# Quasiconvex Analysis of Backtracking Algorithms

**David Eppstein**

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

## A scary recurrence

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

from unpublished joint work with J. Byskov on graph coloring algorithms

## Outline

Where do these recurrences come from and what are they good for?

A method for calculating upper bounds

Algorithms for performing the upper bound technique

Matching lower bounds

Conclusions and open problems

## 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

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

## Two modes of operation:

Exploratory research: need fast solution, numerical approximation ok

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

## Upper bound technique

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

Assume solution has form $T(x) = t(\boldsymbol{w} \bullet \boldsymbol{x})$
where $\boldsymbol{w} \bullet \boldsymbol{x}$ is some weighted combination of recurrence variables
and replace T by t on both sides of recurrence

Resulting single-variable recurrence is easy to solve (e.g. generating fns)
resulting in valid but non-tight upper bound

Different choices of $\boldsymbol{w}$ give different bounds, so choose $\boldsymbol{w}$ carefully

## Upper bound technique (details)

For each term $t_i$ in the recurrence maximization,
form single-variable linear recurrence with just that term,
having solution $O(f_i(\boldsymbol{w})^{\boldsymbol{w} \cdot \boldsymbol{x}})$

Solution to combined recurrence $t(\boldsymbol{w} \bullet \boldsymbol{x})$ is $O(c^{\boldsymbol{w} \cdot \boldsymbol{x}})$ where $c = \max_i f_i(\boldsymbol{w})$

Search for $\boldsymbol{w}$ with $\boldsymbol{w} \bullet \boldsymbol{x}_0 = 1$ minimizing $\max f_i(\boldsymbol{w})$

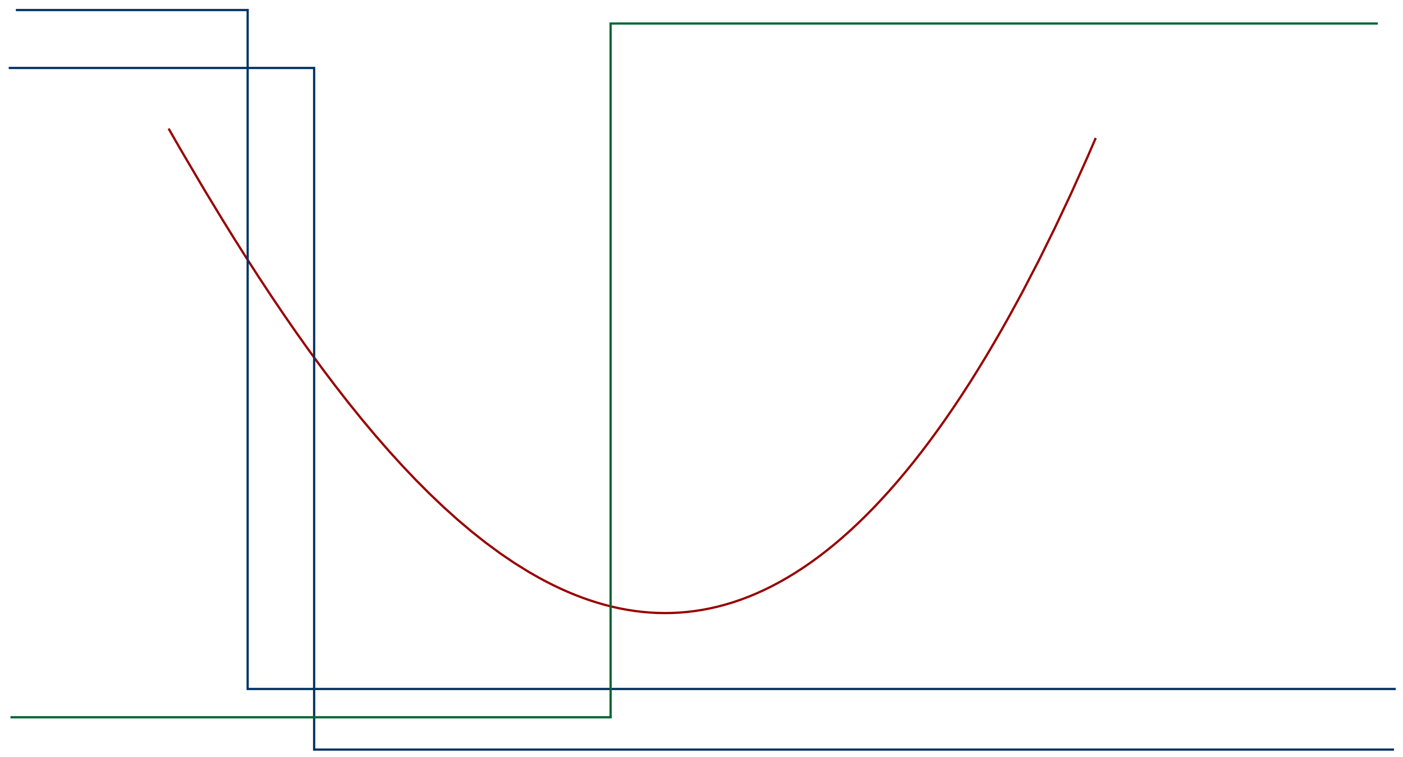
The functions $f_i$ are *quasiconvex*

**This is a *quasiconvex program***

## Quasiconvex Functions

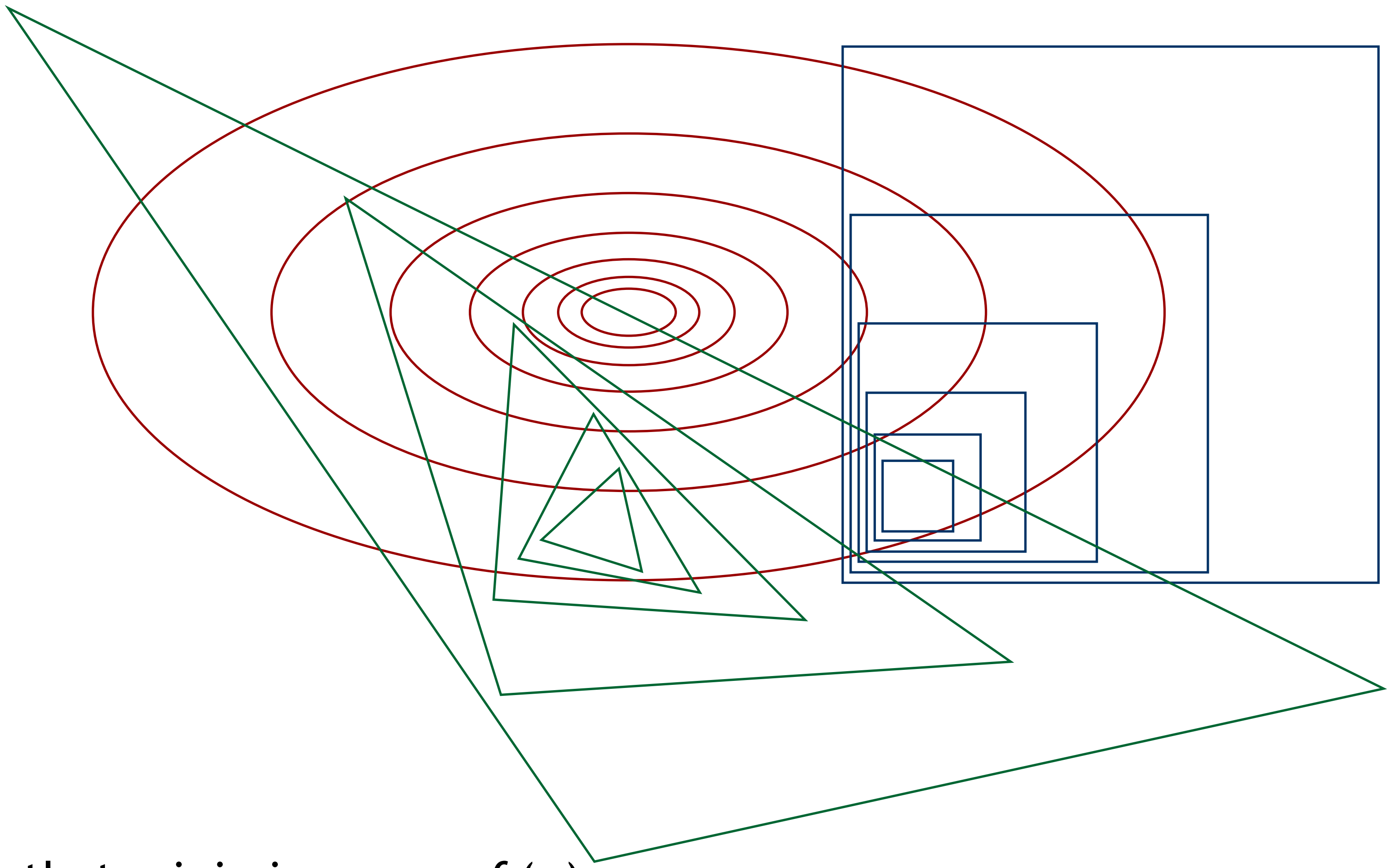
Function $f$ from $\mathbf{R}^d$ to $\mathbf{R}$ is *quasiconvex*
if the lower level sets $\{\boldsymbol{x}: f(\boldsymbol{x}) \leq r\}$ are convex

Examples: convex functions,
step functions (1 inside halfspace, 0 outside)



## Quasiconvex program:

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



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

Introduced & applied by Amenta, Bern, Eppstein for
mesh smoothing, graph drawing, brain flat mapping, color gamut optimization...

## Quasiconvex program as generalized linear program

For any subset S of input, let $f(S)$ = solution to QCP for S

This satisfies GLP axioms:

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

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 $f(S) = f(S \setminus \{x\})$

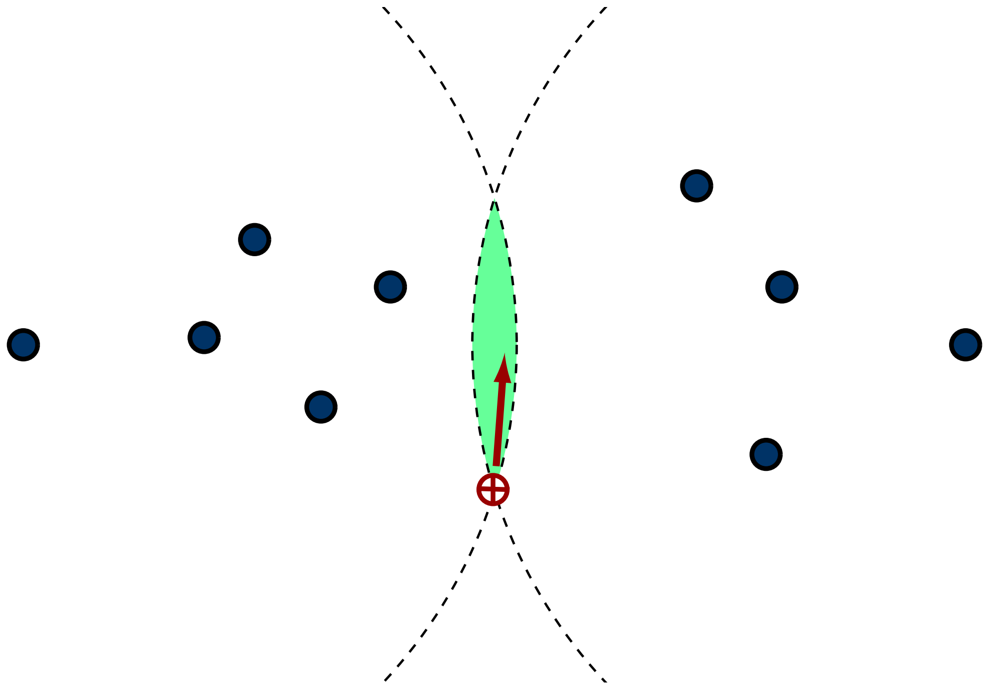
Any such function can be evaluated by randomized dual-simplex algorithms
[Matousek, Sharir, and Welzl, 1992; Amenta, 1994; Gärtner, 1995]

Typically: $O(n)$ feasibility tests (evaluation of an input function at a point)
$o(n)$ basis change operations (solve QCP for constant size subset)

## Alternative QCP solution technique: hill climbing

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

## 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 ***v*** s.t. ***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 ***w*** for all gradients
(lower dimensional minimum enclosing disk of few points)
If not found, algorithm has converged to solution

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

## Implementation

Implemented multi-gradient numerical solution in Python
for low-dimensional QCP (two- and three- variable recurrences)

Two versions:

### Floating point

Finds upper bounds for recurrences such as the one on the first slide
interactively (1-2 seconds on a modern laptop)

### Exact real arithmetic

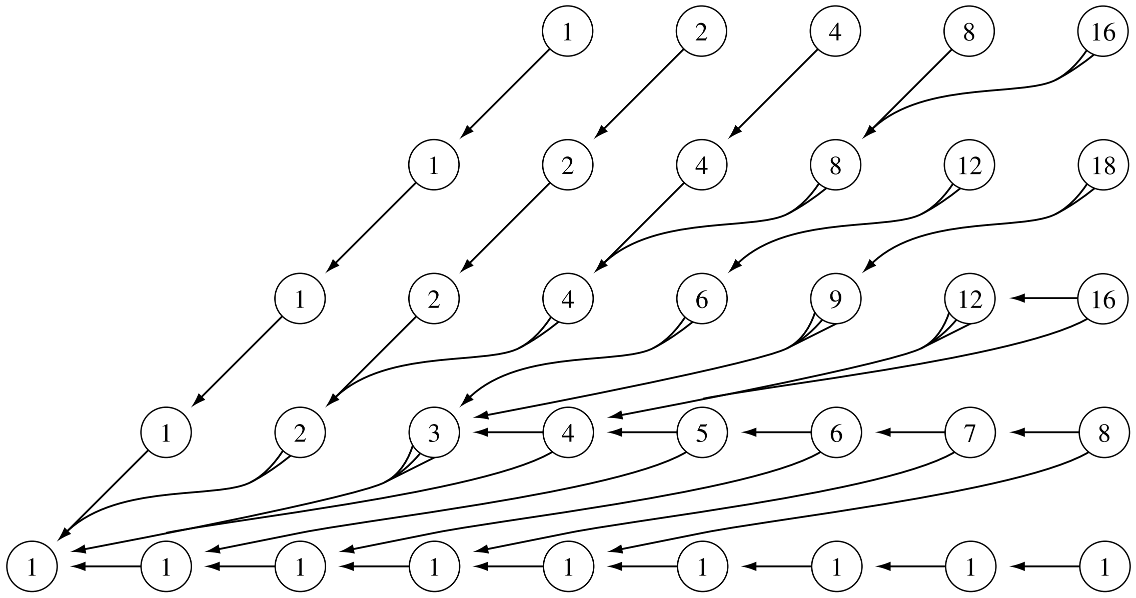
(Keith Briggs' XR package)

Provides results with guaranteed precision
Several hours per solution
Significant speedups likely (e.g. by using a compiled language)

## 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

Technique for upper-bounding backtracking algorithm recurrences, suitable for computer implementation

Bounds found by this technique are within a polynomial factor of tight (a logarithmic factor compared to overall exponential solution)

Floating point implementation is usable for exploratory analysis

Exact real arithmetic implementation is usable for published results but further efficiency improvement desirable

## Open problems

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?

Can automated analysis of backtracking algorithms
be extended to automatic design?