Quasiconvex Programming

David Eppstein

Univ. of California, Irvine
Donald Bren School of Information and Computer Sciences

Overview: What is quasiconvex programming?

Formulation for low-dimensional min-max optimization problems

Leads to efficient numerical and combinatorial algorithms

Many applications

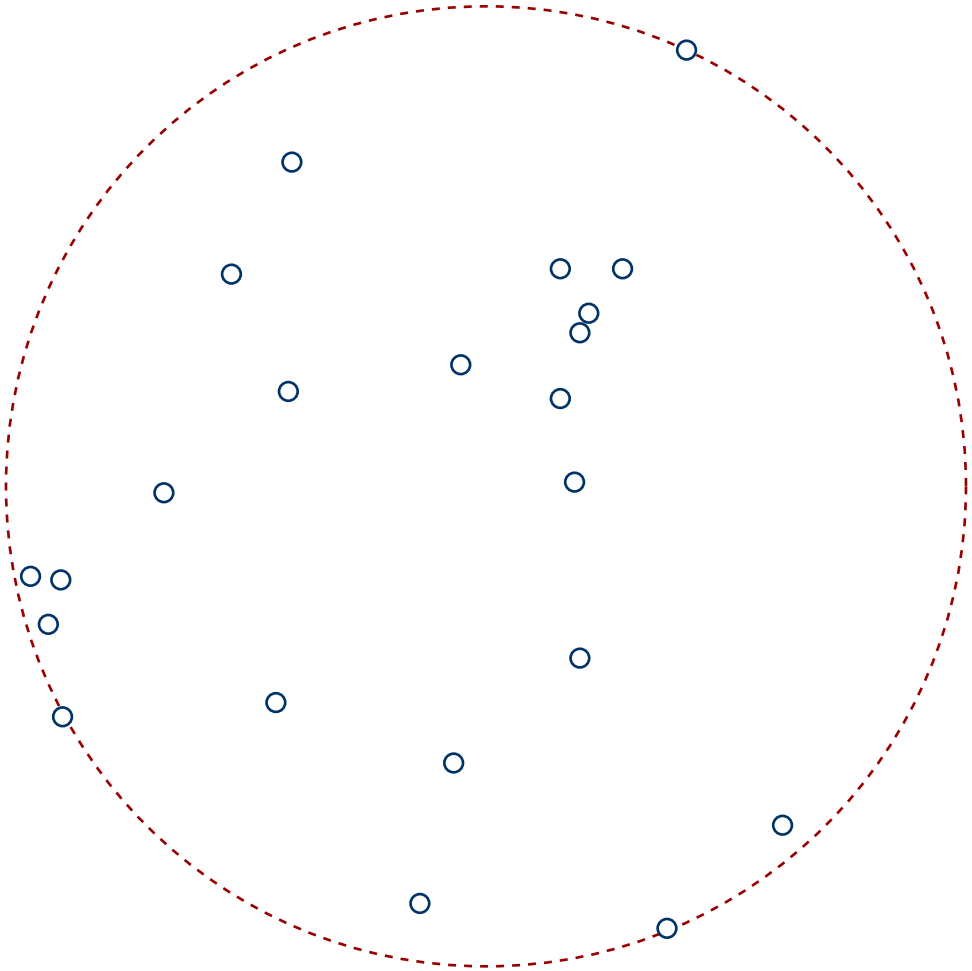
Outline

- Minimum enclosing disk example
- Quasiconvex programming
- Mesh smoothing
- Optimal Möbius transformation
- Color gamut optimization
- Analysis of backtracking algorithms
- Implicit quasiconvex programming
- Robust statistics

Outline

- **Minimum enclosing disk example**
- Quasiconvex programming
- Mesh smoothing
- Optimal Möbius transformation
- Color gamut optimization
- Analysis of backtracking algorithms
- Implicit quasiconvex programming
- Robust statistics

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]

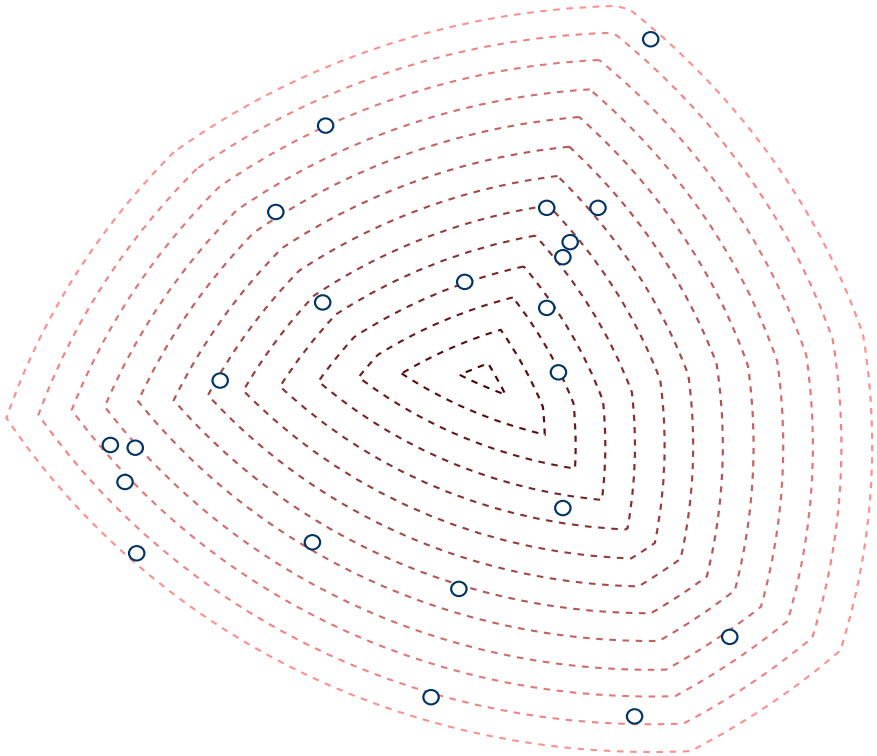
Typically: $O(n)$ feasibility tests (is point inside current circle?)
$o(n)$ basis change operations (find circumradius of constant size subset)

Minimum enclosing disk as quasiconvex program

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

For any x, min enclosing disk centered at x has radius $\max_p d_p(x)$

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



Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

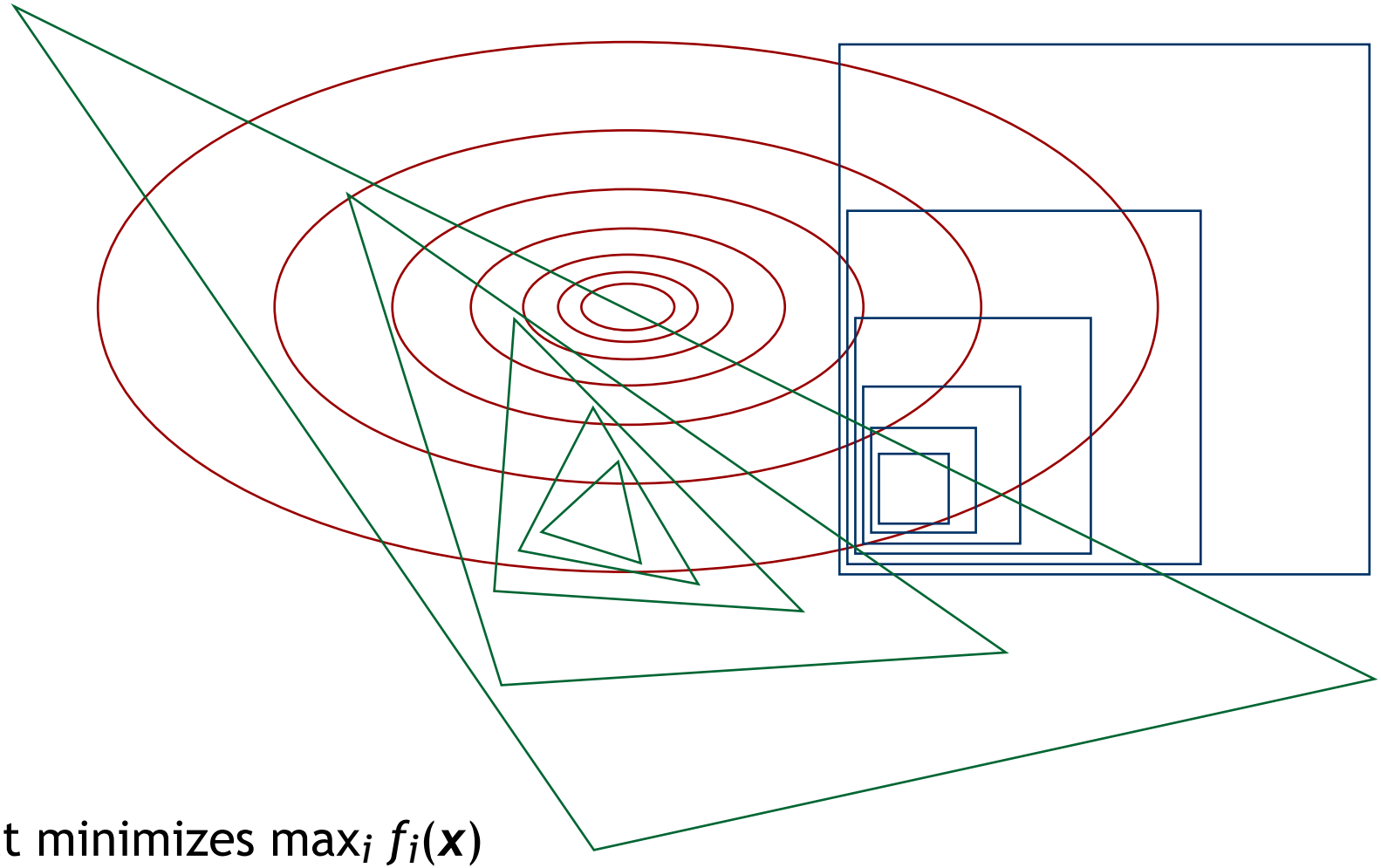
Analysis of backtracking algorithms

Implicit quasiconvex programming

Robust statistics

Quasiconvex program:

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



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

Why use quasiconvex program formulation?

Level of abstraction between convex programs and generalized LP

Includes problems that do not form convex programs

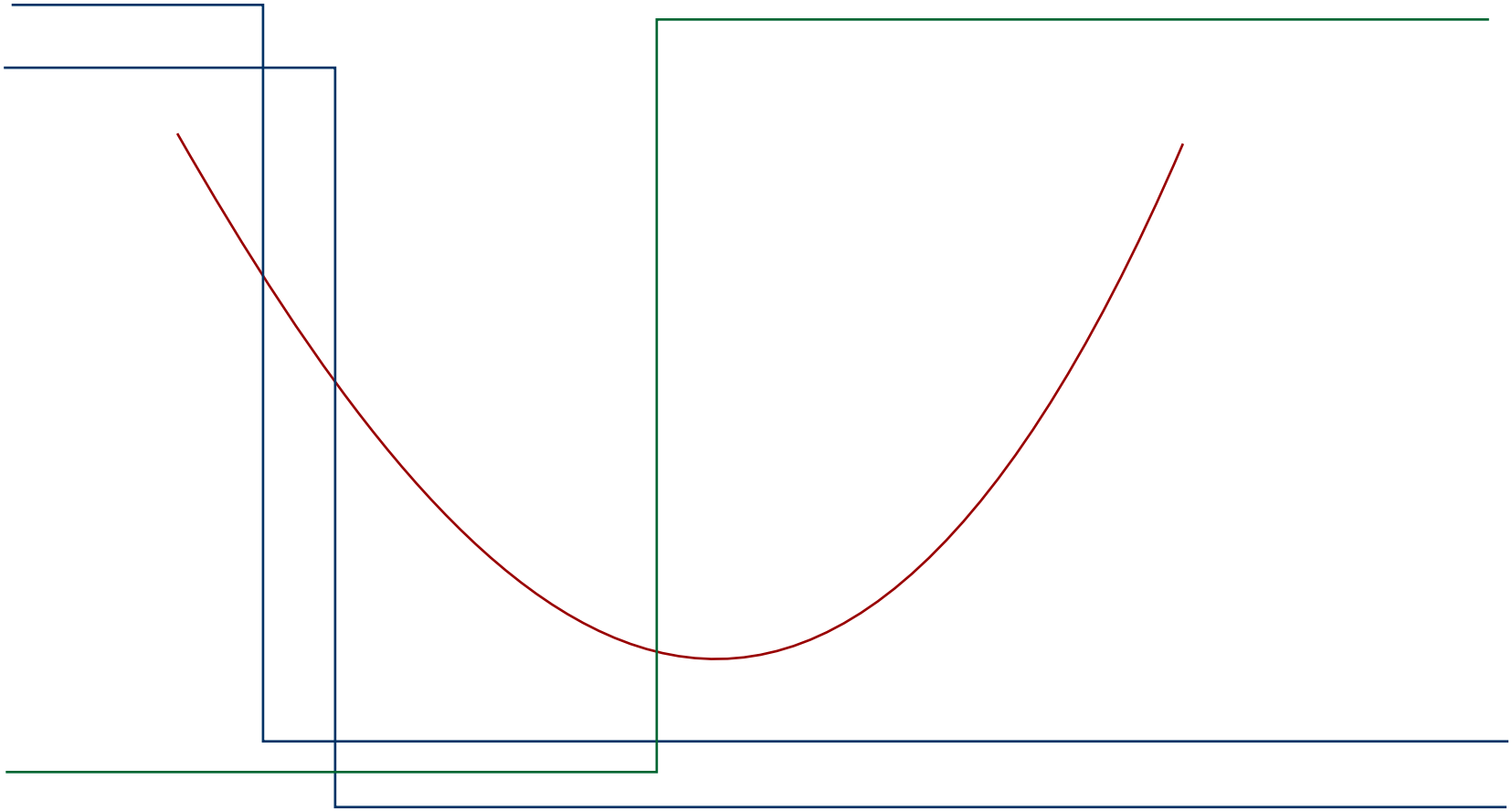
Quasiconvexity may be easier to prove than GLP axioms

Unlike GLP, still permits numerical solution techniques

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)

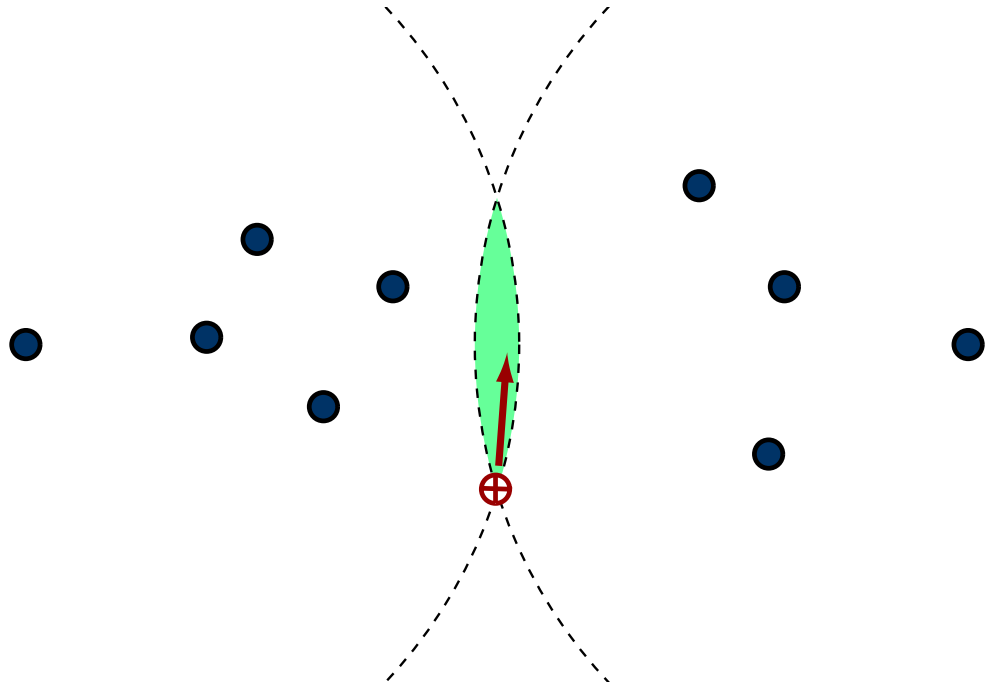
...so can use GLP algorithms if basis change operation can be implemented

(Amenta, Bern, Eppstein, J. Algorithms 1999)

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

(Eppstein, SODA 2004)

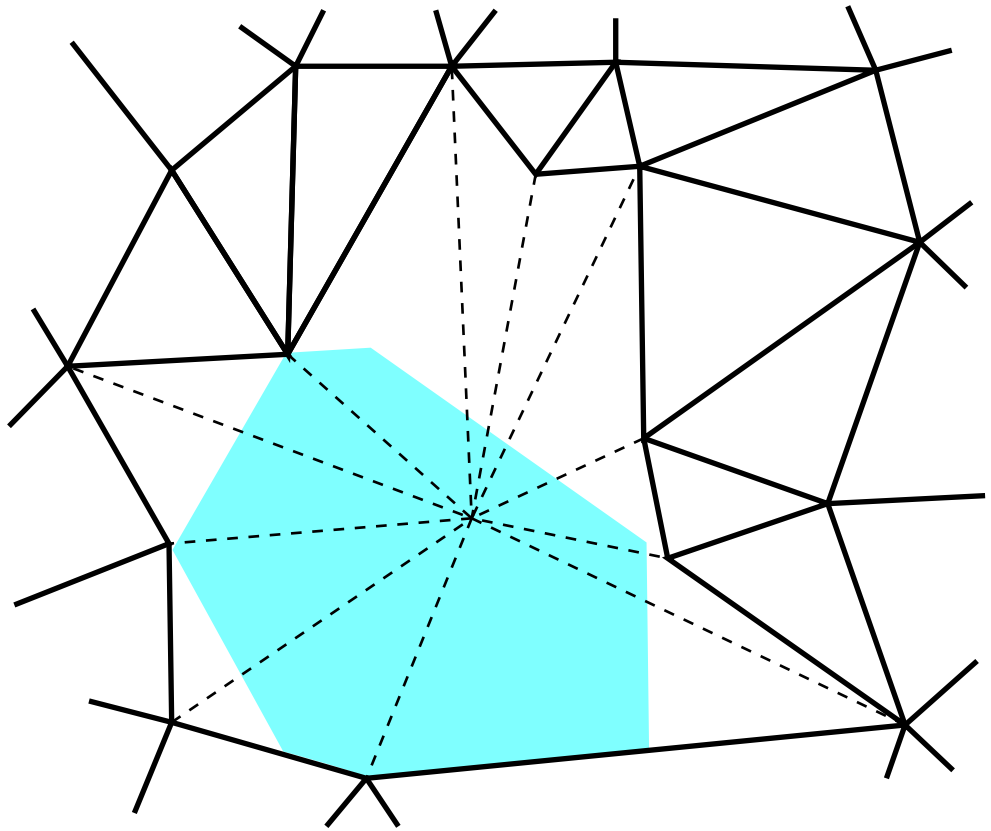
Outline

- Minimum enclosing disk example
- Quasiconvex programming
- **Mesh smoothing**
- Optimal Möbius transformation
- Color gamut optimization
- Analysis of backtracking algorithms
- Implicit quasiconvex programming
- Robust statistics

Mesh improvement problem

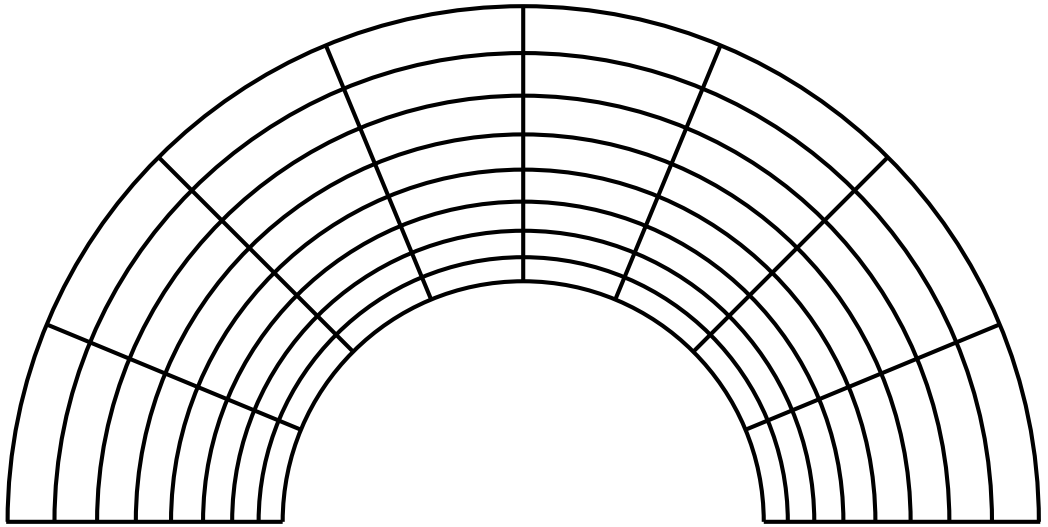
Finite element meshes produced by standard methods (e.g. quadtree) are insufficiently regular (many 45-45-90 triangles, prefer equilateral)

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

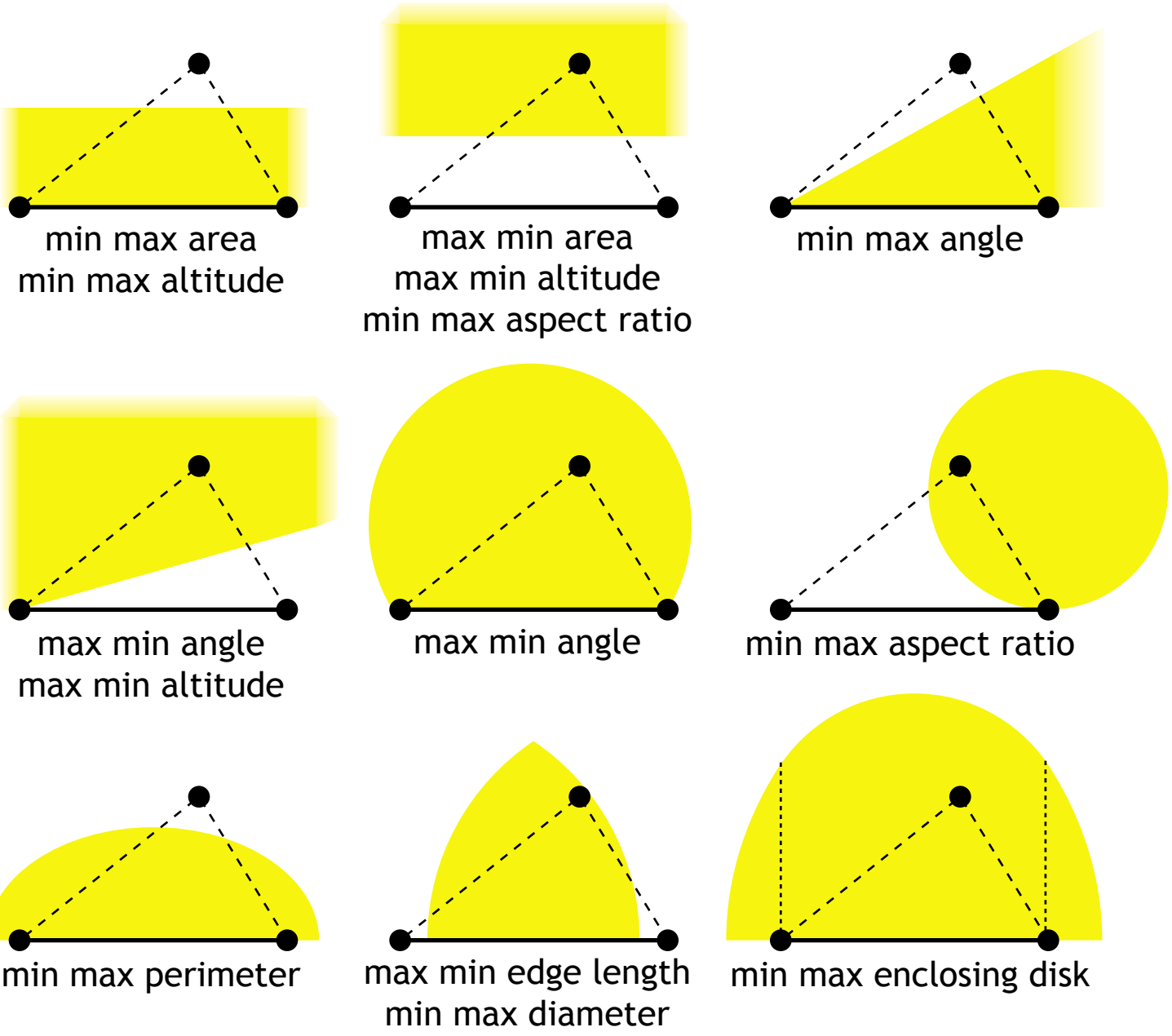
Typical mesh quality measures are quasiconvex
so vertex location problem becomes quasiconvex program

Many possible choices for mesh quality measure
don't affect smoothing efficiency
so can choose by effect on finite element solution quality

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

(Amenta, Bern, Eppstein, J. Algorithms 1999)

Element quality measures and level set shapes



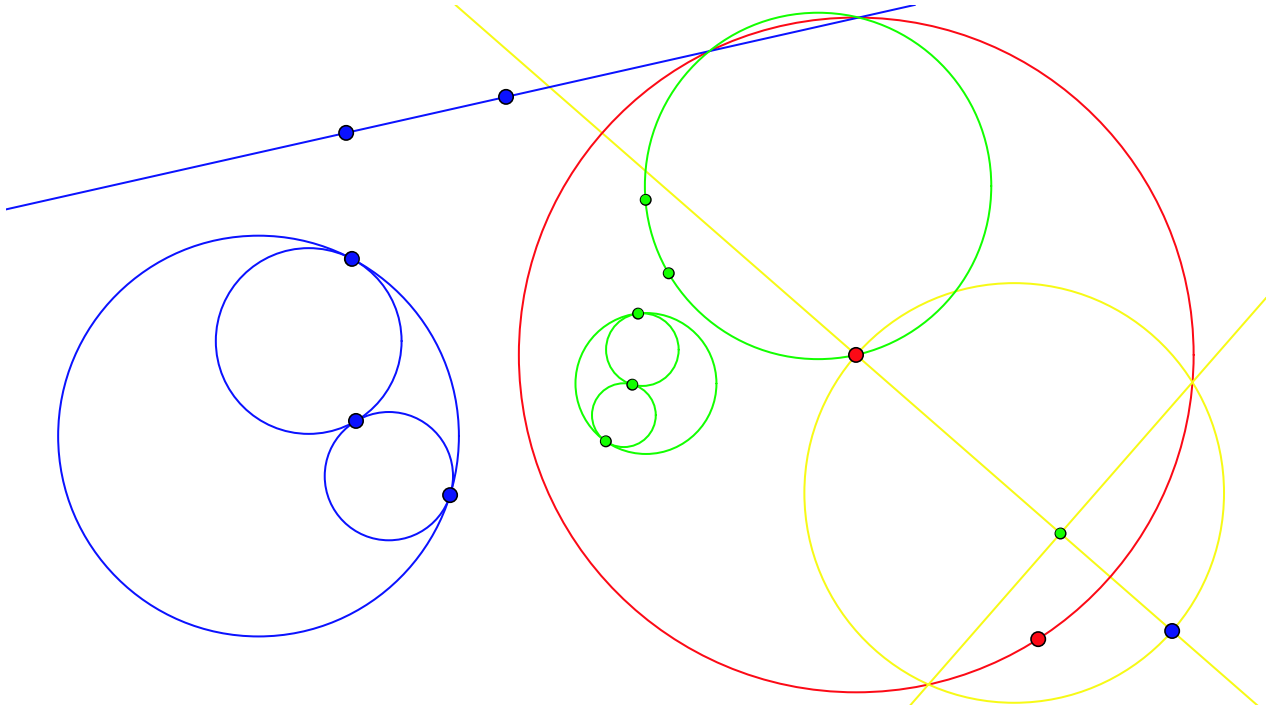
Similar but more complicated in three dimensions...

Outline

- Minimum enclosing disk example
- Quasiconvex programming
- Mesh smoothing
- **Optimal Möbius transformation**
- Color gamut optimization
- Analysis of backtracking algorithms
- Implicit quasiconvex programming
- Robust statistics

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

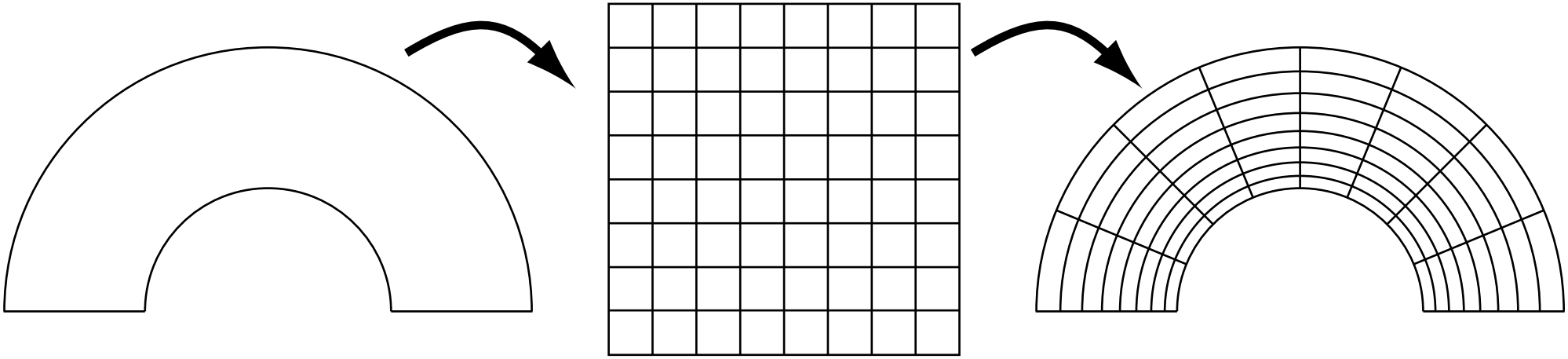
Select the Möbius transformation
that optimizes the shape of the transformed input

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

Can often represent optimal Möbius transformation problem
as quasiconvex program in space of one higher dimension
(via hyperbolic geometry; see Bern and Eppstein, WADS 2001, for details)

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

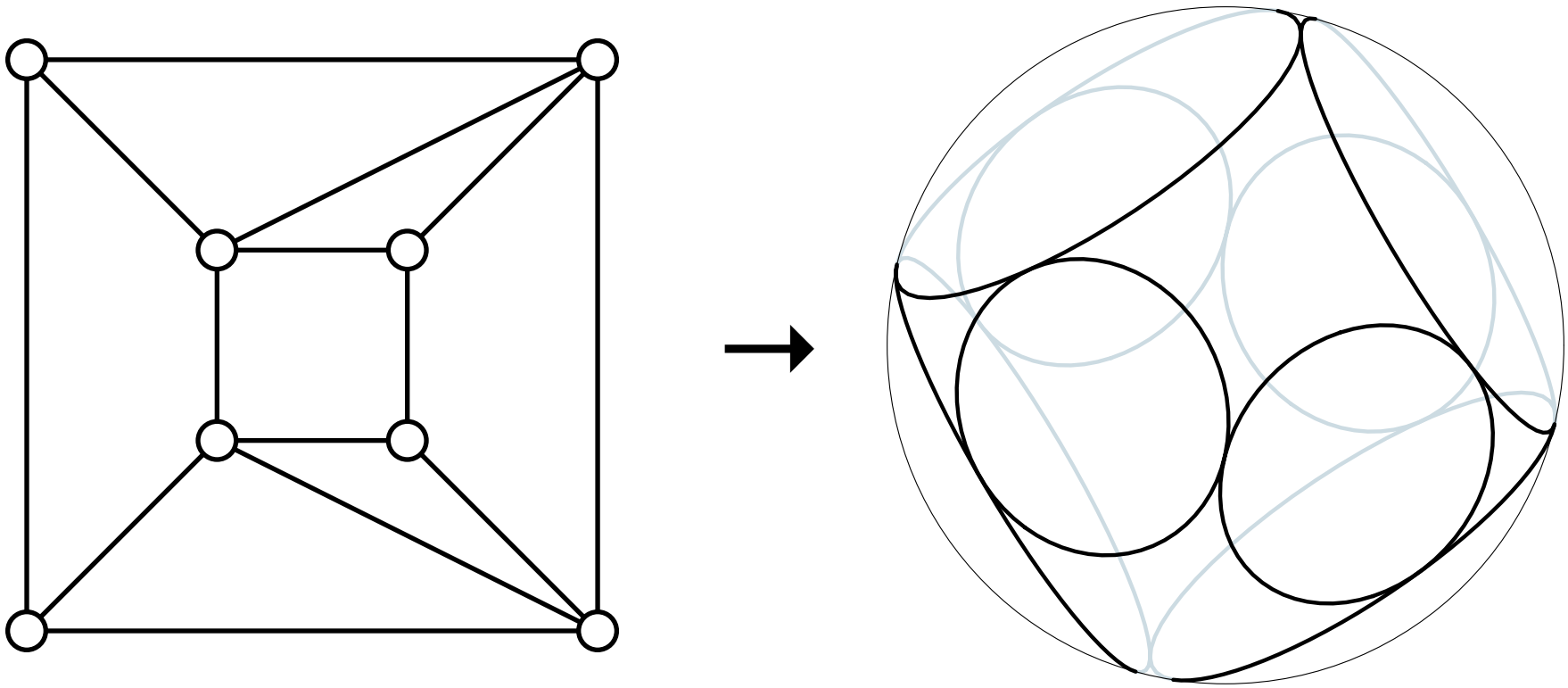
As in mesh gen. problem, mapping unique up to Möbius transformation

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

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

Analysis of backtracking algorithms

Implicit quasiconvex programming

Robust statistics

Tiled projector systems

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

Solution: combine output from multiple LCD projectors

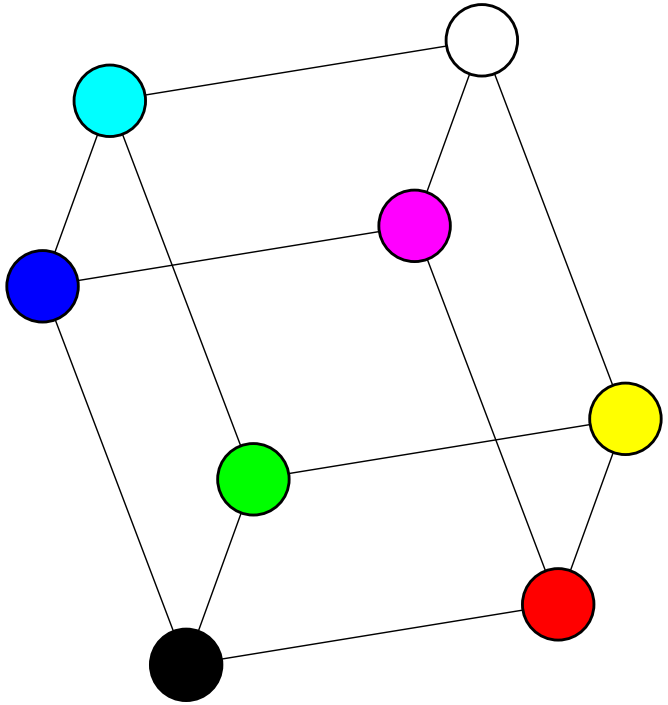
but...

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

Quasiconvex gamut optimization

Represent gamut (parallelepiped) as point in 12-dimensional space

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

Add $48n$ halfspace constraints (quasiconvex step functions)
forcing output gamut to be contained in each input gamut

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?

(Bern, Eppstein, SoCG 2003)

Outline

- Minimum enclosing disk example
- Quasiconvex programming
- Mesh smoothing
- Optimal Möbius transformation
- Color gamut optimization
- **Analysis of backtracking algorithms**
- Implicit quasiconvex programming
- Robust statistics

A scary recurrence

$$T(n,h) \le \max \left\{ \begin{array}{l}
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{array} \right.$$

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 ≤ 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)

Recurrence solution technique

Replace single multi-term recurrence in multiple variables $x_0,x_1,x_2,\ldots$
by multiple single-term recurrences in weighted combination of variables
$y = w_0x_0 + w_1x_1 + w_2x_2 + \ldots$

Solution of j'th recurrence has form $T_j(y) = c_j^y$
Use $(\max_j c_j)^y$ as our overall bound

Compute weights $w_0,w_1,w_2,\ldots$ to give tightest possible bound
QCP, one function $c_j = f(w_0,w_1,w_2,\ldots)$ per term of recurrence
Optimal basis gives dominant terms of recurrence

Can prove that resulting solution is within polynomial factor of optimal

(Eppstein, SODA 2004)

Outline

- Minimum enclosing disk example
- Quasiconvex programming
- Mesh smoothing
- Optimal Möbius transformation
- Color gamut optimization
- Analysis of backtracking algorithms
- **Implicit quasiconvex programming**
- Robust statistics

Implicit quasiconvex program

Defined by a function Q mapping inputs to sets of quasiconvex functions

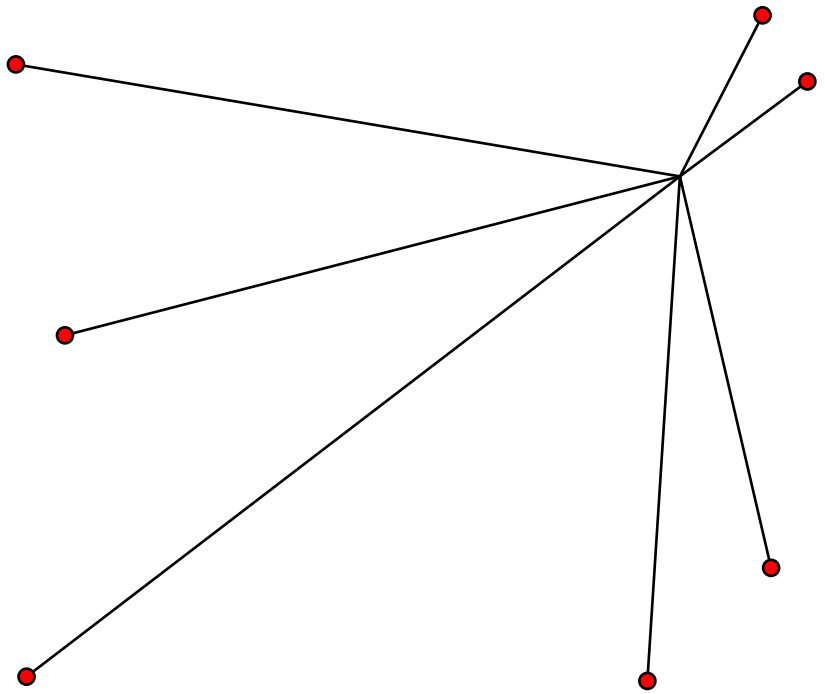
Q(Input) may be much larger than the input itself

Could solve by computing Q then applying any QCP algorithm

Chan [SODA 2004] showed many implicit QCPs can be solved more efficiently using a ***decision oracle*** and a ***subdivision process***

Example implicit QCP: Hub placement (minimum dilation star)

(Eppstein and Wortman, unpublished)



Input: point set S

Want to find location for new hub x

Minimize
max {(|sx|+|xt|)/|st| for s, t in S}

$O(n^2)$ functions (|sx|+|xt|)/|st|
one per pair of input points

Level sets are ellipses, so quasiconvex

Decision oracle

Given implicit QCP input S, point x, value y

Is there a function f in Q(S) with f(x) > y?

Example: decision oracle for hub placement

Given point set S, possible hub x, ratio y
Is max {(|sx|+|xt|)/|st| for s, t in S} > y?

Maximizing pair s,t are among O(1) nearest neighbors of each other
or
they are O(1) steps apart in sorted list of distances from x

So, can compute max and compare with y in time O(n log n)

Subdivision process

Need constants r=O(1), a<1
s.t.
Any input A can be subdivided into r smaller inputs A_0, A_1, A_2, ...
Each A_i is of size at most a•size(A)
Q(A) = union of $Q(A_i)$

Example subdivision process: hub placement

Each quasiconvex function is determined by some pair of points

Partition input S into three equal subsets S_0, S_1, S_2

Define subproblem $A_i = S \setminus S_i$

r=3, a=2/3

Chan's implicit QCP algorithm (Chan, SODA 2004)

Subdivide into subproblems

Apply generalized linear programming
- objects = subproblems
- objective function = value of union of QCPs

Result: randomized implicit QCP algorithm
expected time = O(decision oracle + subdivision process)

(with mild assumptions on that time bound)

Example: hub placement
- Decision oracle takes time O(n log n)
- Subdivision process takes time O(n)
- So can find optimal placement in O(n log n) time

Outline

Minimum enclosing disk example

Quasiconvex programming

Mesh smoothing

Optimal Möbius transformation

Color gamut optimization

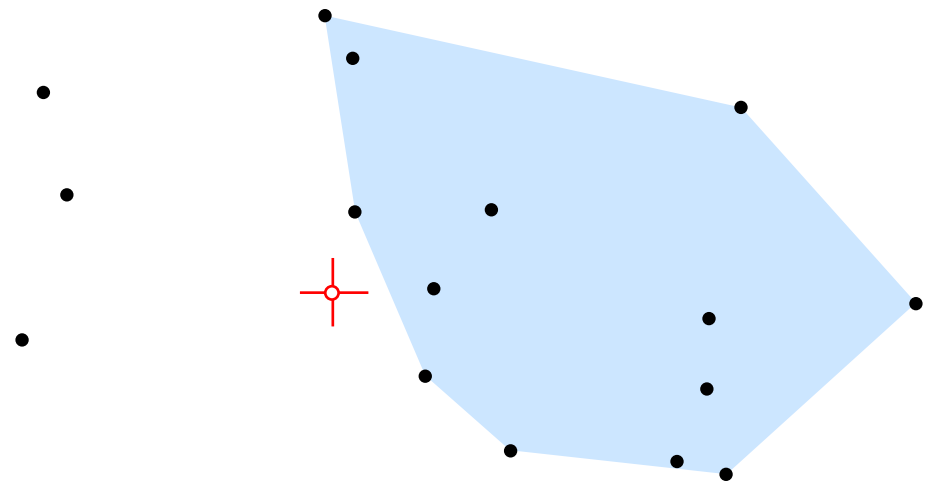
Analysis of backtracking algorithms

Implicit quasiconvex programming

Robust statistics

Tukey depth

Depth of x with respect to point set S
= min |T| s.t. x is outside convex hull of S\T
= min |S intersect H| s.t. halfspace H contains x



Tukey median = point of maximum Tukey depth

Statistical estimator of location of point cloud
robust against large number of arbitrary outliers

Also useful in mesh partitioning algorithms (Teng et al.)

Tukey median as quasiconvex program

For each halfspace H, define function $q_H(x)$
= -|S intersect H| if x in H
= -|S| otherwise

$q_H(x)$ is quasiconvex (level sets are halfspace and whole space)

Tukey depth of x = $-\max_H q_H(x)$

Tukey median = point minimizing $\max_H q_H(x)$

Problem: too many quasiconvex functions

Infinitely many as defined above

$O(n^d)$ if we consider only combinatorially distinct halfspaces

Tukey median as implicit quasiconvex program [Chan, SODA 04]

Subdivision process:

Difficulty: qcfs depend on more than O(1) points each
So can't use simple subdivision like hub problem

Epsilon-cutting: triangulate dual hyperplane arrangement
O(1) simplices, each crossed by epsilon*n hyperplanes

One subproblem per simplex in the cutting

Decision oracle: find max $q_H(x)$ among hyperplanes H within simplex

H maximizing $q_H(x)$ lies on simplex boundary or on dual(x)
Intersect arrangement with boundary+dual(x), test each cell

Overall result:

$O(n^{d-1} + n \log n)$ expected time randomized algorithm

Conclusions

QCP has many varied applications

Applicable in hyperbolic as well as Euclidean geometry

Allows both efficient numerical solutions
and linear-time exact algorithms
(if basis change operations can be implemented)