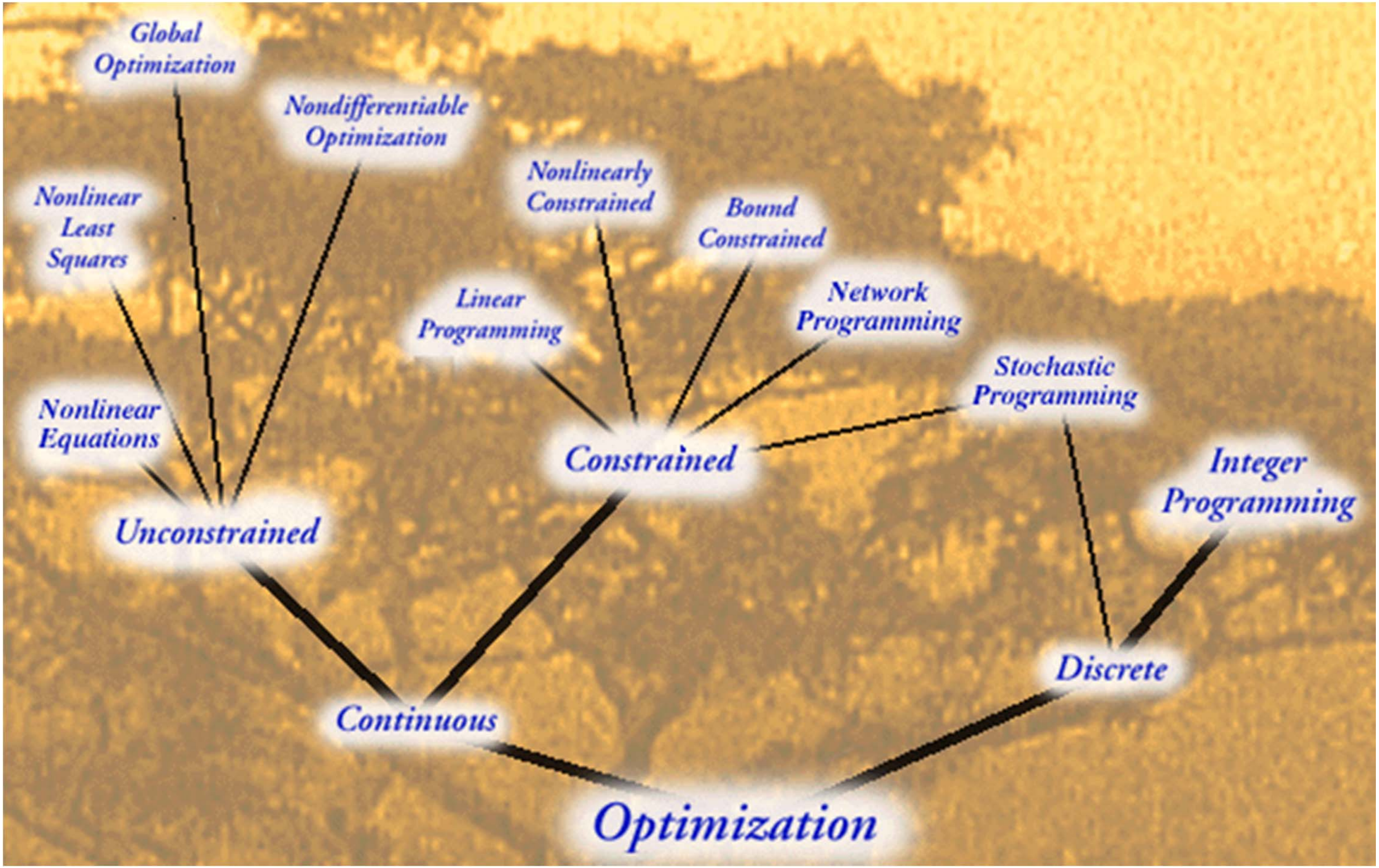
Lecture 4

3B1B Optimization Michaelmas 2015 A. Zisserman

- Convexity
- Robust cost functions
- Optimizing non-convex functions
  - grid search
  - branch and bound
  - multiple coverings
  - simulated annealing

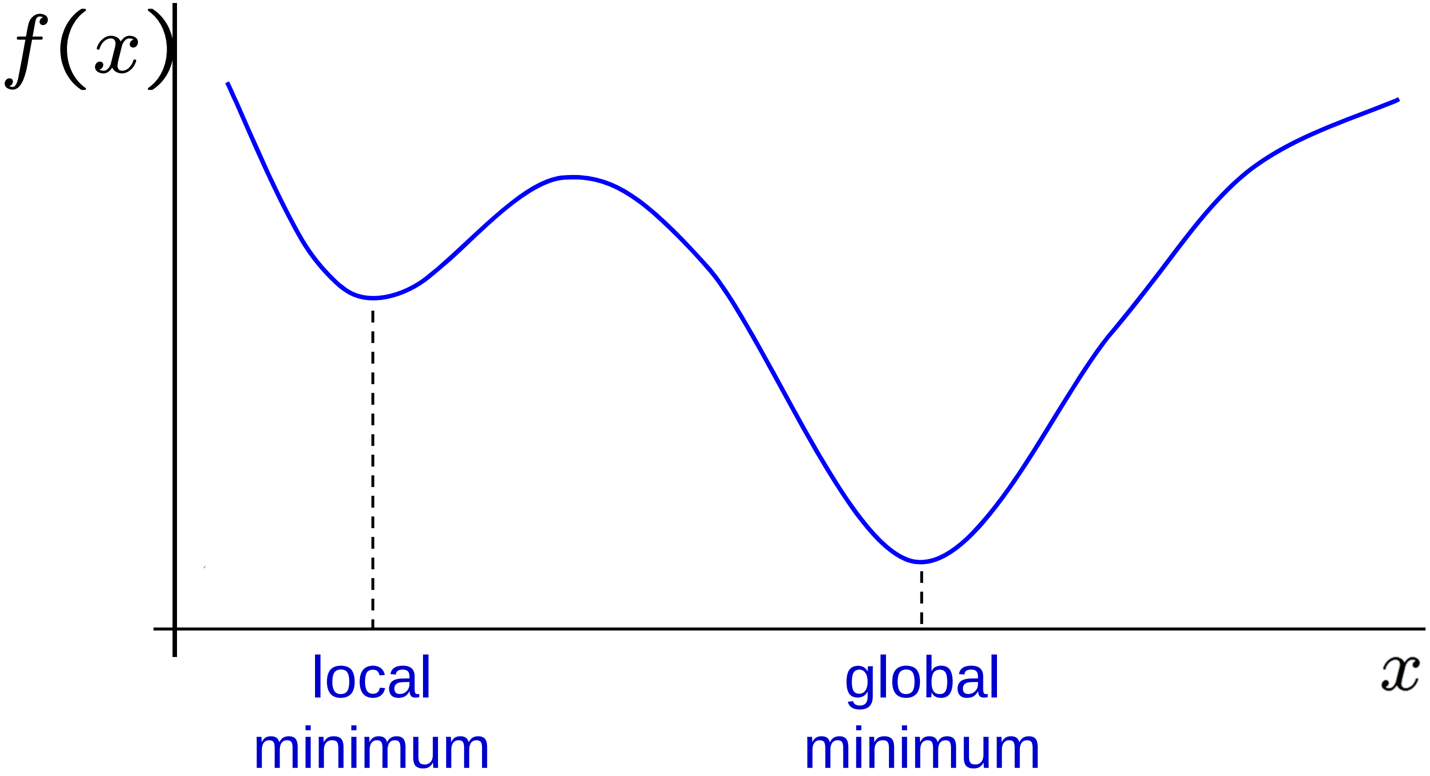
The Optimization Tree



Unconstrained optimization

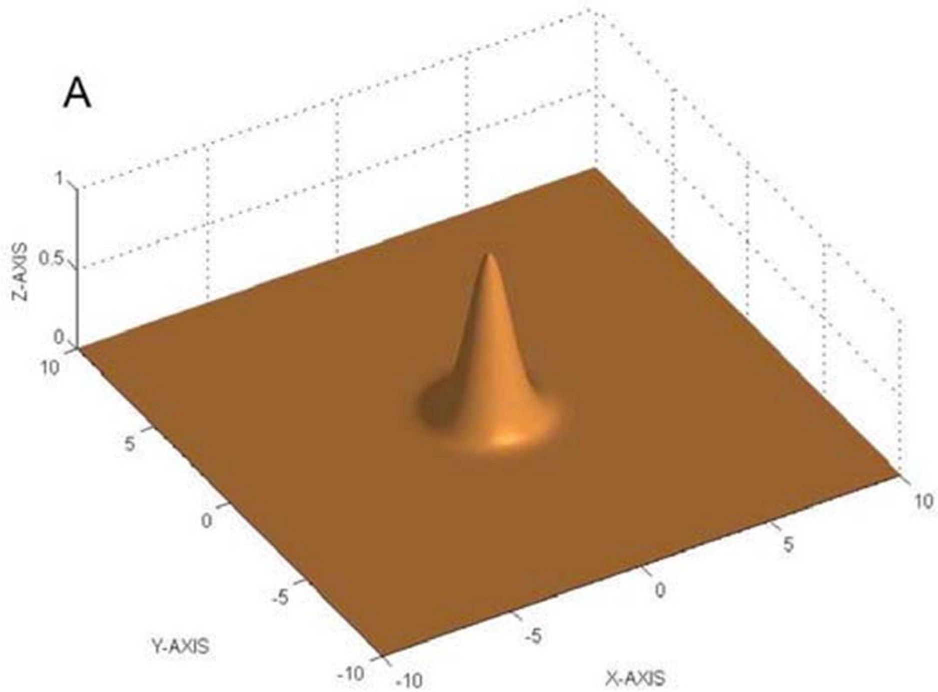
function of one variable

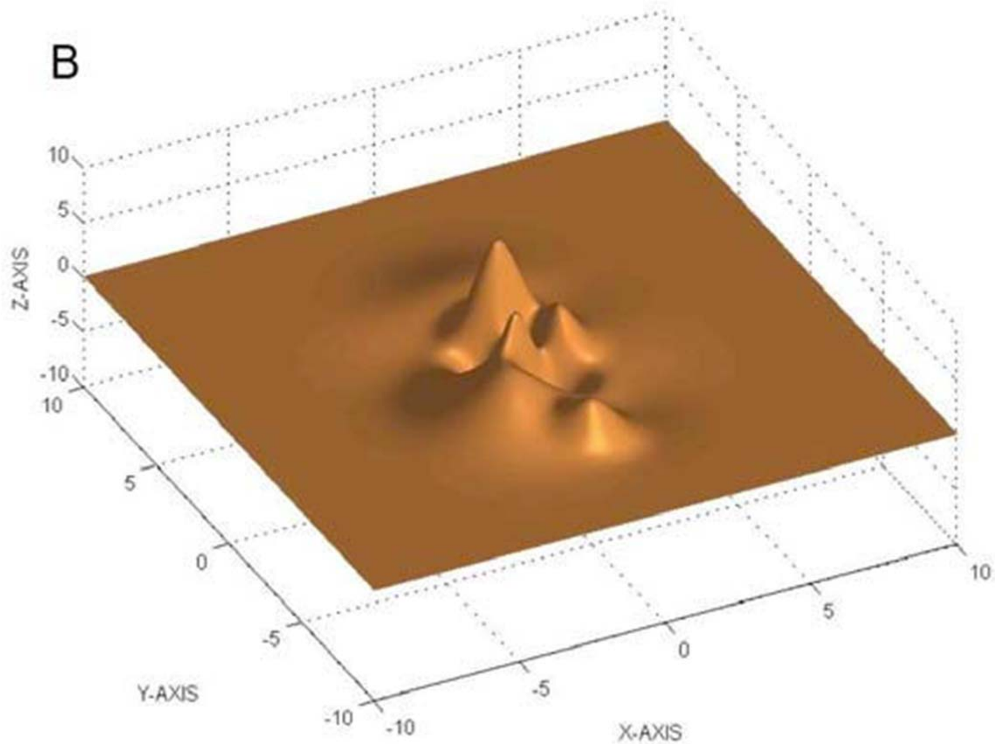
$$\min_{x} f(x)$$



- down-hill search (gradient descent) algorithms can find local minima
- which of the minima is found depends on the starting point
- such minima often occur in real applications

Cost functions in 2D





Global maximum

Multiple optima

How can you tell if an optimization has a single optimum?

The answer is: see if the optimization problem is convex.

If it is, then a local optimum is the global optimum.

First, we need to introduce

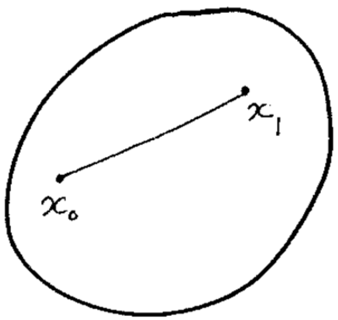
- Convex Sets, and
- Convex Functions

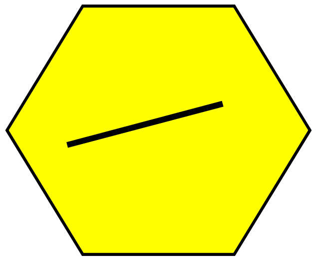
Note – sketch introduction only

Convex set

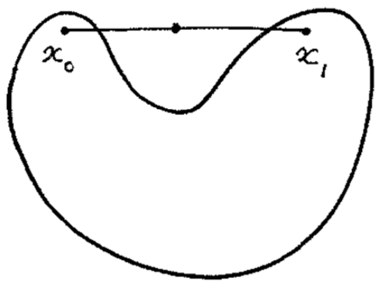
A set $D \subset R^n$ is convex if the line joining points $\mathbf{x}_0$ and $\mathbf{x}_1$ lies inside D.

convex

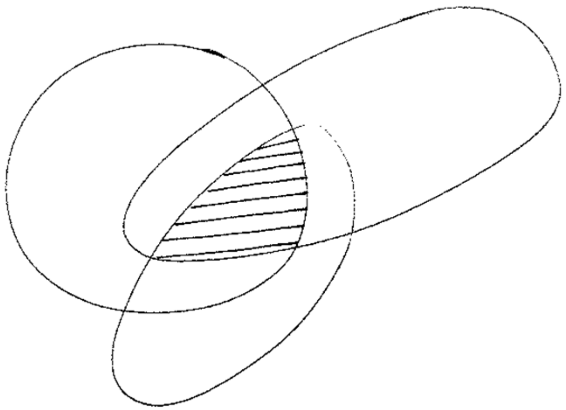




Not convex



Intersection of convex sets in convex.



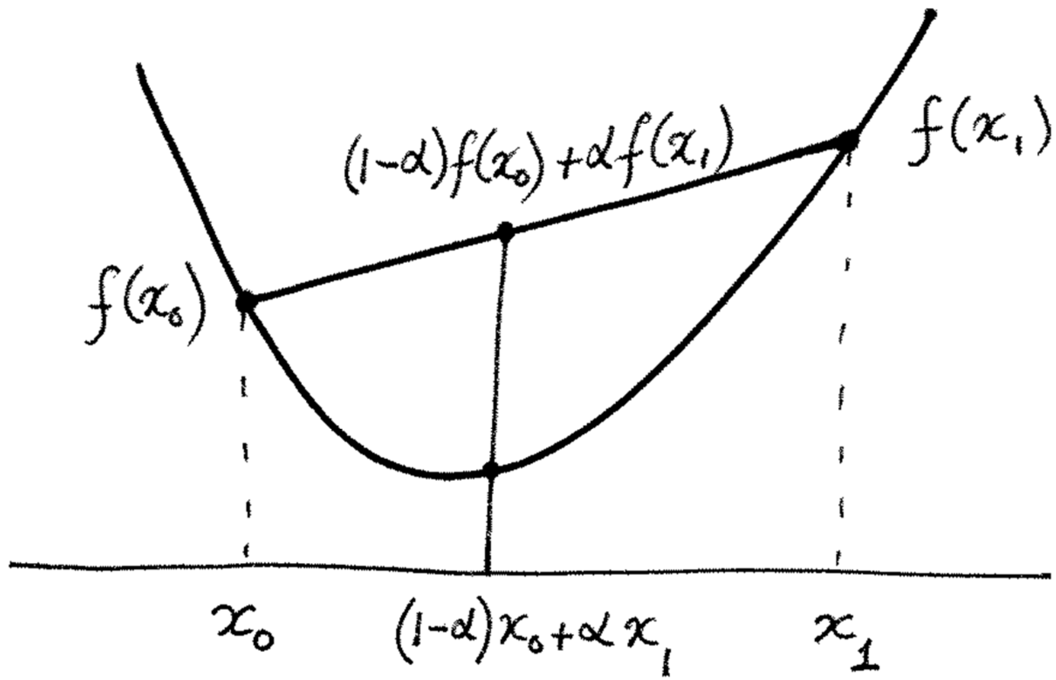
Convex functions

D – a domain in $\mathbb{R}^n$.

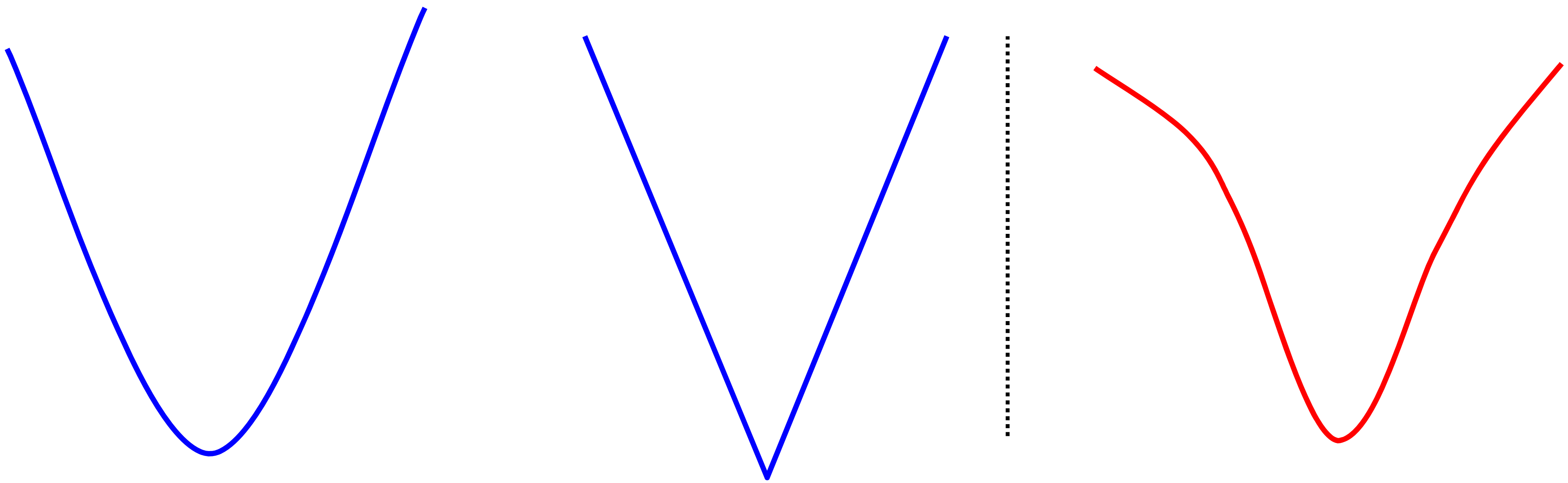
A convex function $f : D \to \mathbb{R}$ is one that satisfies, for any $\mathbf{x}_0$ and $\mathbf{x}_1$ in D:

$$f((1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1) \le (1-\alpha)f(\mathbf{x}_0) + \alpha f(\mathbf{x}_1) \quad .$$

Line joining $(\mathbf{x}_0, f(\mathbf{x}_0))$ and $(\mathbf{x}_1, f(\mathbf{x}_1))$ lies above the function graph.



Convex function examples



convex

Not convex

A non-negative sum of convex functions is convex

Convex Optimization Problem

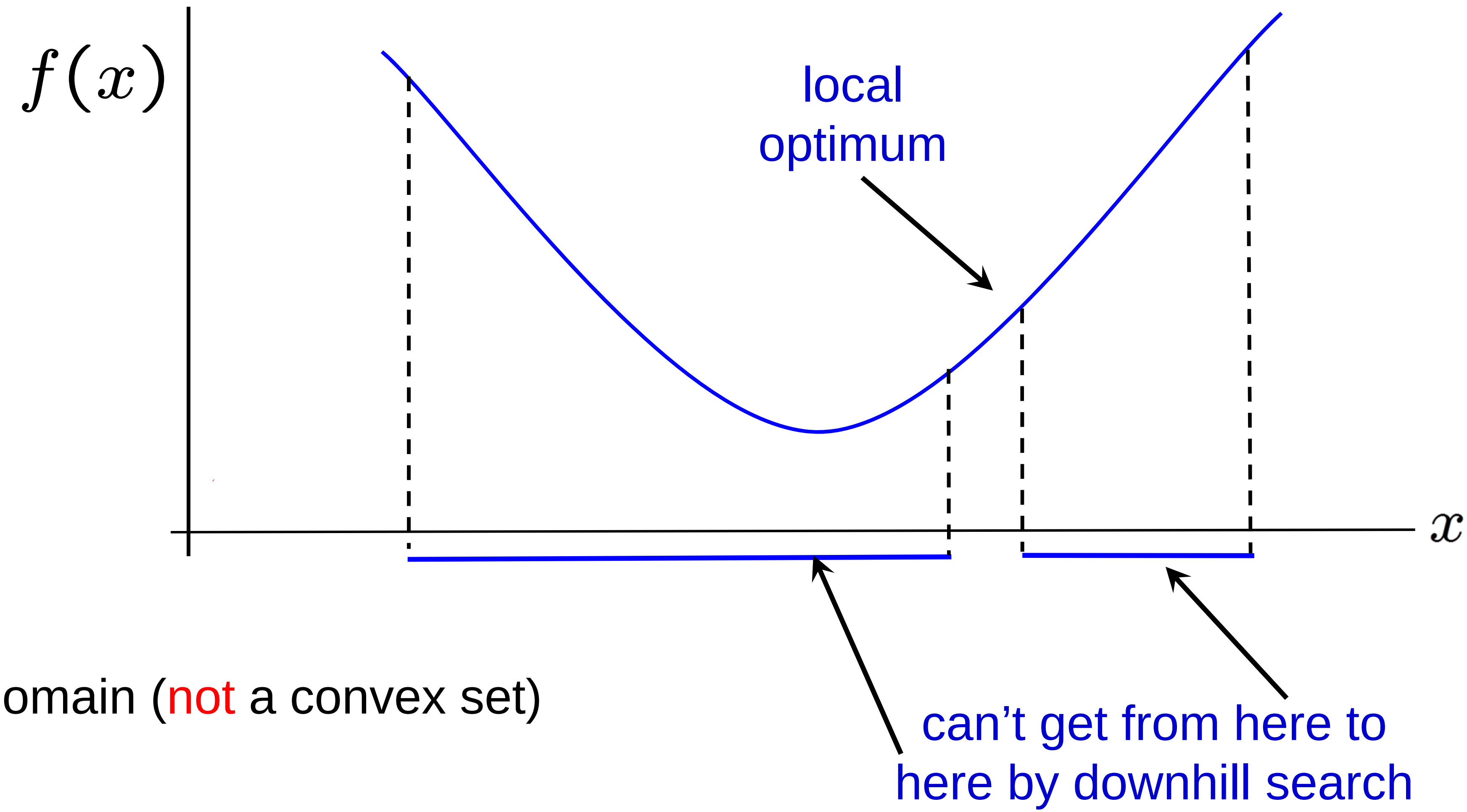
Minimize:

- a convex function
- over a convex set

Then locally optimal points are globally optimal

Also, such problems can be solved both in theory and practice

Why do we need the domain to be convex?



domain (not a convex set)

Examples of convex optimization problems

1. Linear programming

2. Least squares

$$f(\mathbf{x}) = (\mathtt{A}\mathbf{x} - \mathbf{b})^2, \text{ for any } \mathtt{A}$$

3. Quadratic functions

$$f(\mathbf{x}) = \mathbf{x}^\top \mathtt{P}\mathbf{x} + \mathbf{q}^\top \mathbf{x} + r, \text{ provided that } \mathtt{P} \text{ is positive definite}$$

Many more useful examples, see Boyd & Vandenberghe

First-order condition

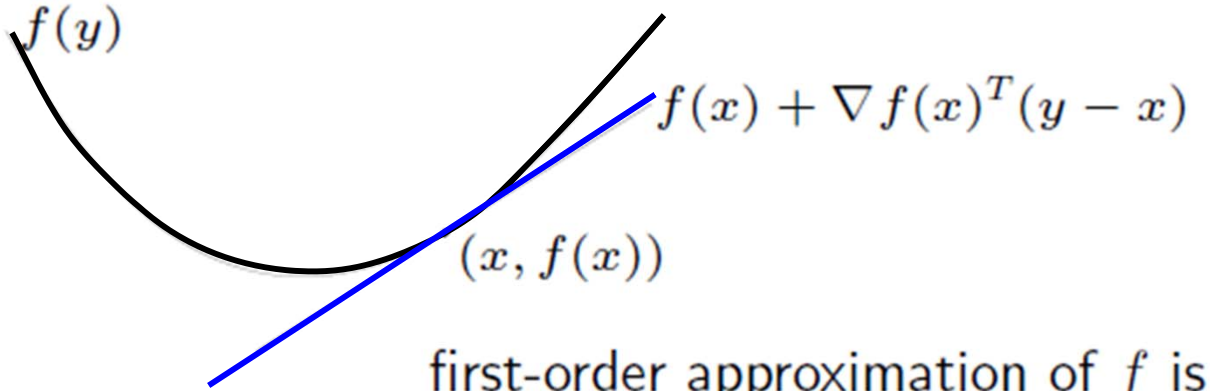
f is **differentiable** if $\mathbf{dom}\, f$ is open and the gradient

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

exists at each $x \in \mathbf{dom}\, f$

1st-order condition: differentiable f with convex domain is convex iff

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for all } x, y \in \mathbf{dom}\, f$$



first-order approximation of f is global underestimator

Second order condition

The Hessian of a function $f(x_1, x_2, \ldots, \mathrm{x}_n)$ is the matrix of partial derivatives

$$\mathtt{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

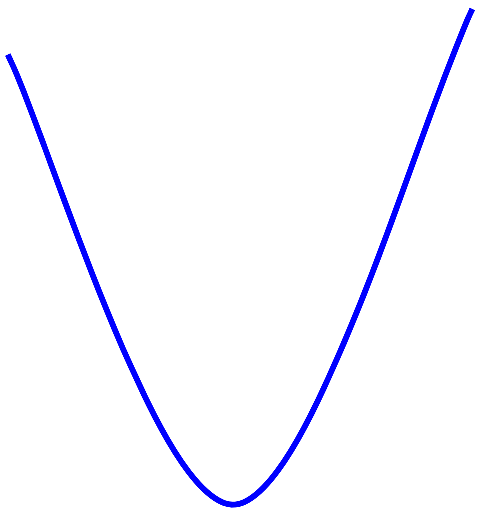
Diagonalize the Hessian by an orthogonal change of coordinates.
Diagonals are the eigenvalues.

If the eigenvalues are all positive, then the Hessian is positive definite, and f is convex.

Strictly convex

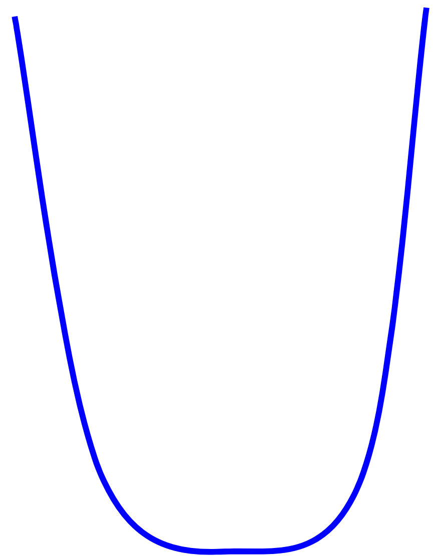
A function $f(x)$ is strictly convex if

$$f((1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1) < (1-\alpha)f(\mathbf{x}_0) + \alpha f(\mathbf{x}_1) \quad .$$



strictly convex

one global optimum



Not strictly convex

multiple local optima
(but all are global)

Robust Cost Functions

- In formulating an optimization problem there is often some room for design and choice

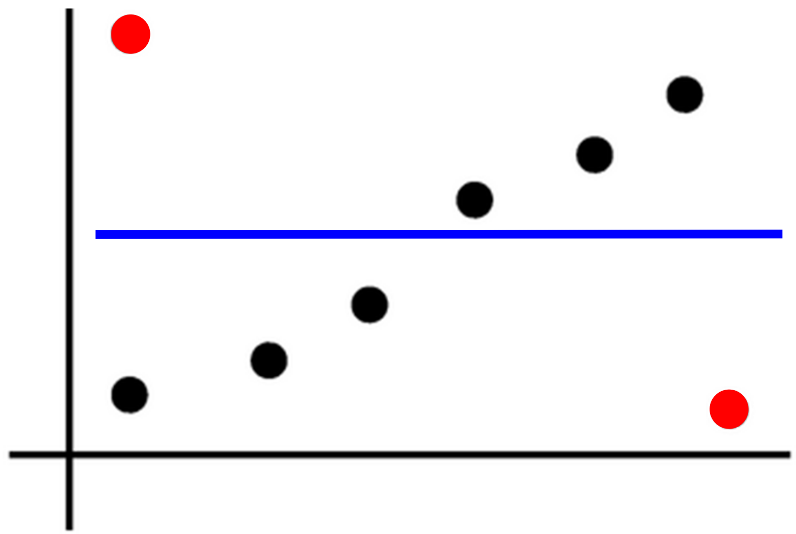
- The cost function can be chosen to be:
  - convex
  - robust to noise (outliers) in the data/measurements

Motivation

Fitting a 2D line to a set of 2D points

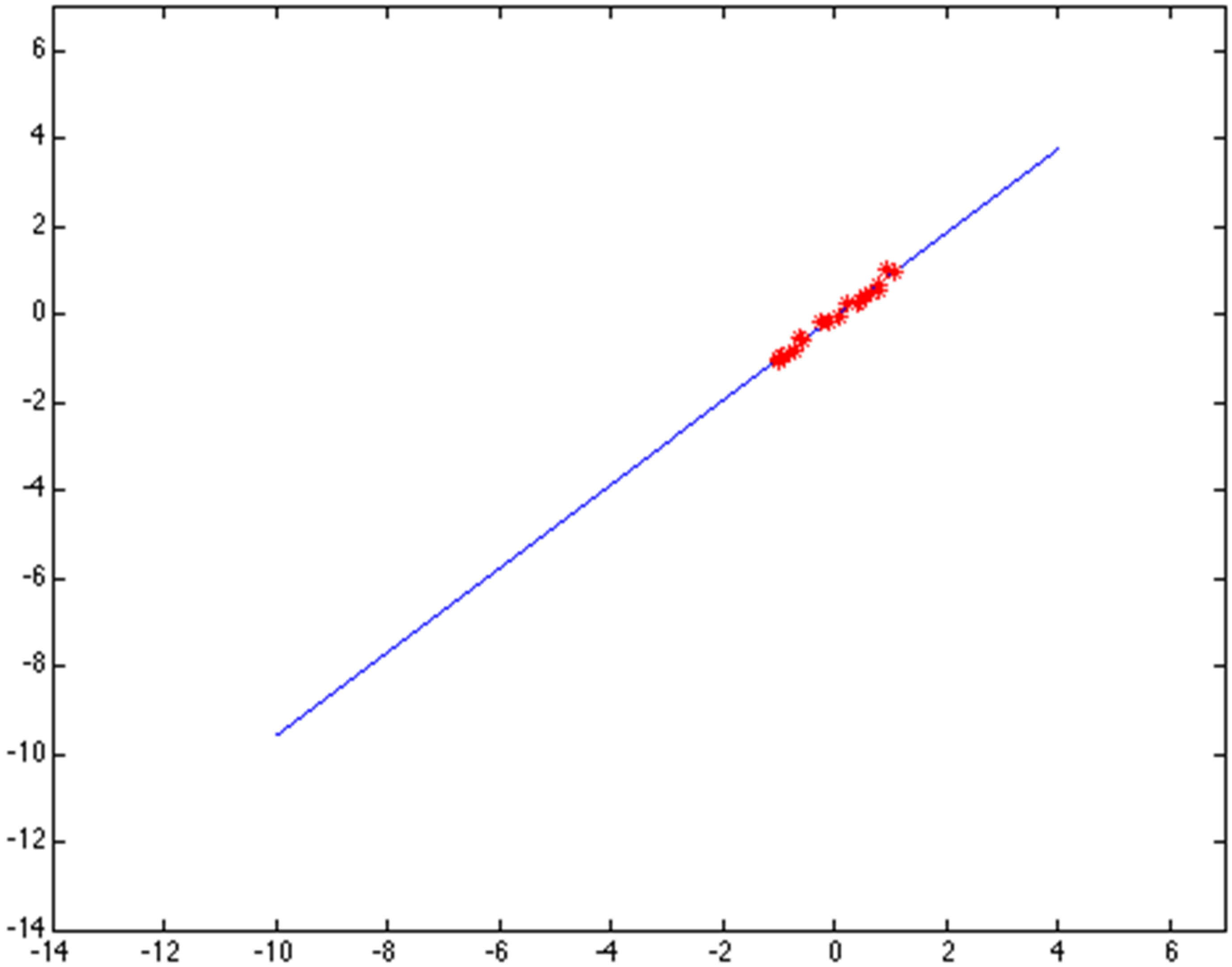
- Suppose you fit a straight line to data containing **outliers** – points that are not properly modelled by the assumed measurement noise distribution

- The usual method of least squares estimation is hopelessly corrupted



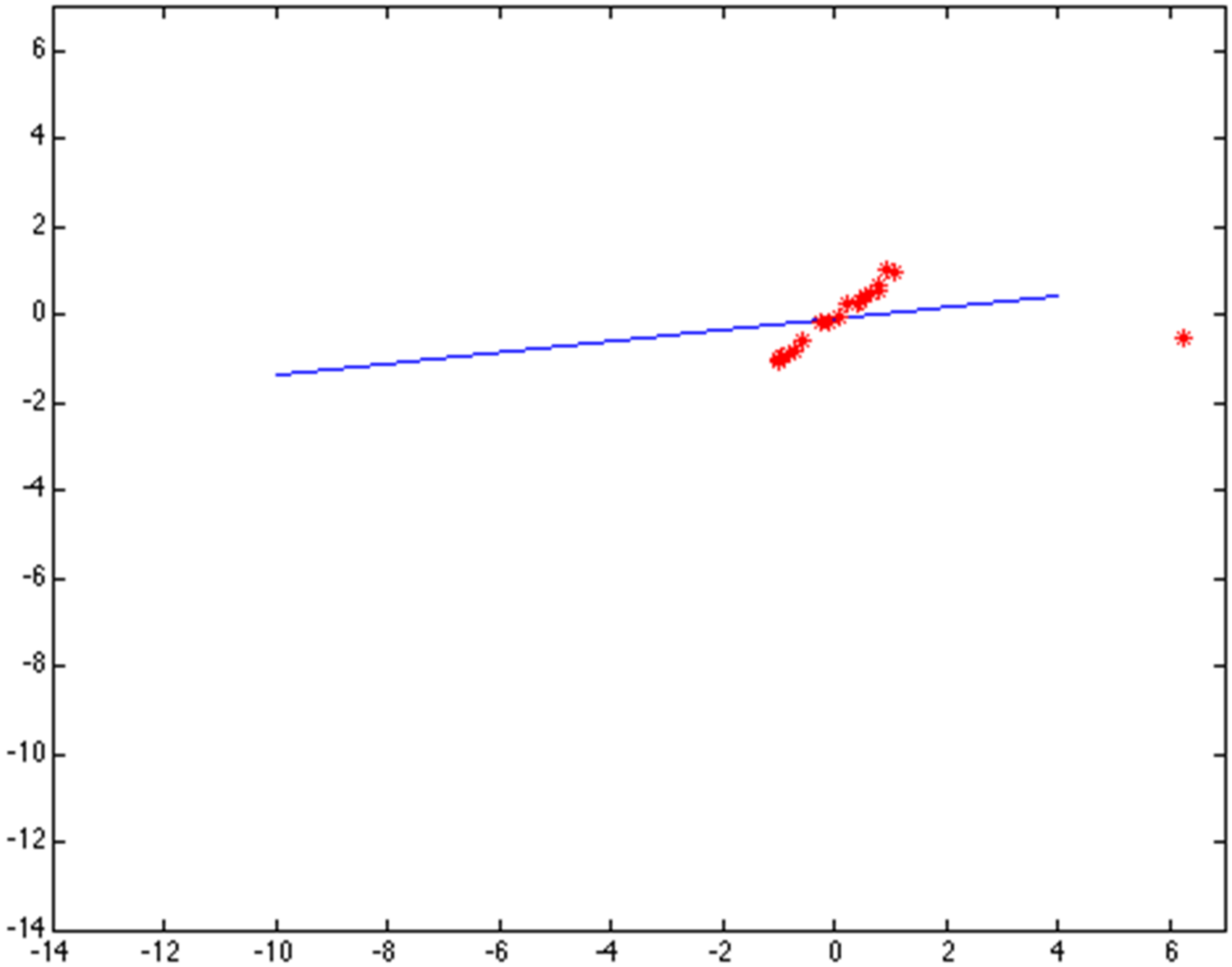
Least squares: Robustness to noise

Least squares fit to the red points:



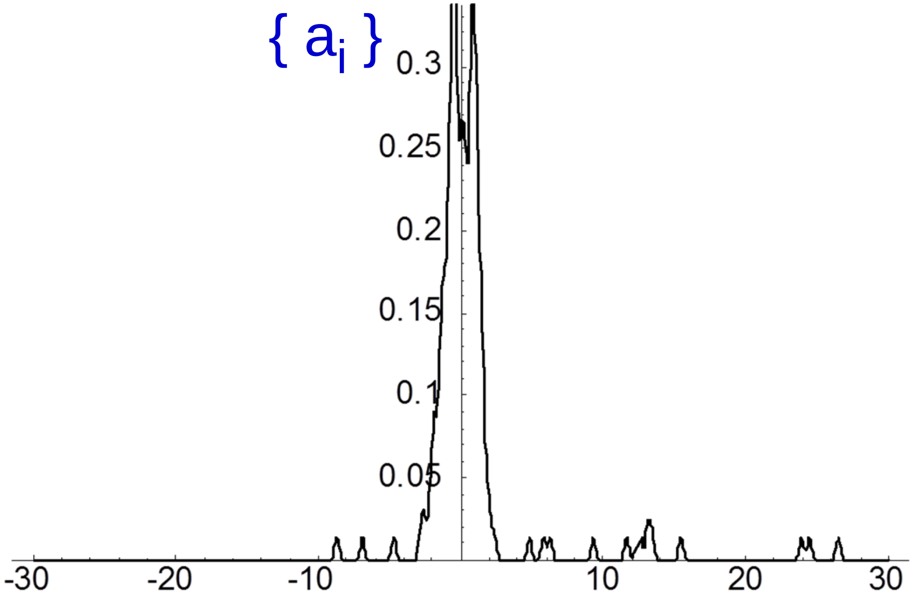
Least squares: Robustness to noise

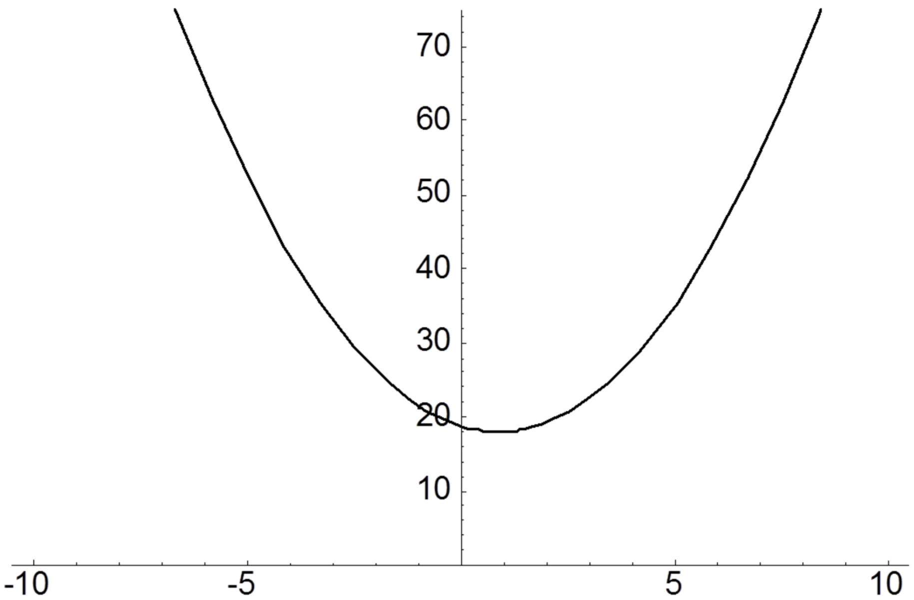
Least squares fit with an outlier:



Problem: squared error heavily penalizes outliers

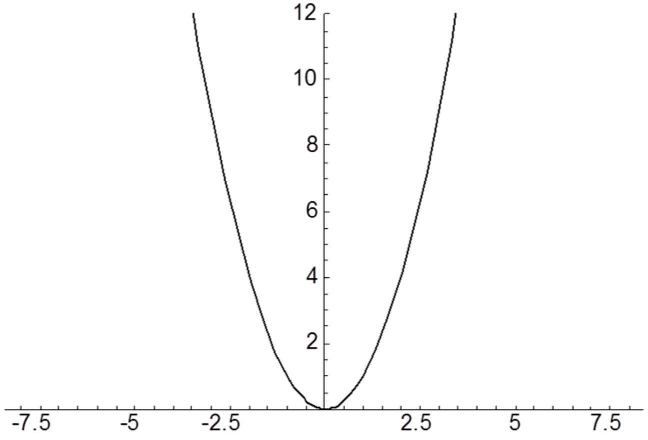
Consider minimizing the cost function $f(x) = \sum_i (x - a_i)^2$



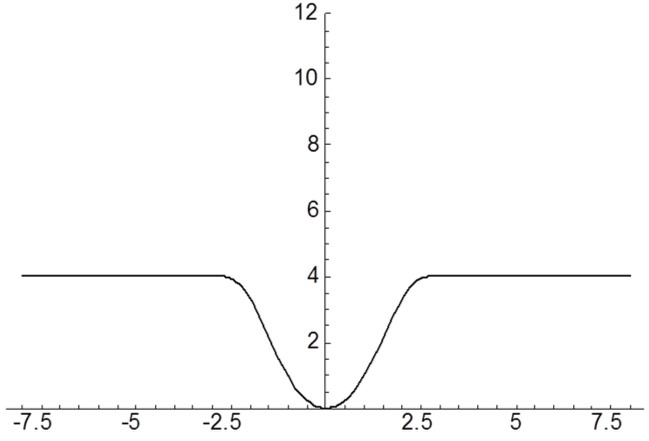


- the data $\{ a_i \}$ may be thought of as repeated measurements of a fixed value (at 0), subject to Gaussian noise and some outliers
- it has 10% of outliers biased towards the right of the true value
- the minimum of f(x) does not correspond to the true value

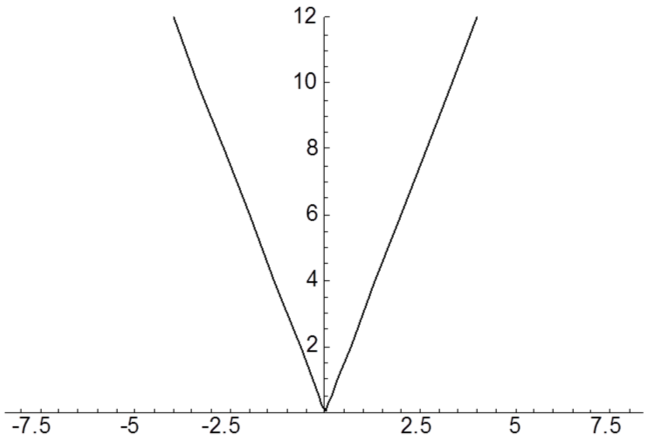
Examine the behaviour of various cost functions $f(x) = \sum_i C(|x - a_i|)$



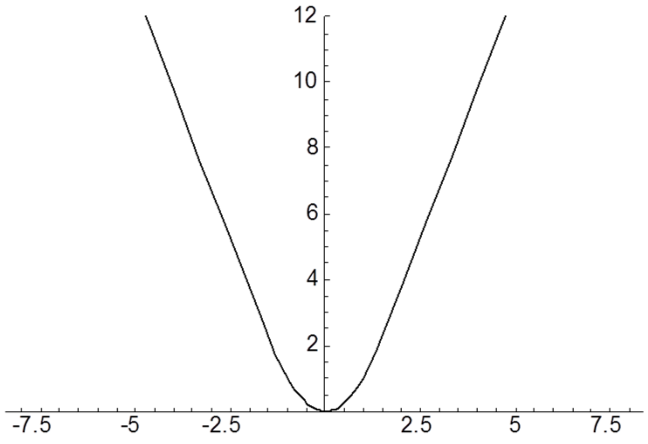
quadratic



truncated quadratic



L_1



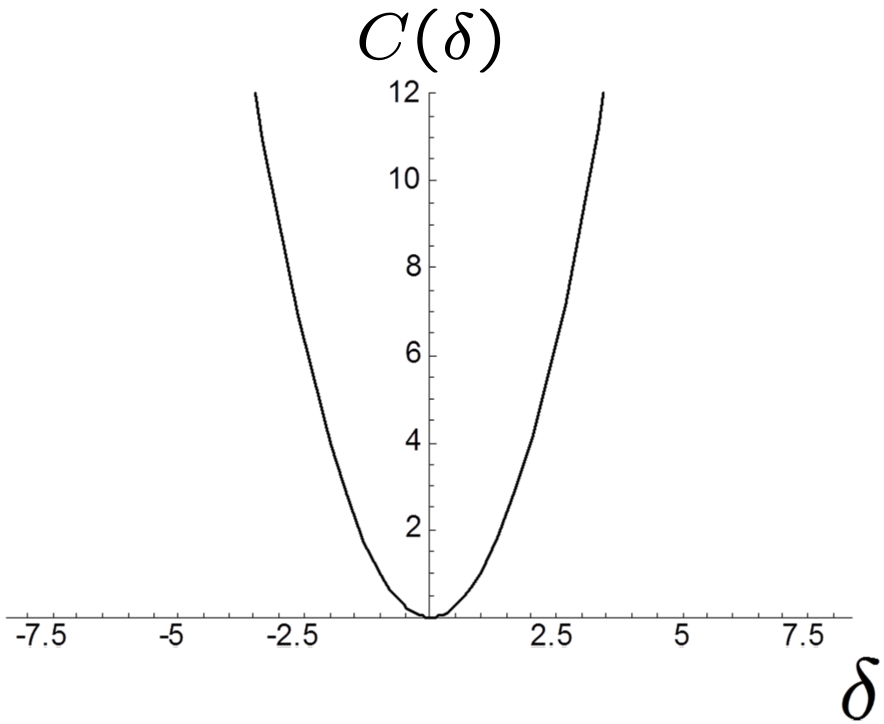
huber

Quadratic cost function

- squared error – the usual default cost function

- arises in Maximum Likelihood Estimation for Gaussian noise

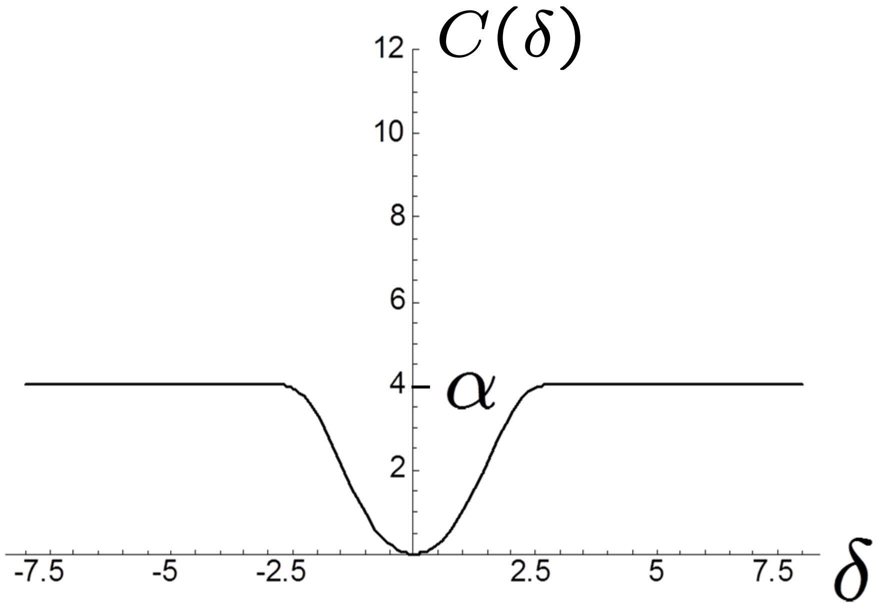
- convex



$$C(\delta) = \delta^2$$

Truncated Quadratic cost function

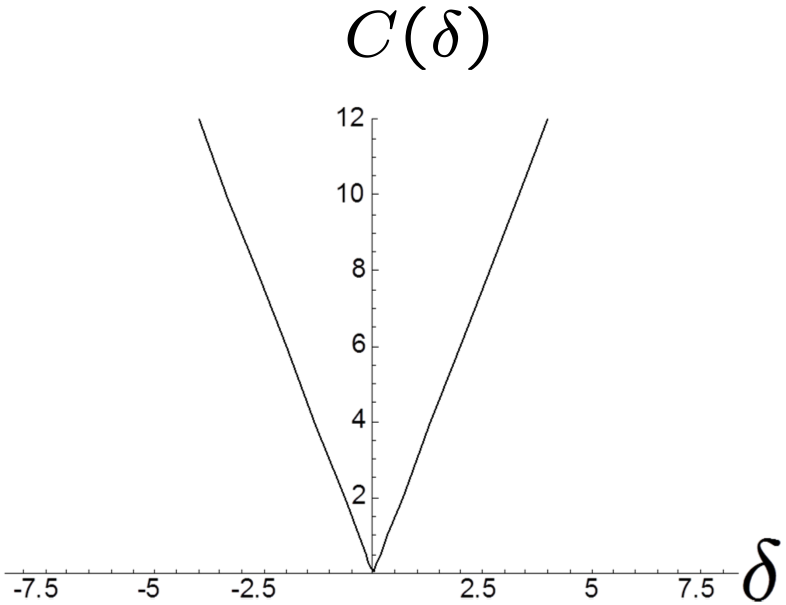
- for inliers behaves as a quadratic
- truncated so that outliers only incur a fixed cost
- non-convex



$$
\begin{aligned}
C(\delta) &= \mathsf{min}(\delta^2, \alpha) \\
&= \begin{cases} \delta^2 & \text{if } |\delta| < \sqrt{\alpha} \\ \alpha & \text{otherwise.} \end{cases}
\end{aligned}
$$

L_1 cost function

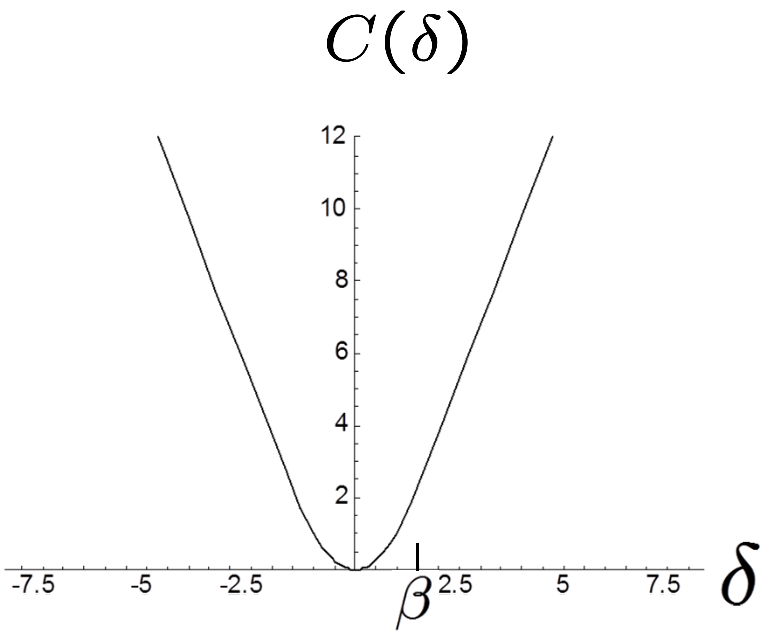
- absolute error
- called 'total variation'
- convex
- non-differentiable at origin
- finds the **median** of { a_i }



$$C(\delta) = |\delta|$$

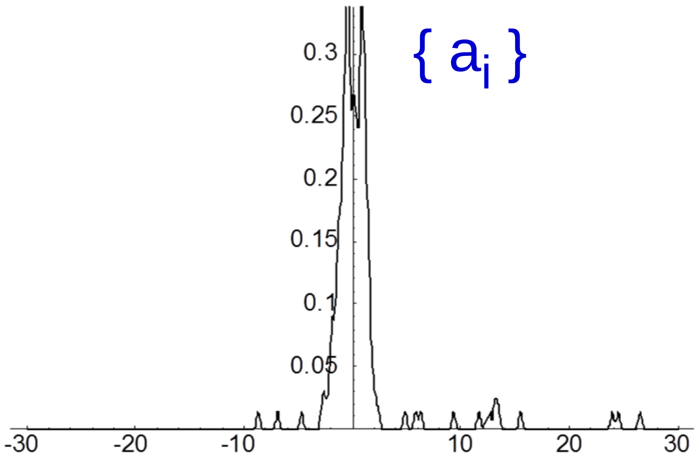
Huber cost function

- hybrid between quadratic and L_1
- continuous first derivative
- for small values is quadratic
- for larger values becomes linear
- thus has the outlier stability of L_1
- convex



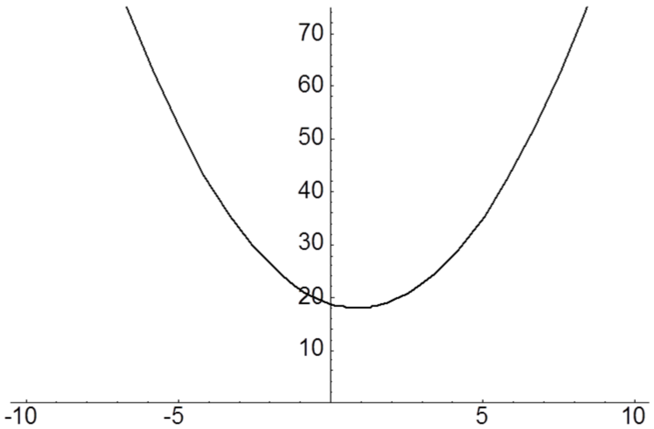
$$C(\delta) = \begin{cases} \delta^2 & \text{if } |\delta| < \beta \\ 2\beta|\delta| - \beta^2 & \text{otherwise.} \end{cases}$$

Example 1: measurements with outliers

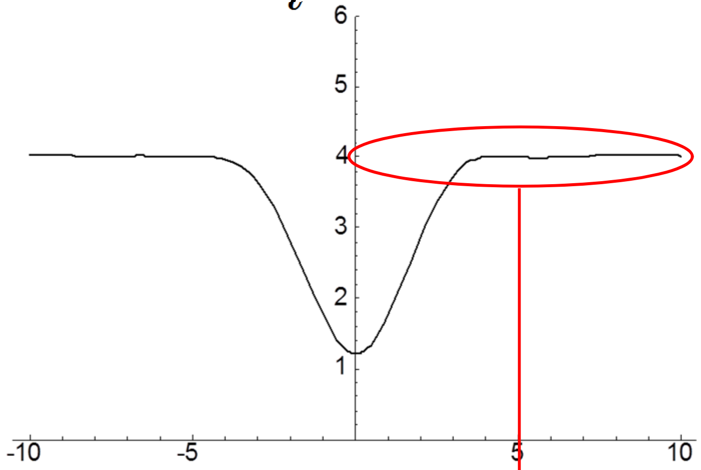


$\{ a_i \}$

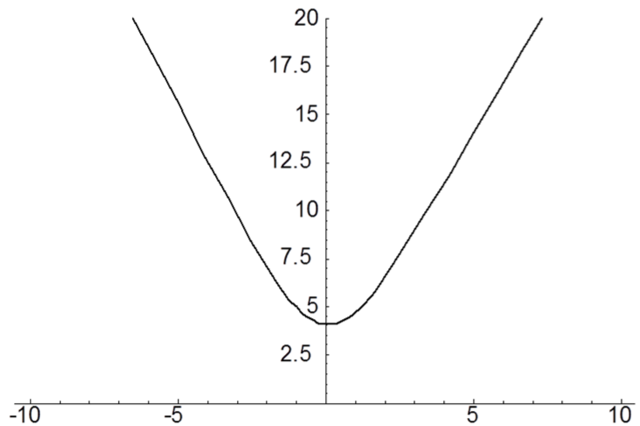
$$f(x) = \sum_i C(|x - a_i|)$$



quadratic



truncated quadratic

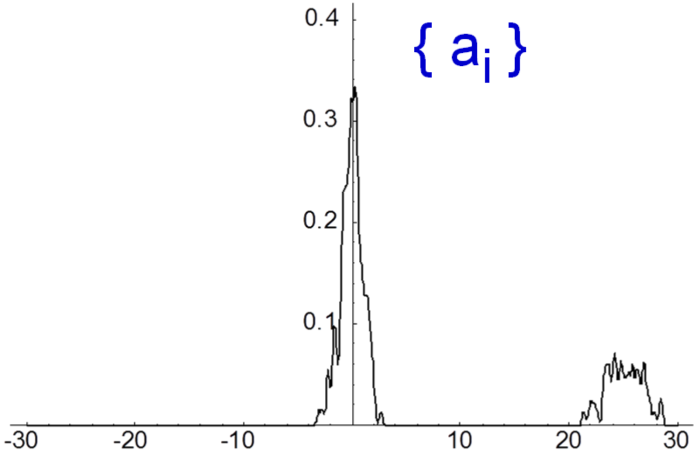


huber

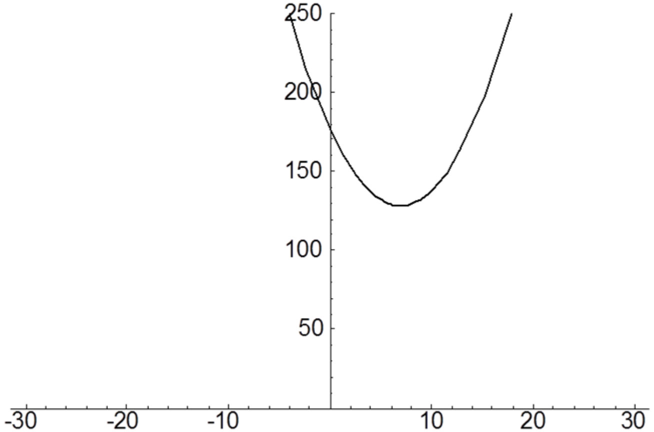
zoom

Example 2: bimodal measurements

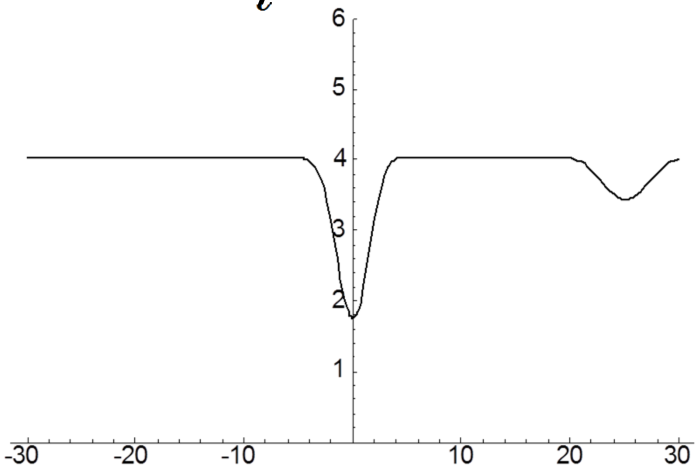
- 70% in principal mode
- 30% in outlier mode



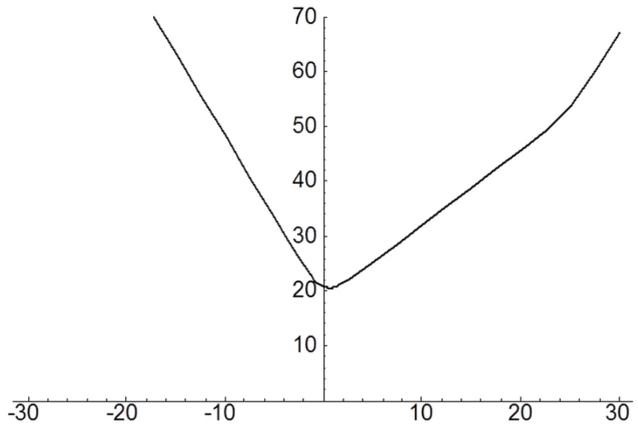
$$f(x) = \sum_i C(|x - a_i|)$$



quadratic



truncated quadratic



huber

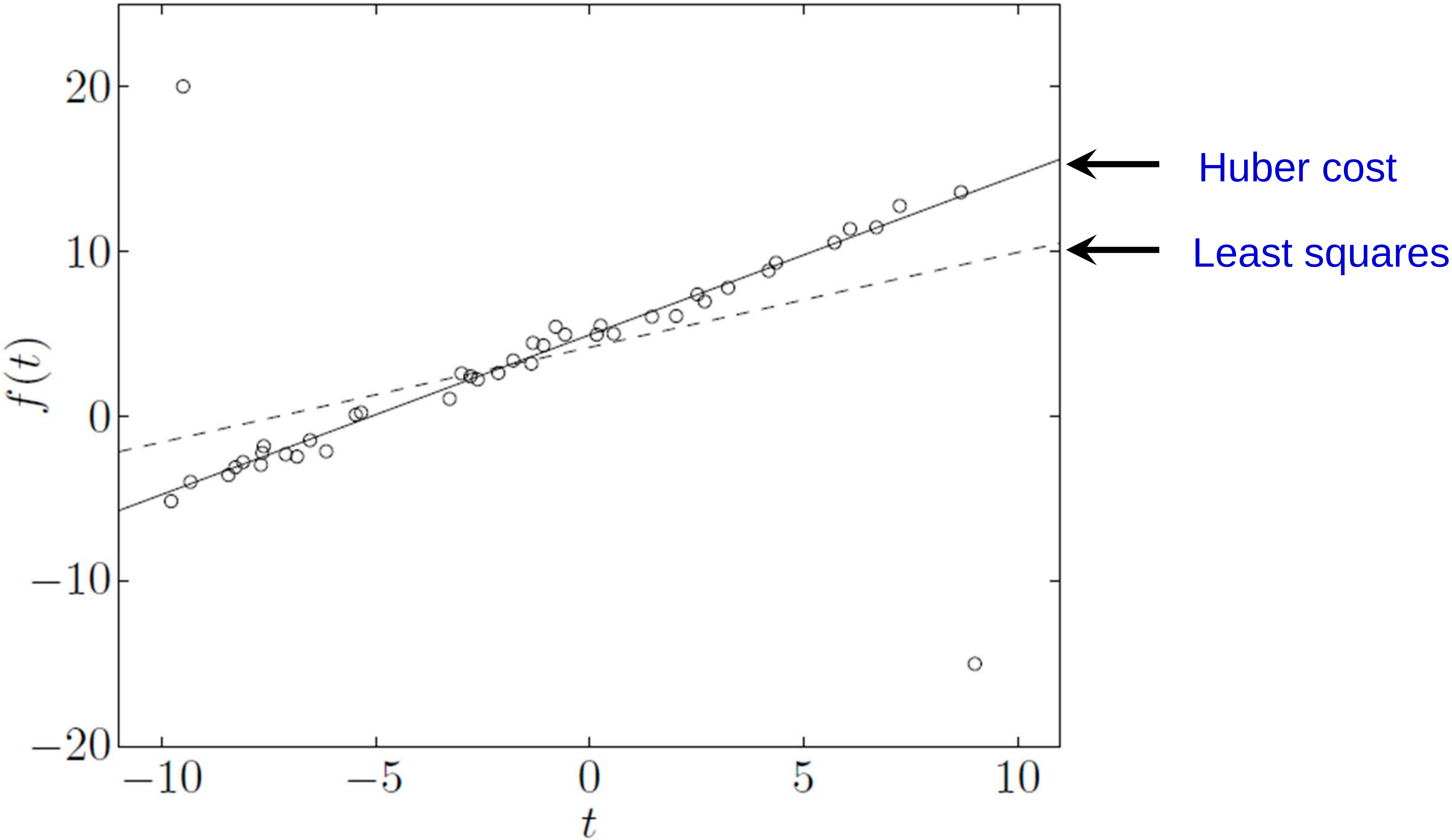
Summary

• Squared cost function very susceptible to outliers

• Truncated quadratic has a stable minimum, but is non-convex and also has other local minima. Also basin of attraction of global minimum limited

• Huber has stable minimum and is convex

Application 1: Robust line fitting



(See also RANSAC algorithm)

Boyd & Vandenberghe

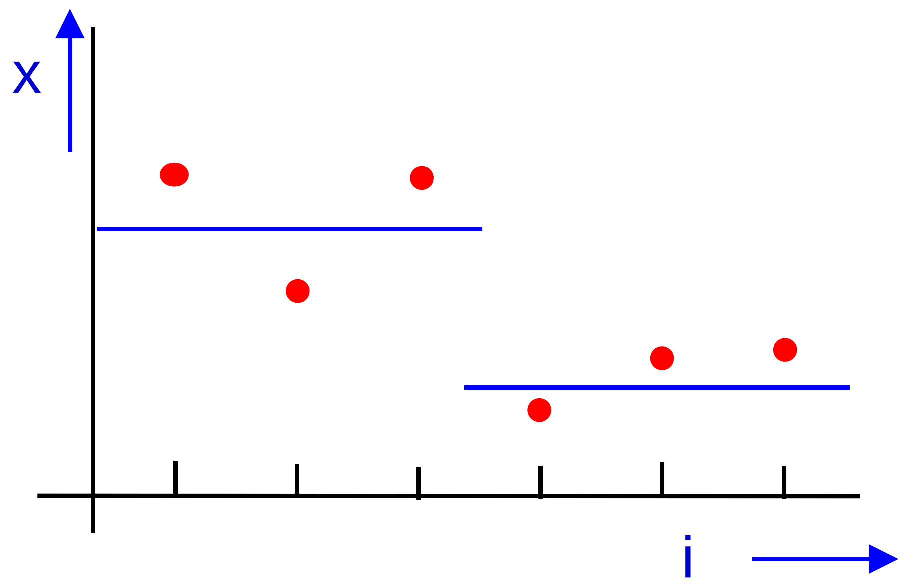
Application 2: Signal restoration

Measurements z_i are original signal x_i corrupted with additive noise

$$z_i = x_i + w_i$$

where

$$w_i \sim N(0, \sigma^2)$$



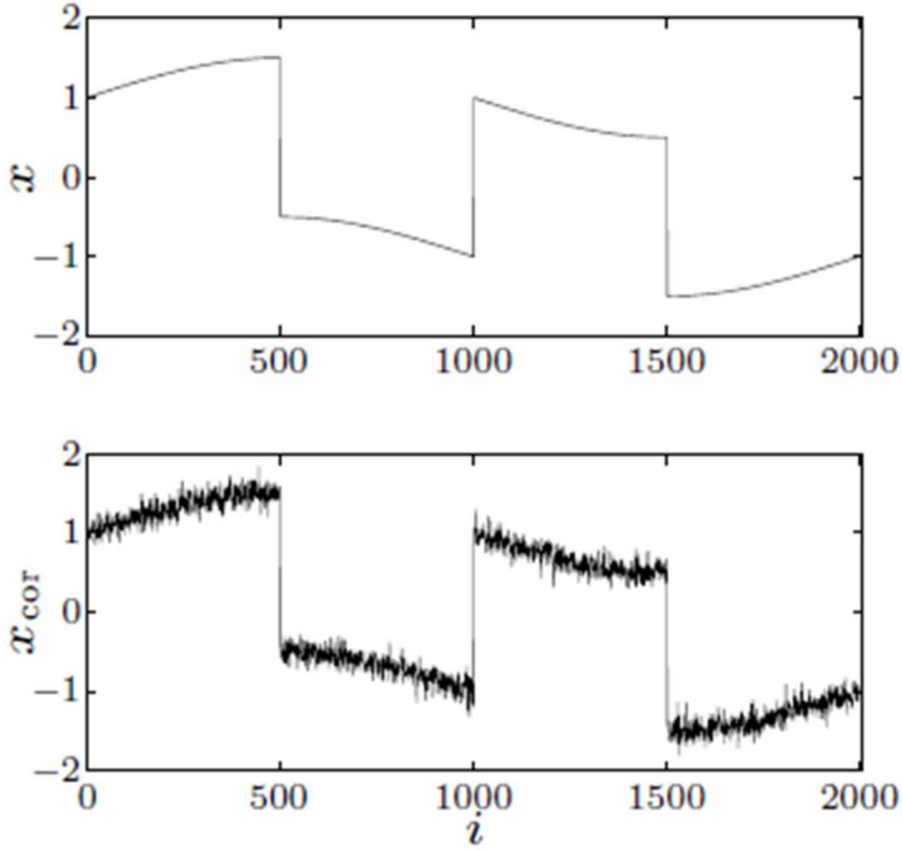
Compare:

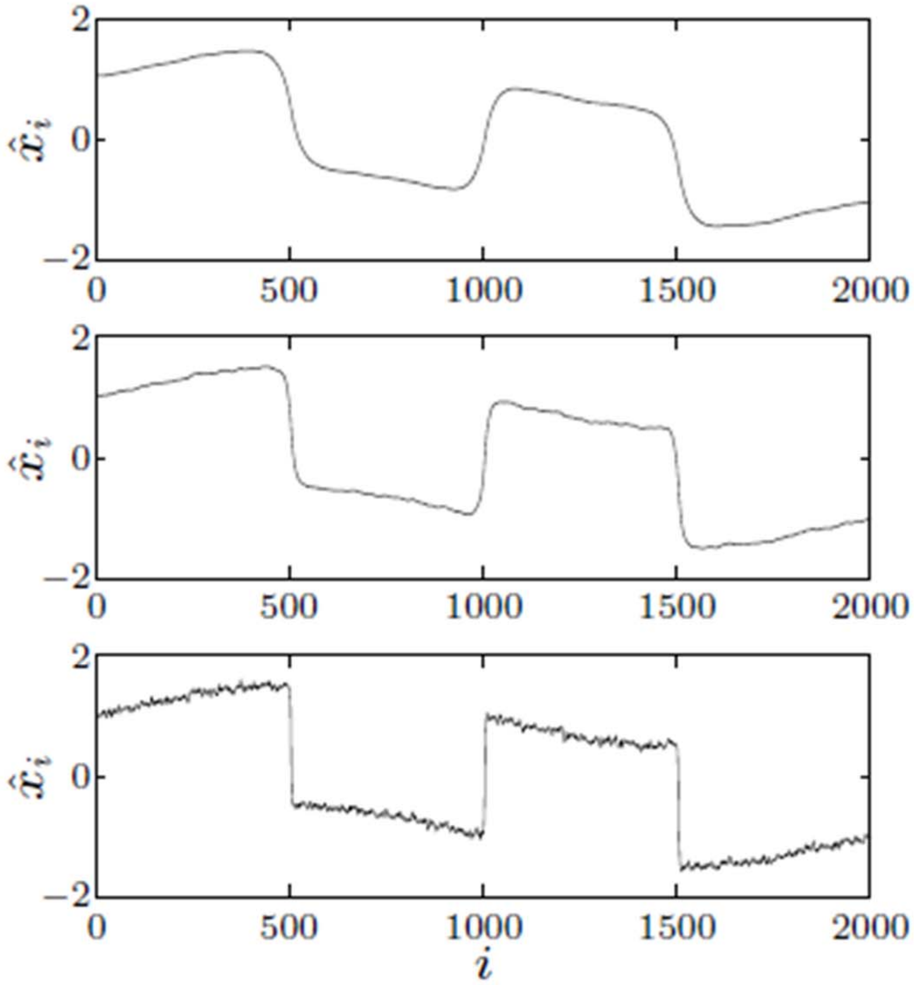
$$f(\mathbf{x}) = \sum_i (z_i - x_i)^2 + \lambda (x_i - x_{i-1})^2$$ Quadratic smoothing

$$f(\mathbf{x}) = \sum_i (z_i - x_i)^2 + \lambda |x_i - x_{i-1}|$$ Total variation

Quadratic smoothing

$$f(\mathbf{x}) = \sum_i (z_i - x_i)^2 + \lambda (x_i - x_{i-1})^2$$



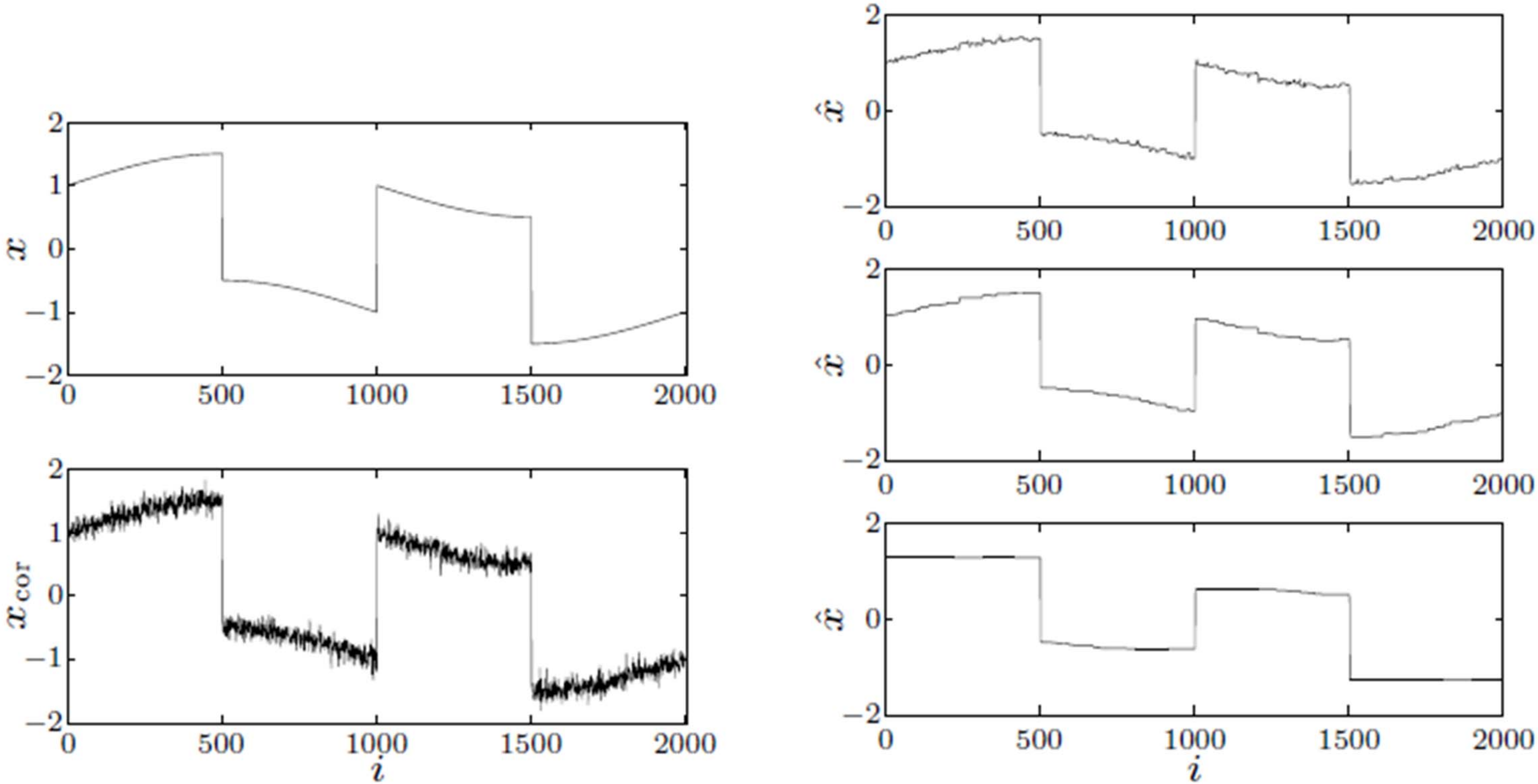


Quadratic smoothing smooths out noise **and** steps in the signal

Boyd & Vandenberghe

Total variation

$$f(\mathbf{x}) = \sum_i (z_i - x_i)^2 + \lambda |x_i - x_{i-1}|$$



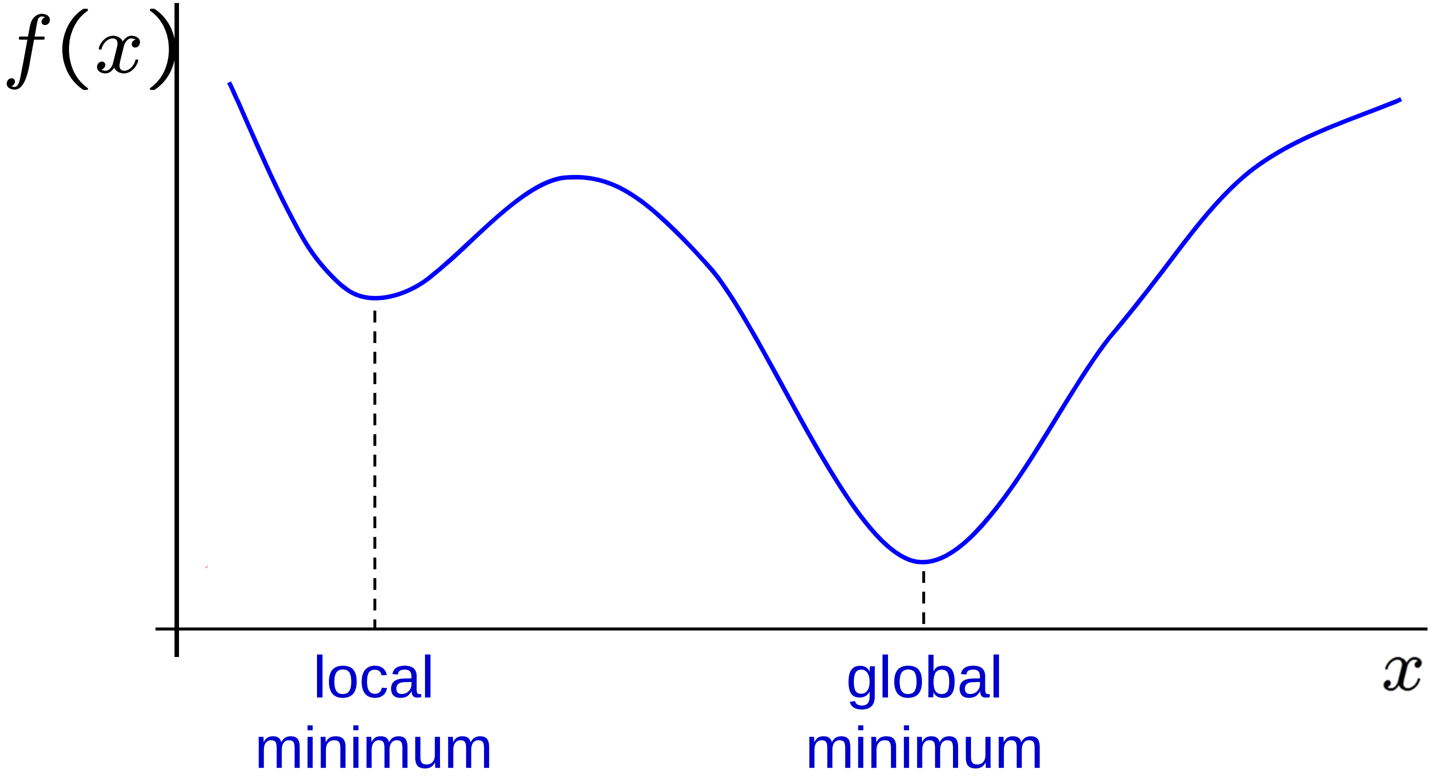
Total variation smoothing preserves steps in the signal

Boyd & Vandenberghe

Optimizing non-convex functions

function of one variable

$$\min_x f(x)$$

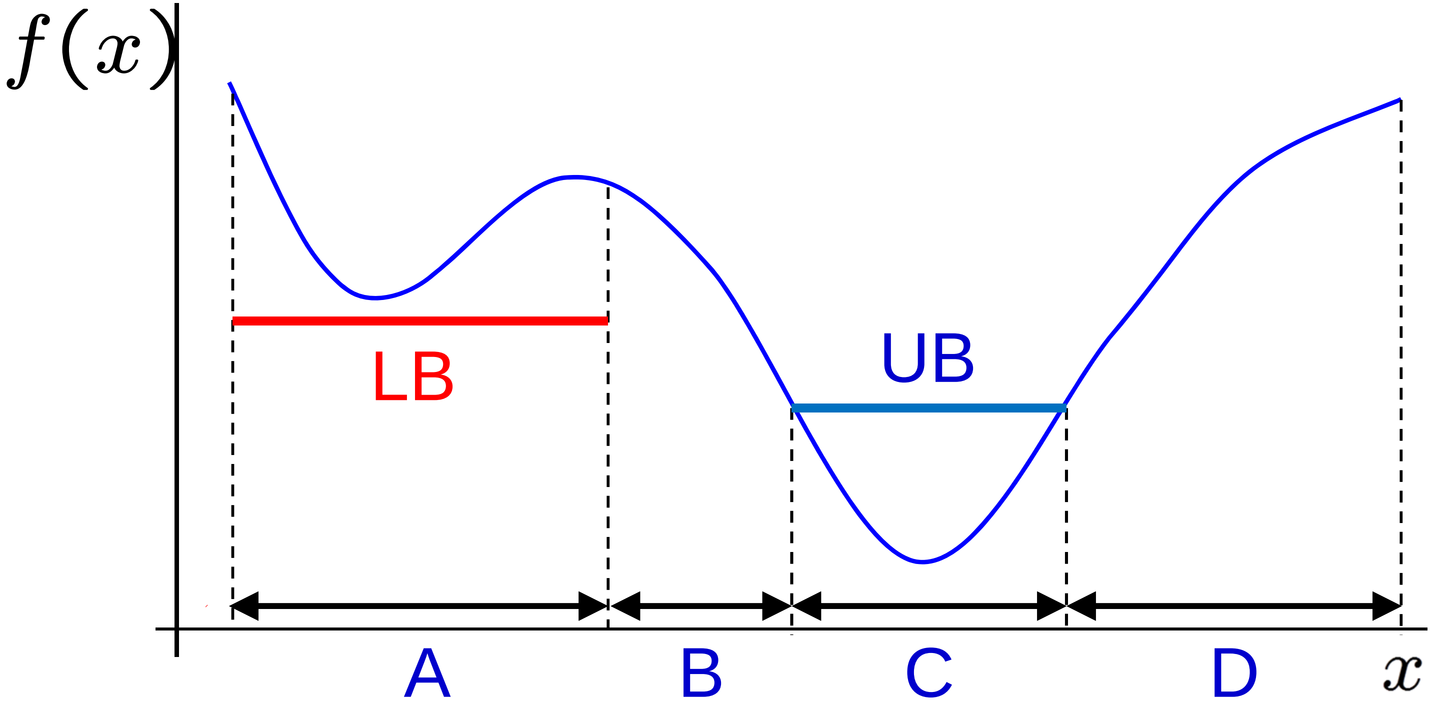


Sketch four methods:

1. grid search: uniform grid space covering
2. branch and bound
3. multiple coverings: Newton like methods within regions
4. simulated annealing: stochastic optimization

Branch and bound

$$\min_x f(x)$$



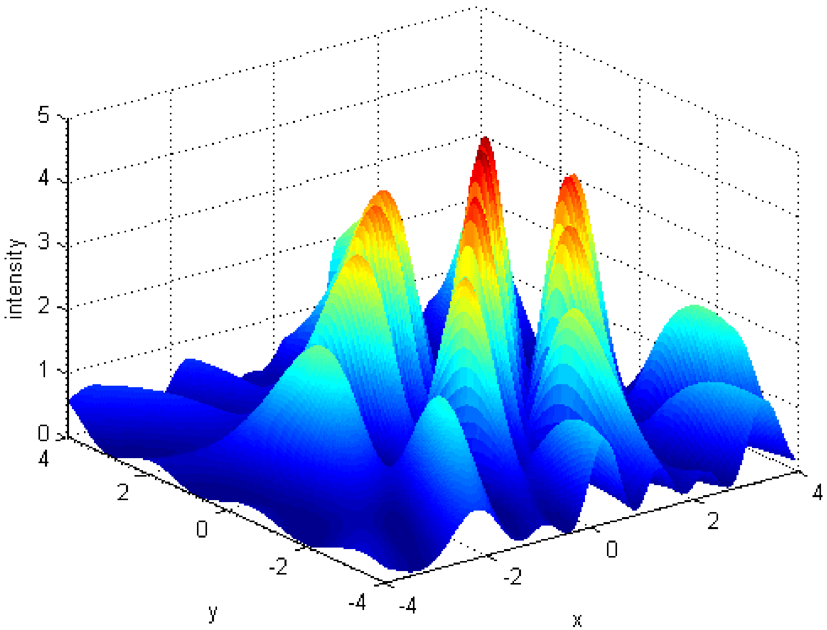
Key idea:

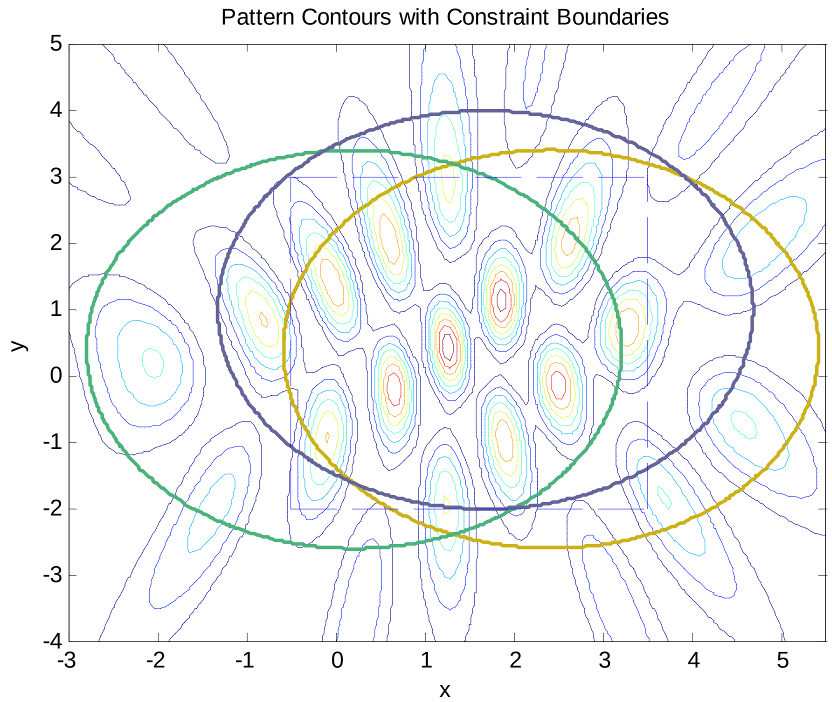
- Split region into sub-regions and compute bounds
- Consider two regions A and C
- If lower bound of A is greater than upper bound of C then A can be discarded
- divide (branch) regions and repeat

Multiple coverings

Key idea is to cover the parameter space with overlapping regions to deal with local optima, and then take advantage of efficient continuous optimization for each region.

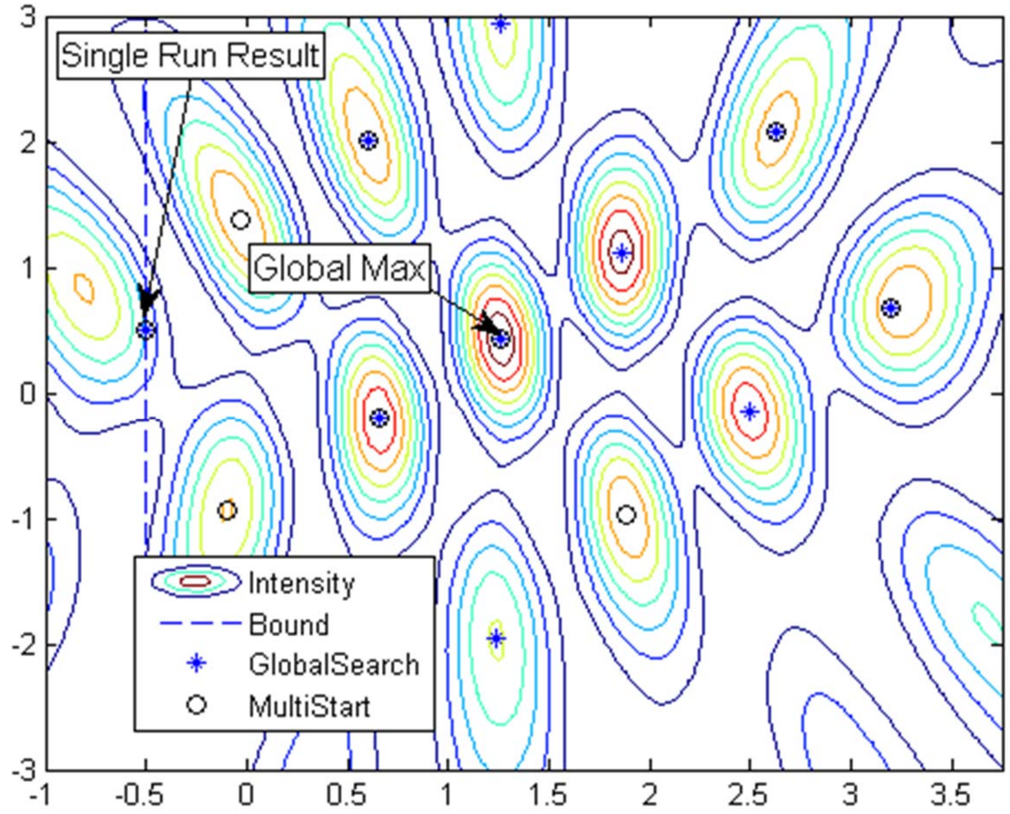
Example from Matlab Global Optimization toolbox



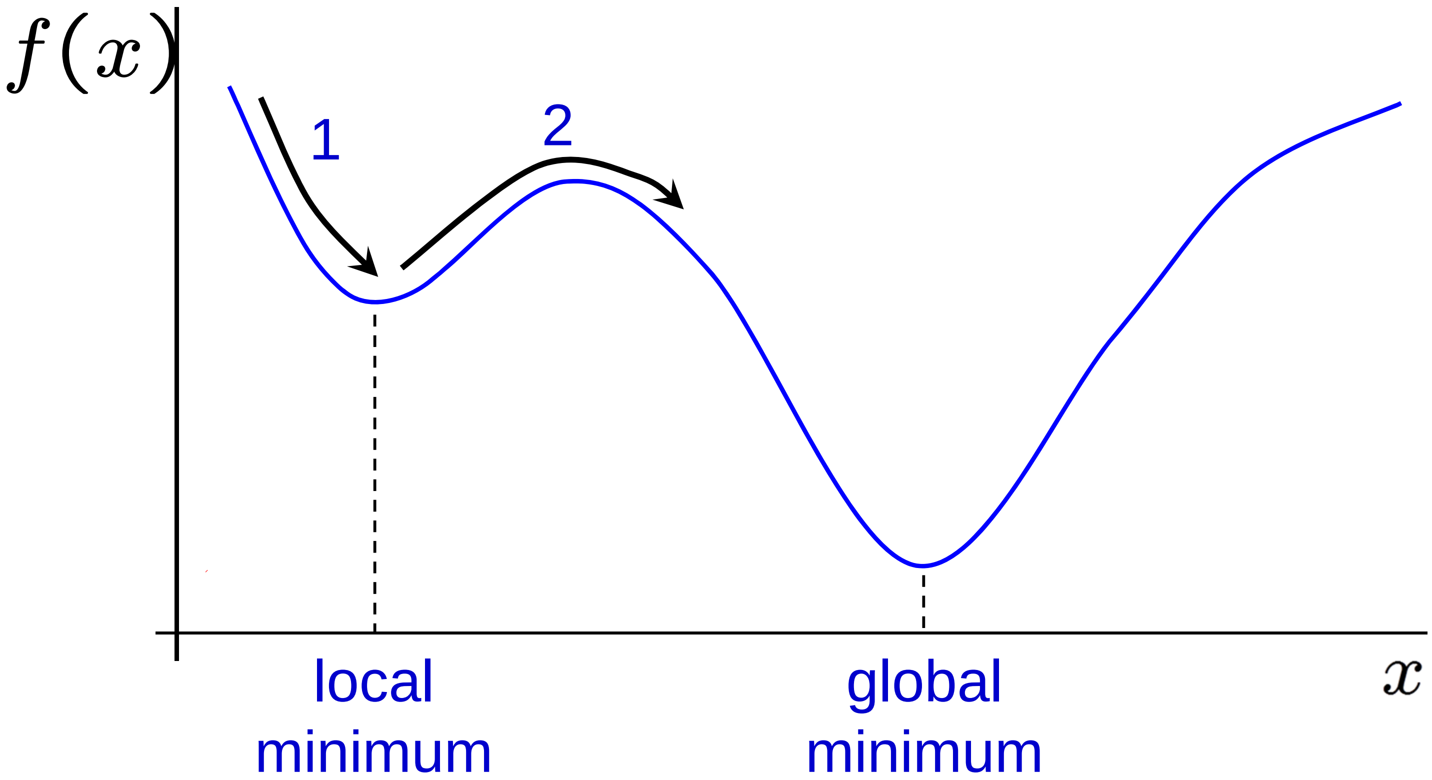


Multiple coverings ctd

- Use multiple starting points
- Continuous optimization method for each
- Record optimum for each starting point
- Sort values to find global optimum



Simulated Annealing



- The algorithm has a mechanism to jump out of local minima
- It is a stochastic search method, i.e. it uses randomness in the search

Simulated annealing algorithm

- At each iteration propose a move in the parameter space
- If the move decreases the cost, then accept it
- If the move increases the cost by ΔE, then
  - accept it with a probability $\propto \exp(-\Delta E/T)$,
  - Otherwise, don't move

  Note probability depends on temperature T
- Decrease the temperature according to a schedule so that at the start cost increases are likely to be accepted, and at the end they are not

Boltzmann distribution and the cooling schedule

- start with T high, then exp(-ΔE/T) is approx. 1, and all moves are accepted

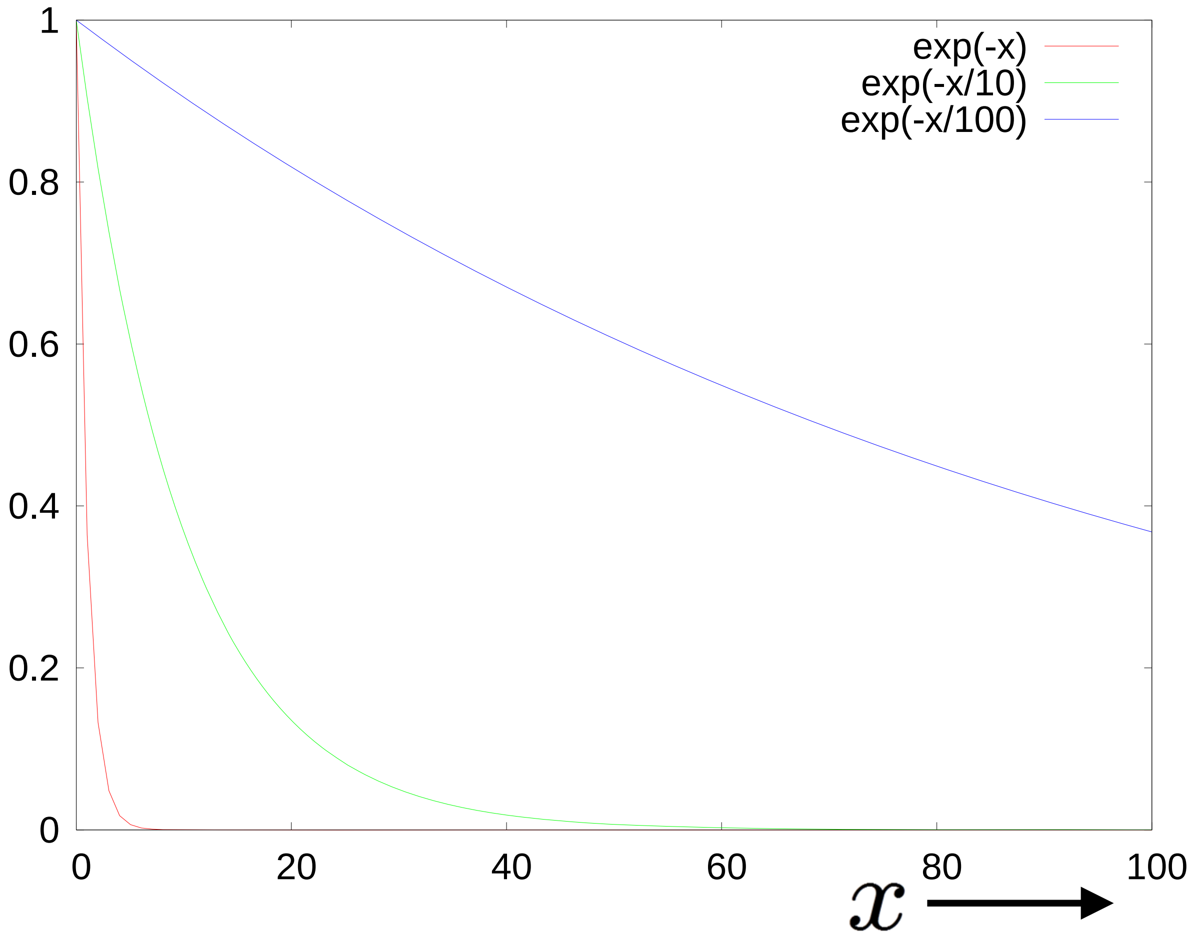
- many cooling schedules are possible, but the simplest is

$$T_{k+1} = \alpha T_k,\ 0 < \alpha < 1$$

where k is the iteration number

- The algorithm can be very slow to converge …

Boltzmann distribution exp(-ΔE/T)



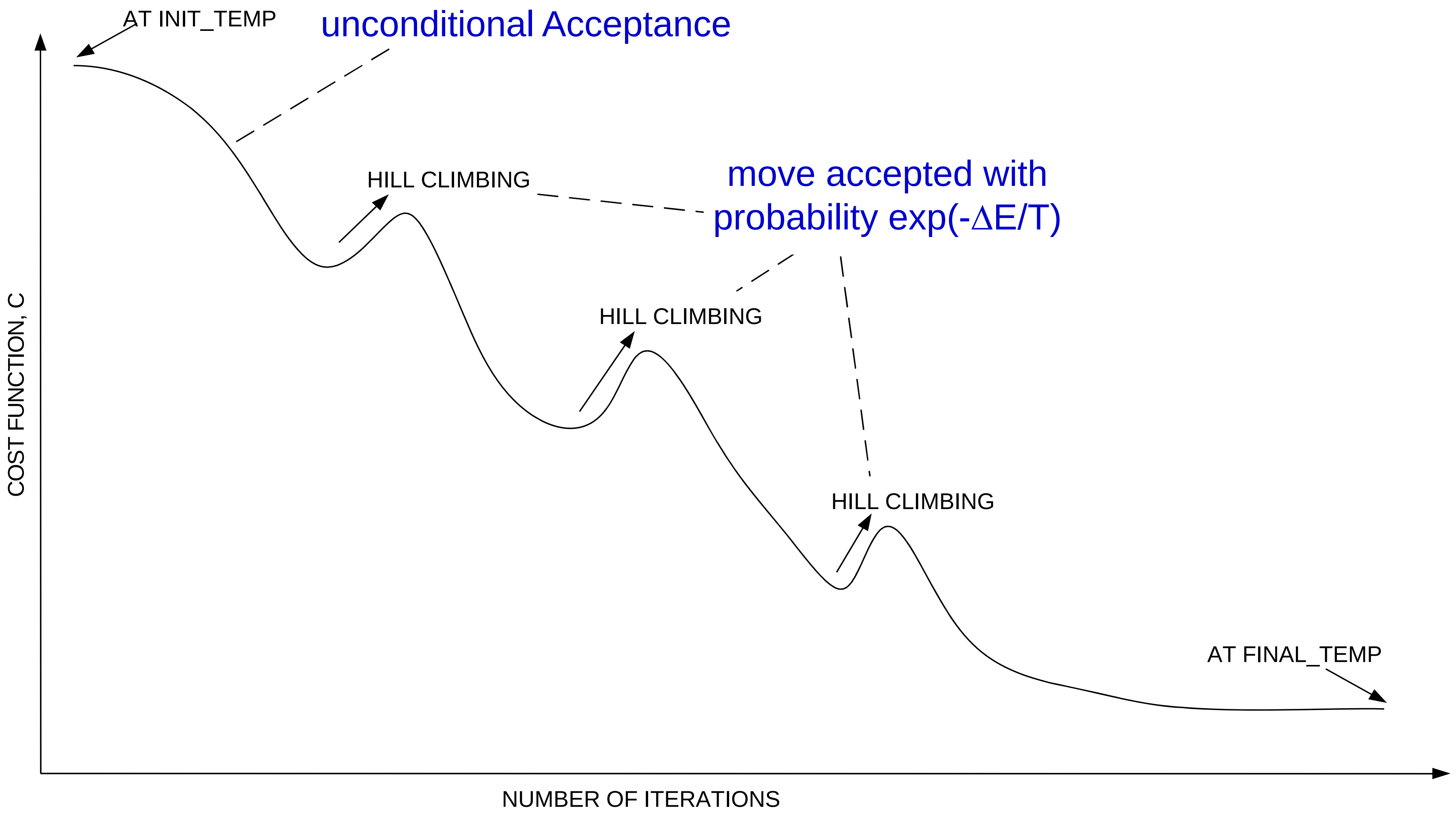
Simulated annealing

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects.

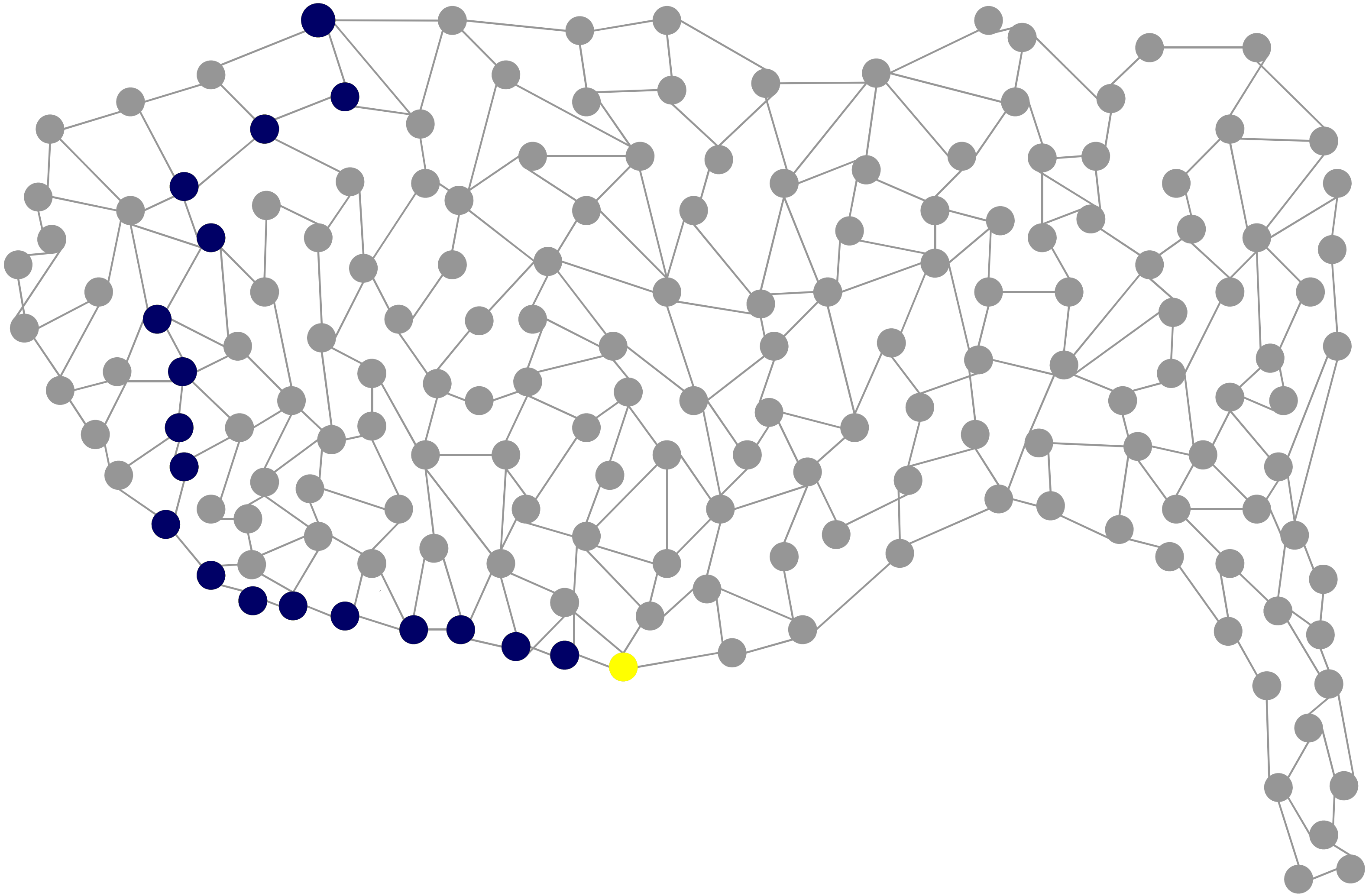
The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

Algorithms due to: Kirkpatrick *et al*. 1982; Metropolis *et al*.1953.

Example: Convergence of simulated annealing

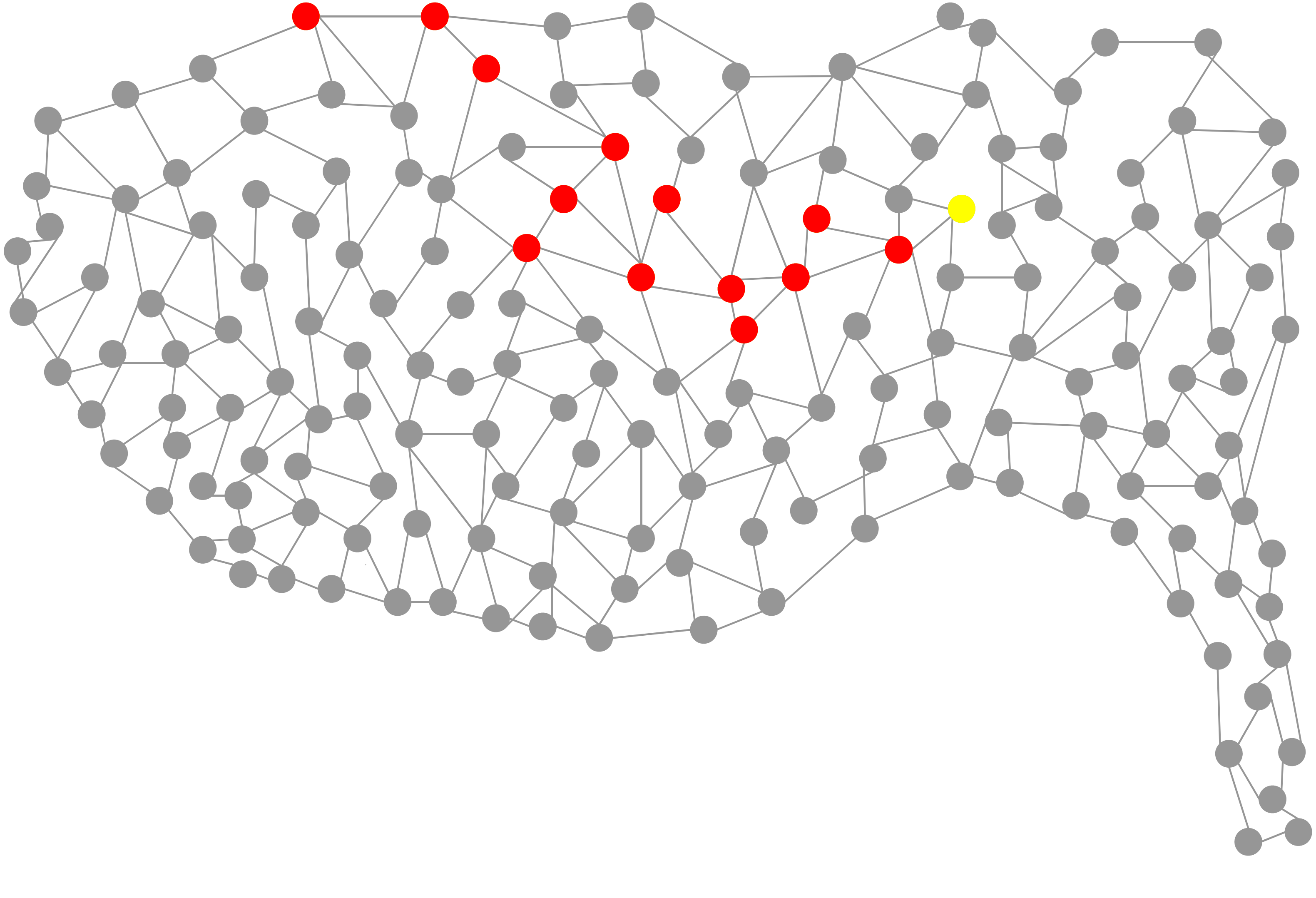


Steepest descent on a graph

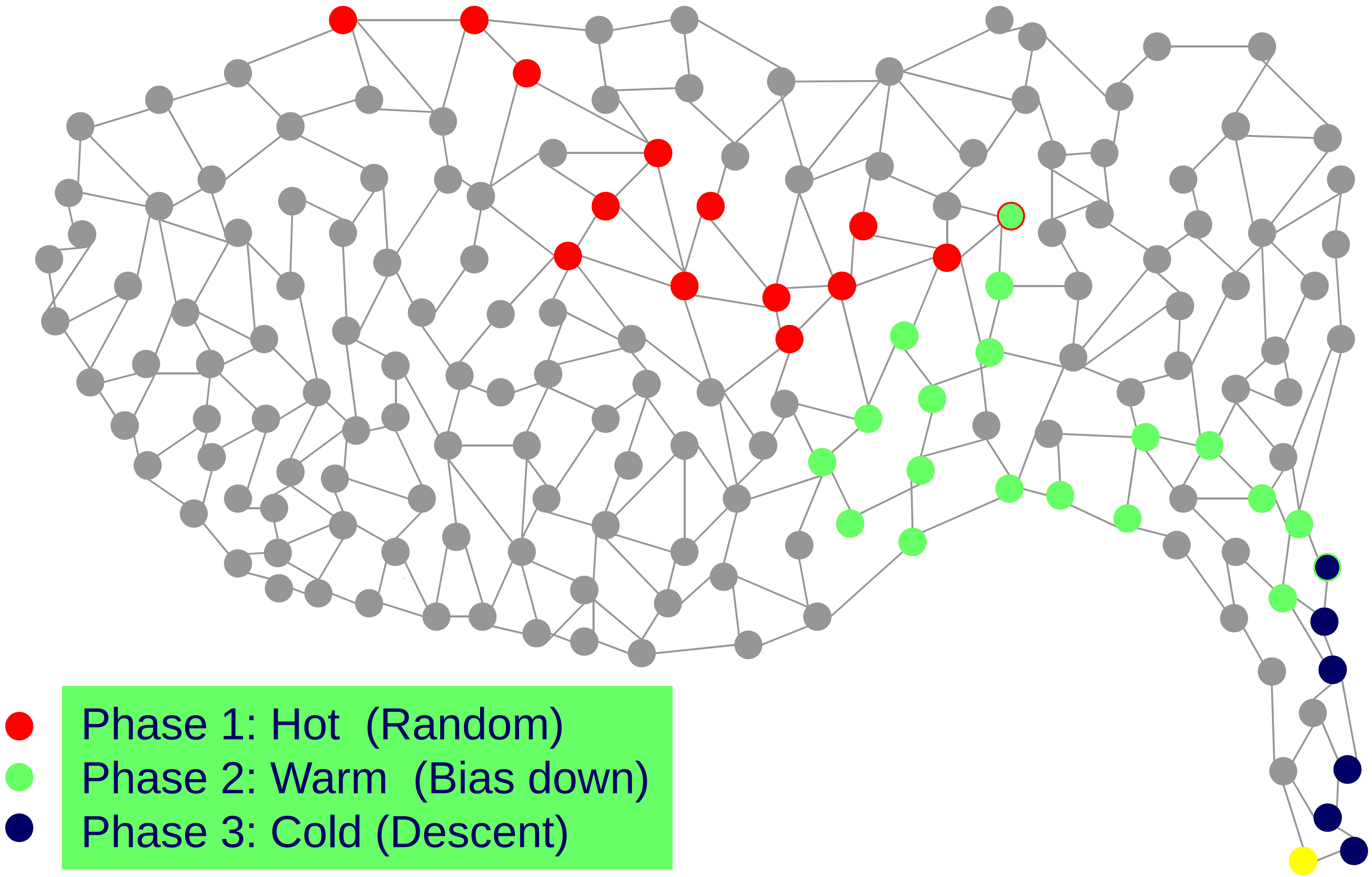


Slide from Adam Kalai and Santosh Vempala

Random Search on a graph



Simulated Annealing on a graph



Simulated annealing for the Rosenbrock function

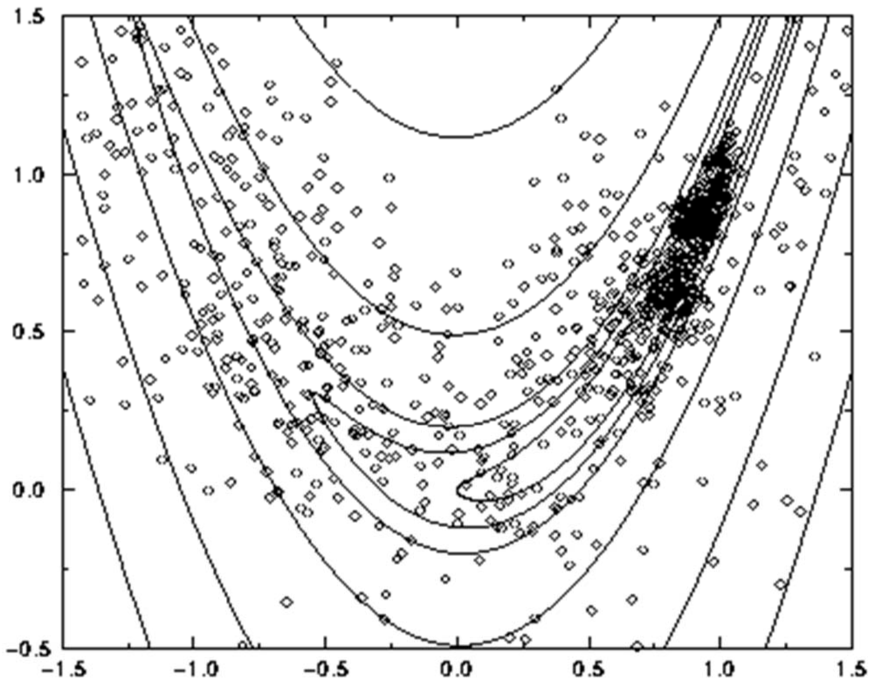


Figure 18: Minimization of the two-dimensional Rosenbrock function by simulated annealing--search pattern.

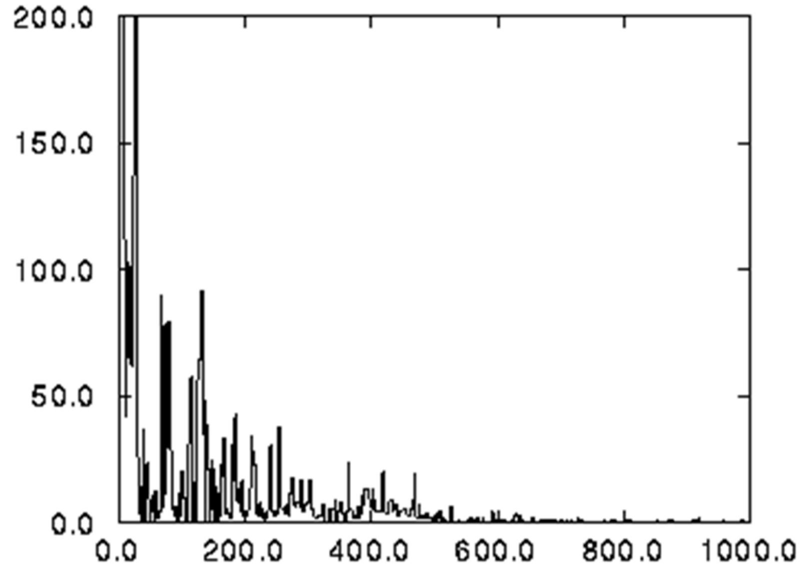


Figure 19: Minimization of the two-dimensional Rosenbrock function by simulated annealing objective reduction.

Figures from http://www.phy.ornl.gov/csep/CSEP/MO/MO.html

There is more …

There are many other classes of optimization problem, and also many efficient optimization algorithms developed for problems with special structure. Examples include:

- Combinatorial and discrete optimization
- Dynamic programming
- Max-flow/Min-cut graph cuts
- …

See the links on the web page

http://www.robots.ox.ac.uk/~az/lectures/b1/index.html

and come to the C Optimization lectures next year