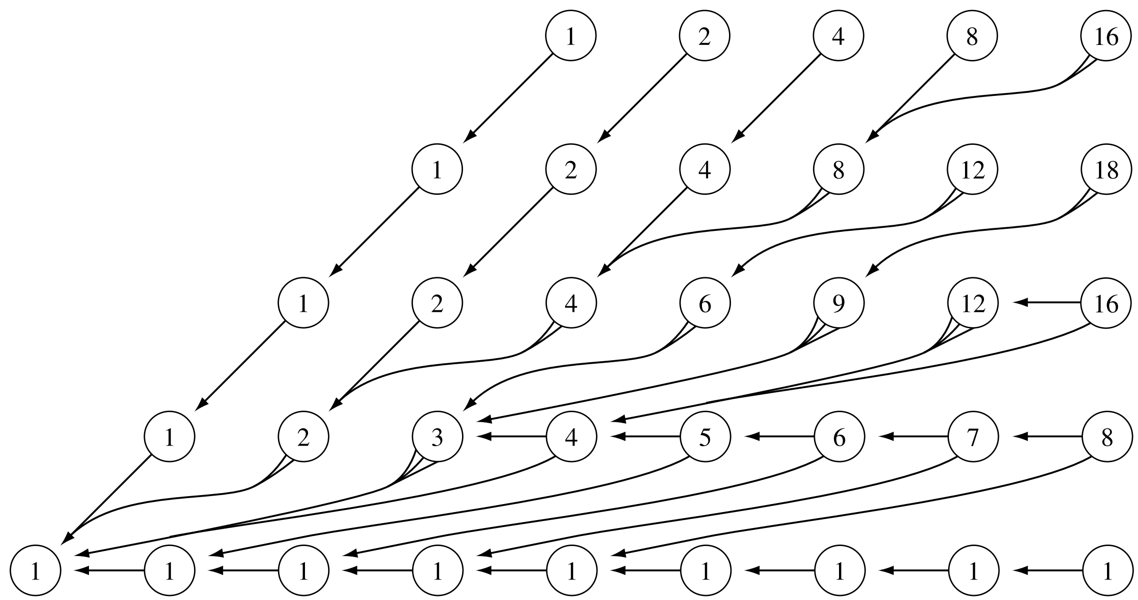
Assume solution has form $O(c^{\boldsymbol{w} \cdot \boldsymbol{x}})$
where $\boldsymbol{w} \cdot \boldsymbol{x}$ is some weighted combination of recurrence variables

For each term t_i, define quasiconvex function
$f_i(\boldsymbol{w})$ = minimum c s.t. $c^{\boldsymbol{w} \cdot \boldsymbol{x}}$ satisfies one-term recurrence

Find $\boldsymbol{w}$ with $\boldsymbol{w} \cdot \boldsymbol{x}_0 = 1$ minimizing $c = \max f_i(\boldsymbol{w})$
Gives best possible bound $T(n\,\boldsymbol{x}_0) = O(c^n)$ of assumed form
(but different test vectors may give different bounds)

Lower bound technique

Interpret recurrence as #paths to origin in an infinite graph on $\mathbf{Z}^d$
Connection pattern from vertex $\boldsymbol{x}$ is determined by term giving max for $\boldsymbol{x}$



Modify graph by choosing connection pattern randomly
Perform random walk from $\boldsymbol{x}_0$ on replacement graph

Gradiants from smooth QCP algorithm surround origin →
can choose appropriate connection pattern and walk probabilities →
polynomial fraction of random walks from $n\ \boldsymbol{x}_0$ reach the origin
and probability of taking any particular walk is $c^{\boldsymbol{w} \cdot \boldsymbol{x}}$

Conclusions

QCP has many varied applications

Applicable in hyperbolic as well as Euclidean geometry

Avoids difficulty of exact basis change calculations
by allowing efficient numerical solutions

Open problems

Recurrence lower bound technique hints at theory of QCP duality?

Generalized Voronoi diagram of optimal bases for parametrized problem?
e.g. in recurrence problem avoid need for test vector

Ellipsoid method or other more sophisticated LP techniques?
Can confine optimal point to low-volume ellipsoid
But when is volume small enough to jump to unique basis?

References

Amenta, Bern, and Eppstein.
Optimal point placement for mesh smoothing.
SODA 1997, 528–537; *J. Algorithms* 1999, 30:302–322.

Bern and Eppstein.
Optimal Möbius transformations for information visualization and meshing.
WADS 2001, 14–25.

Bern and Eppstein.
Optimized color gamuts for tiled displays.
SoCG 2003, to appear.

Eppstein.
Quasiconvex analysis of backtracking algorithms.
Preprint, 2003.