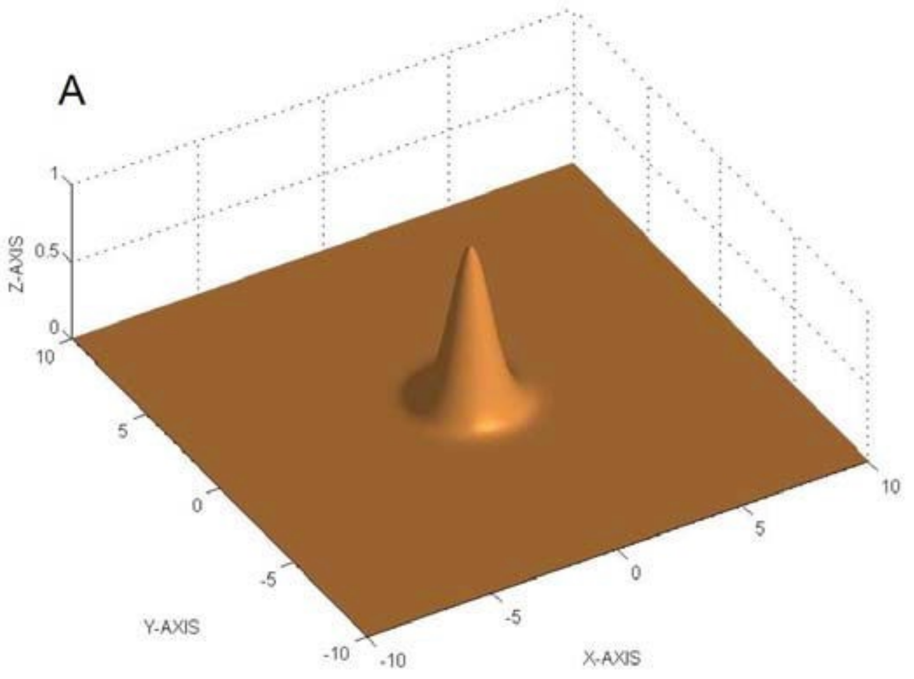
Lecture 4

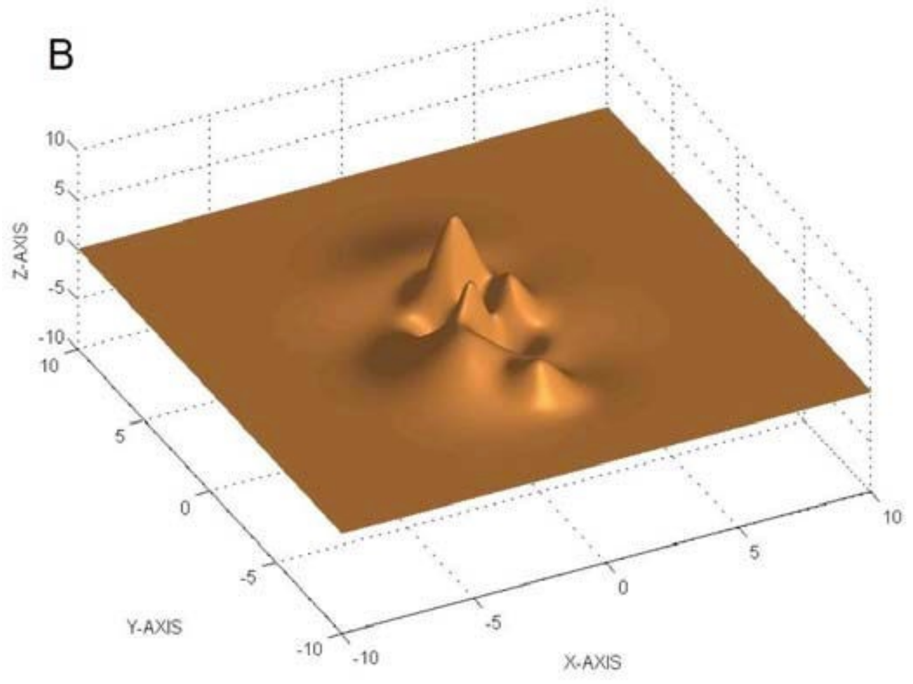
B1 Optimization - Michaelmas 2023 - V. A. Prisacariu

- Convexity
- Robust cost functions
- Optimizing non-convex functions:
    - grid search
    - branch and bound
    - simulated annealing
    - evolutionary optimization

Cost Functions in 2D



One (global) optima



Multiple optima

- 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.

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

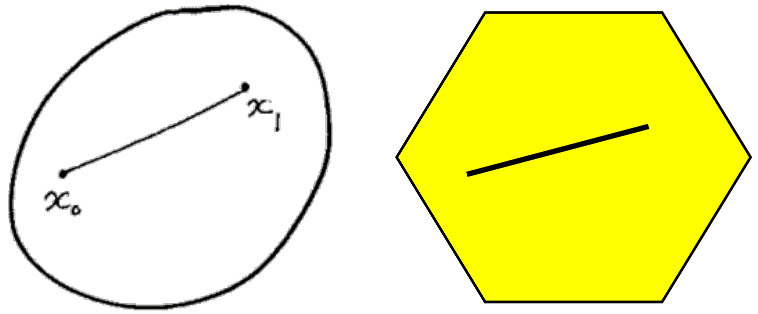
- Convex Sets, and
- Convex Functions

[Note – sketch introduction only]

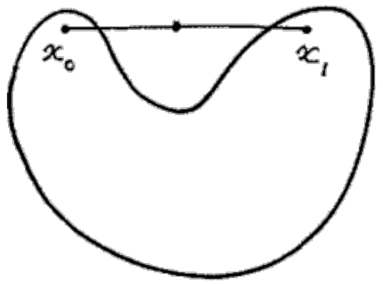
Convex Set

A set $D \subset \mathbb{R}^n$ is convex if the line joining points x_0 and x_1 lines inside D.

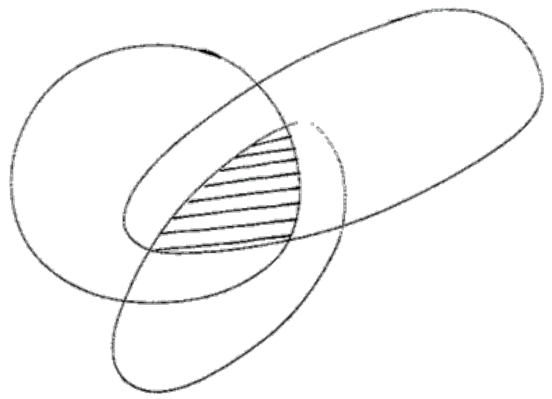
Convex



Not-convex



Intersection of convex sets is convex.



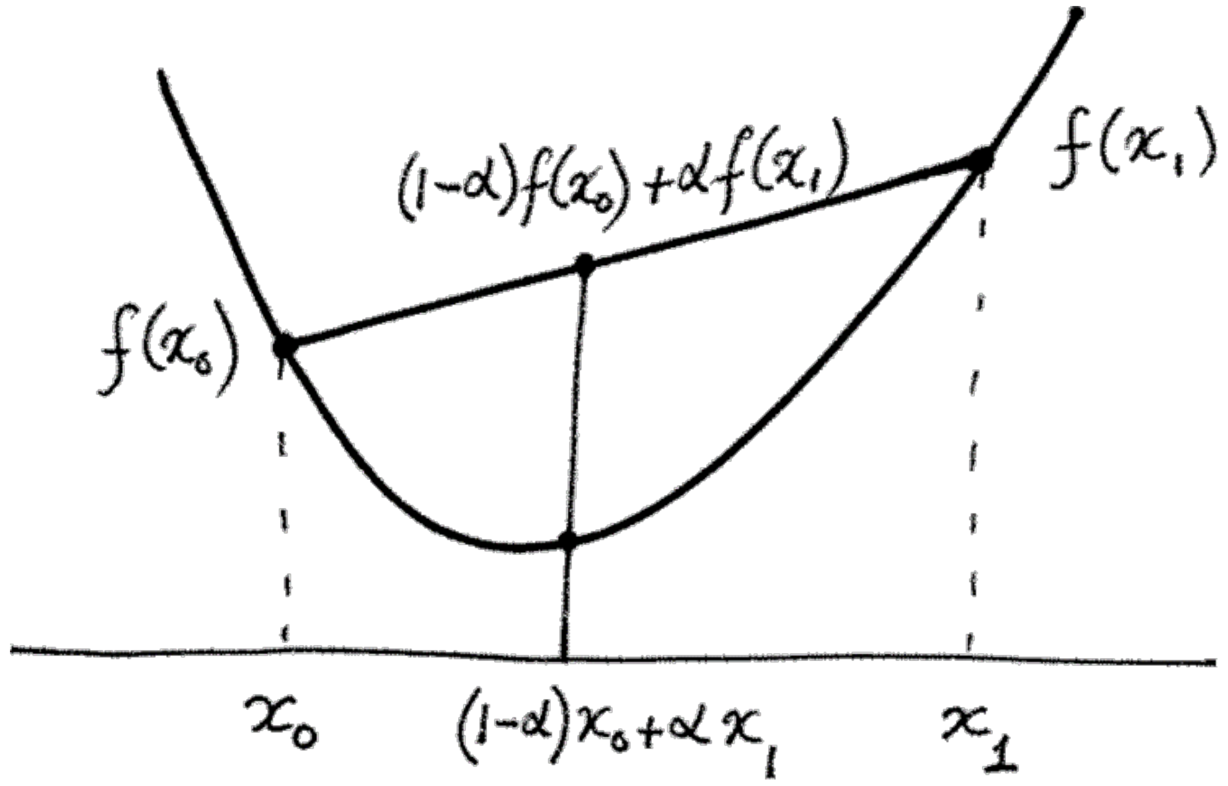
Convex Functions

D – a domain in $\mathbb{R}^n$

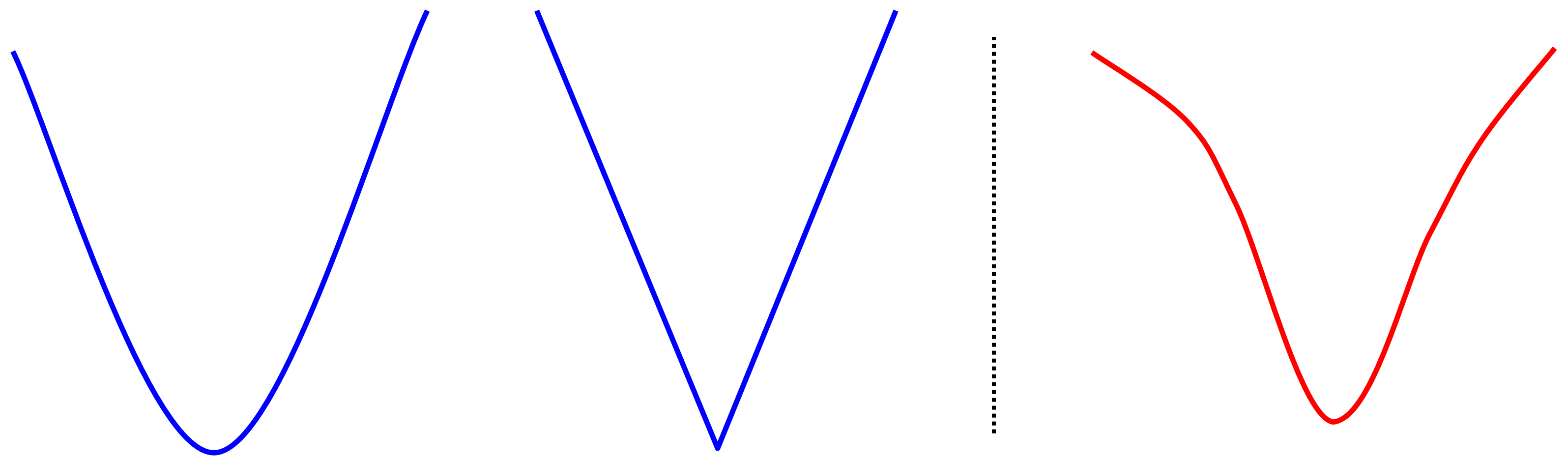
A convex function $f: D \to \mathbb{R}$ is one that satisfies, for any x_0 and x_1 in D:

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

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



Convex Functions Example



A non-negative sum of convex functions is convex

Convex Optimization Problem

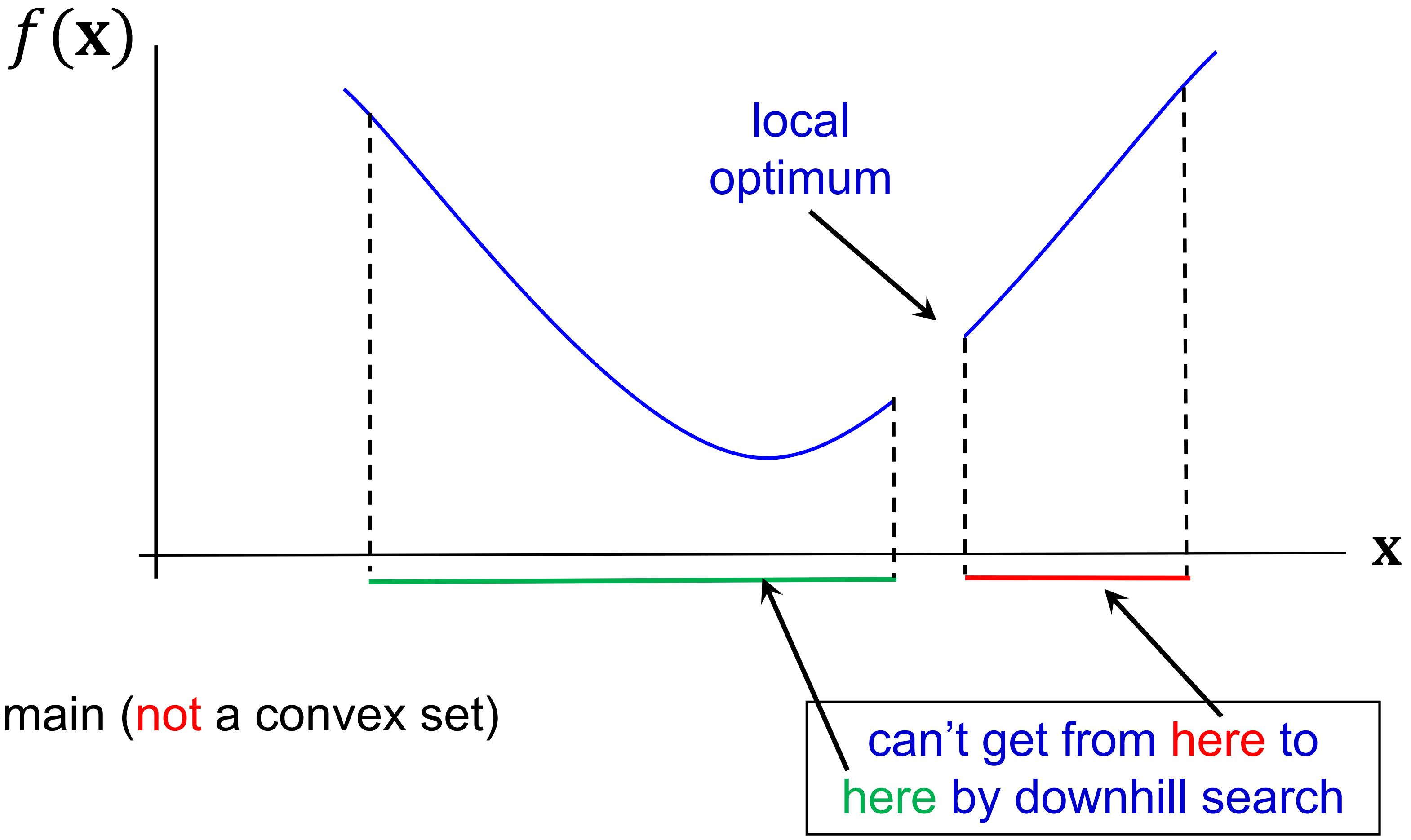
Minimize:

- 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:

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

3. Quadratic Functions:

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

Many more useful examples, see Boyd & Vandenberghe.

Examples of convex functions

Examples in 1D:

- Affine: $ax + b$, for any $a, b \in \mathbb{R}$
- Exponential: e^{ax}, for any $a \in \mathbb{R}$
- Power of absolute values: $|x|^p$, for $p \geq 1$

Examples in nD:

- Affine function: $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + b$
- Norms: $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$, for $p \geq 1$.

First-order Condition

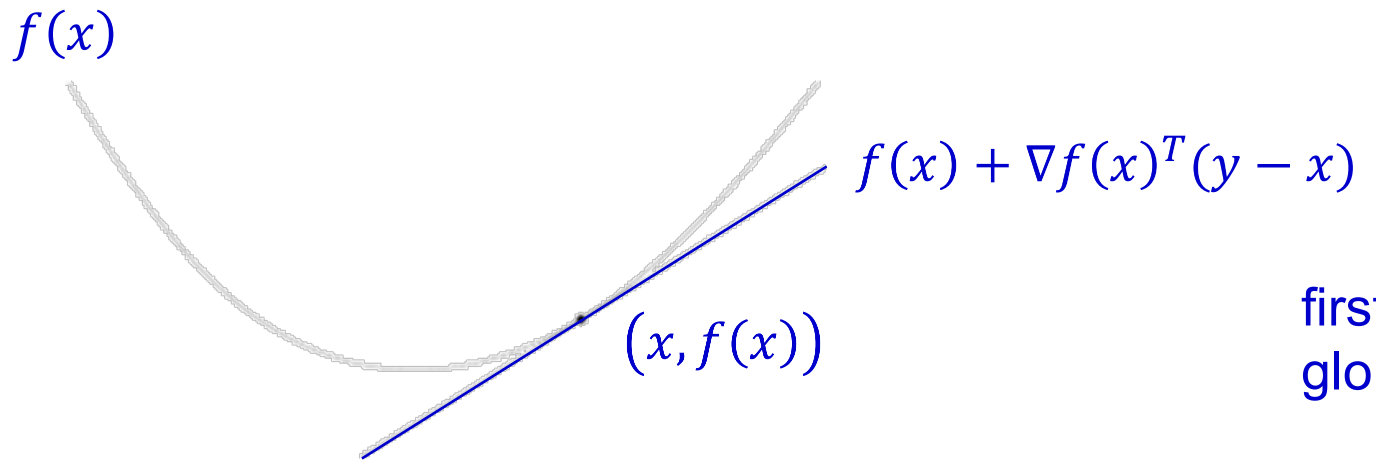
f is differentiable if $\mathbf{dom}\, f$ is open and the gradient

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

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

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

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



first-order approximation of f is global underestimator

Second-order Condition

The Hessian function $f(x_1, x_2, \dots, x_n)$ is the matrix of partial derivatives:

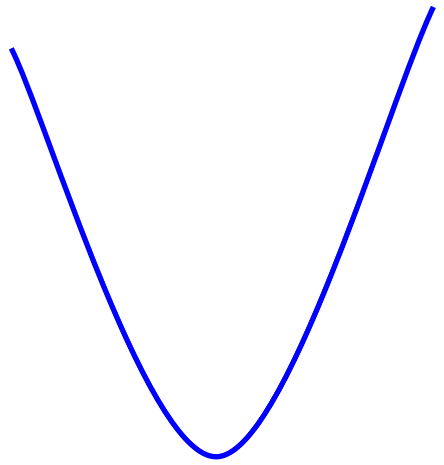
$$\mathrm{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

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

Strictly Convex

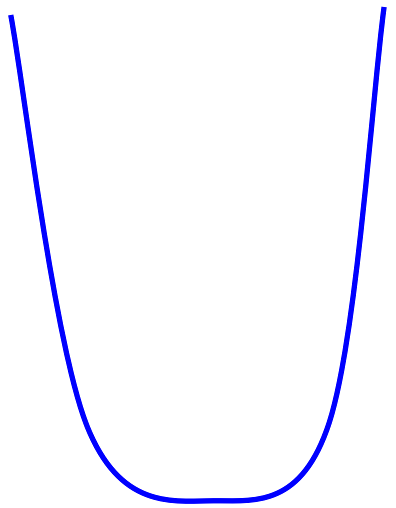
A function $f(\mathbf{x})$ is strictly convex if

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



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.

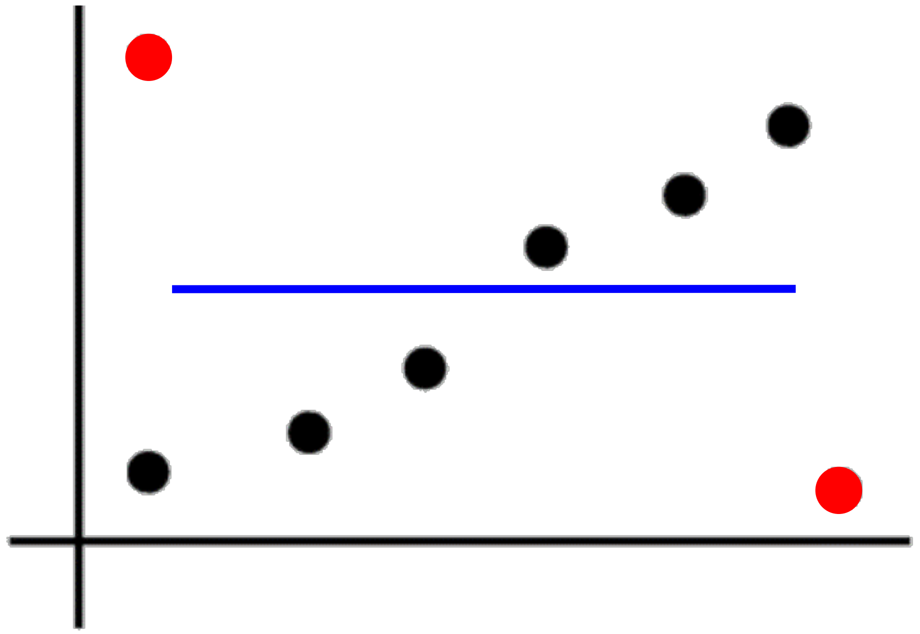
The cost function can be chosen to be:

- Convex.
- Robust to noise (outliers) in the data/measurements.

Motivation

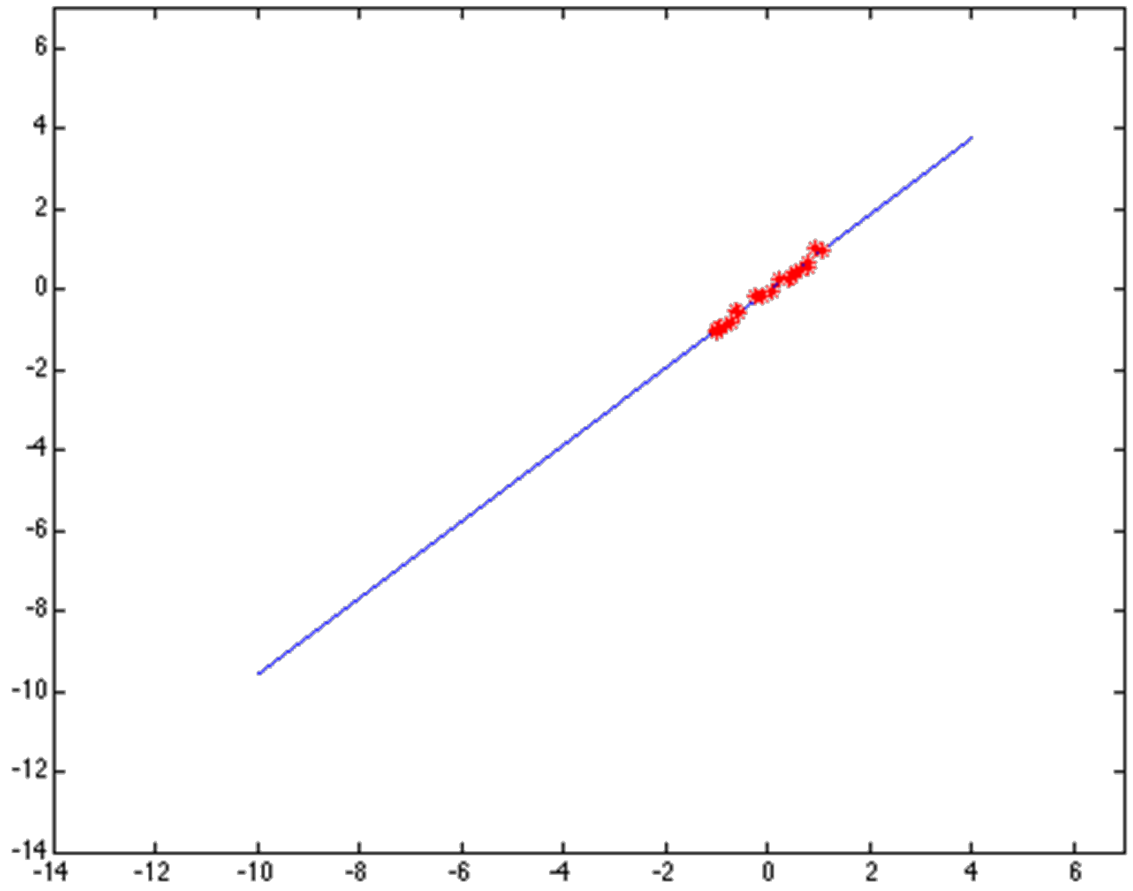
Fitting a 2D line to a set of 2D points

- 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



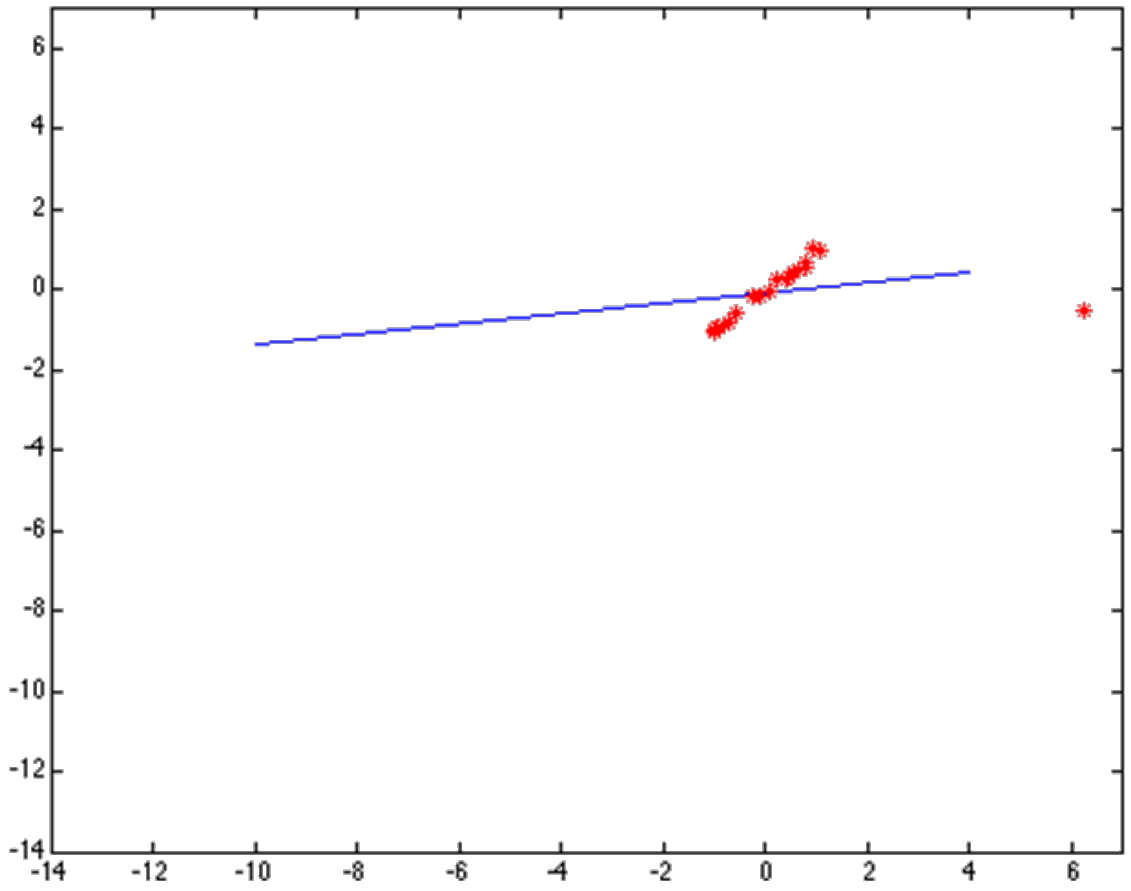
Least squares: Robustness to Noise

Least squares fit to the red points:



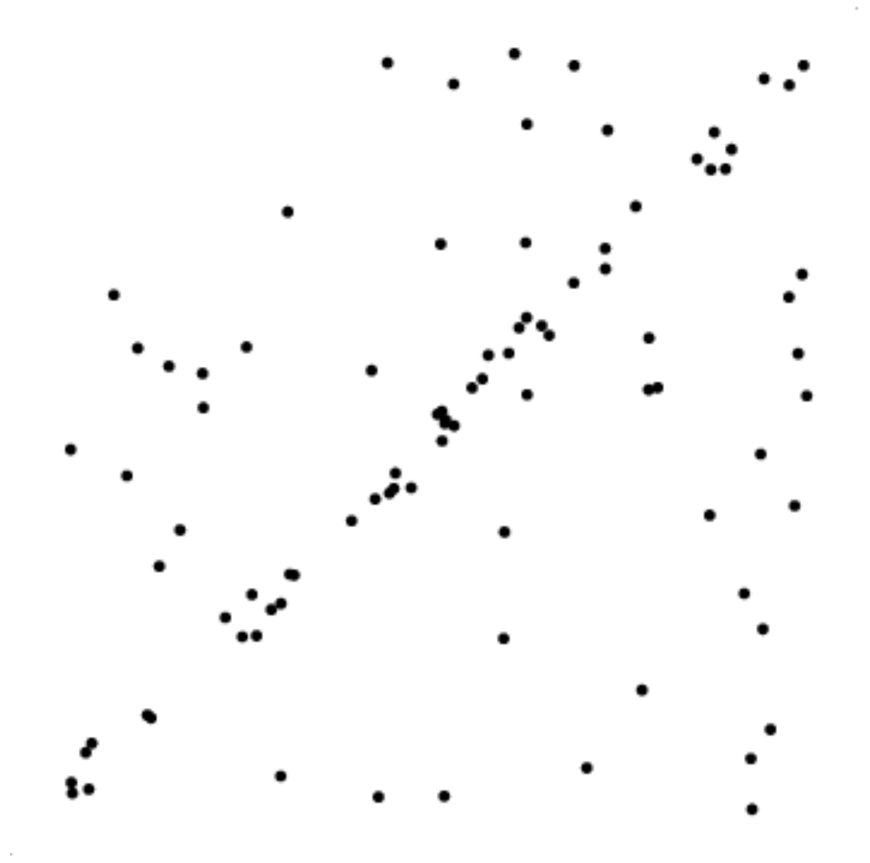
Least squares: Robustness to Noise

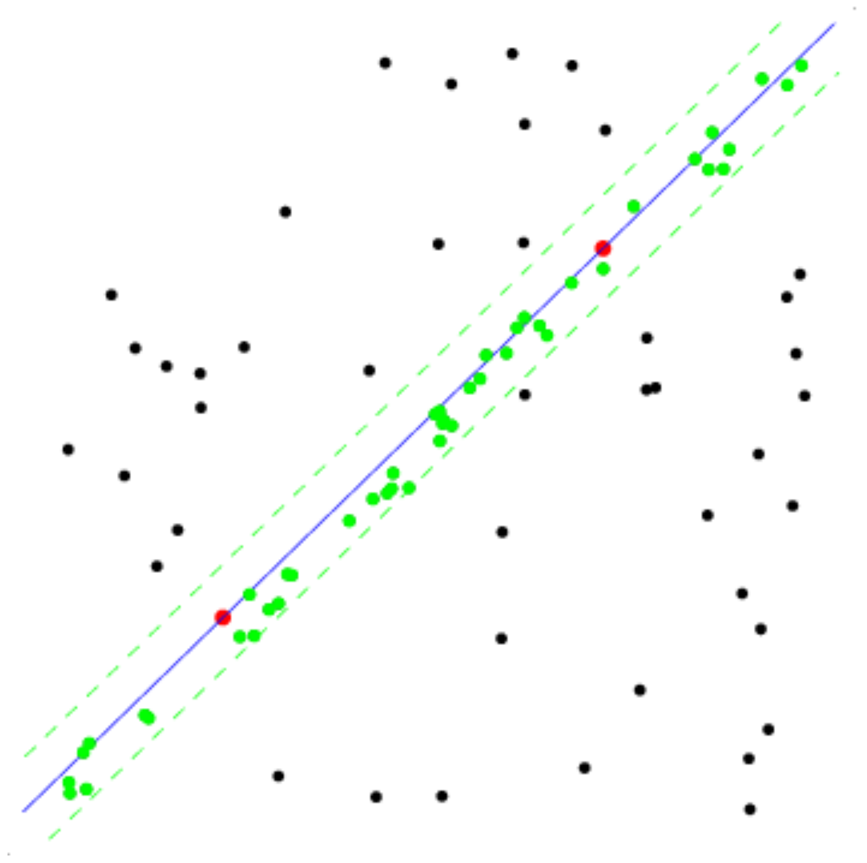
Least squares fit to the red points:



Problem: squared error heavily penalizes outliers

Least squares: Robustness to Noise

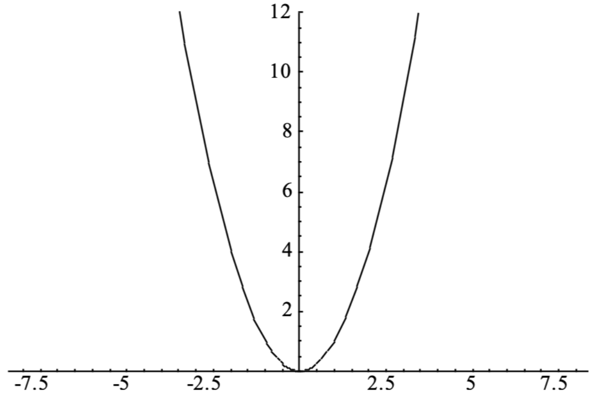


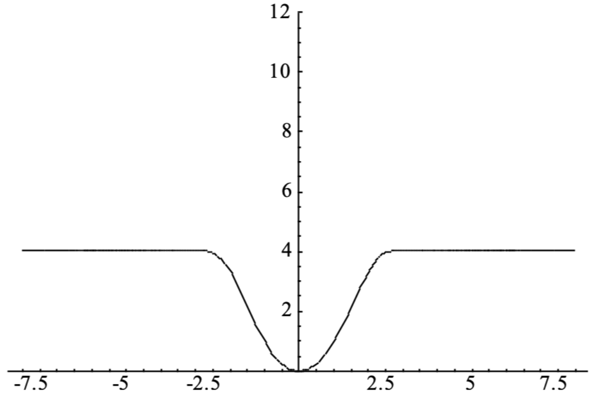


Least squares: Robustness to Noise

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

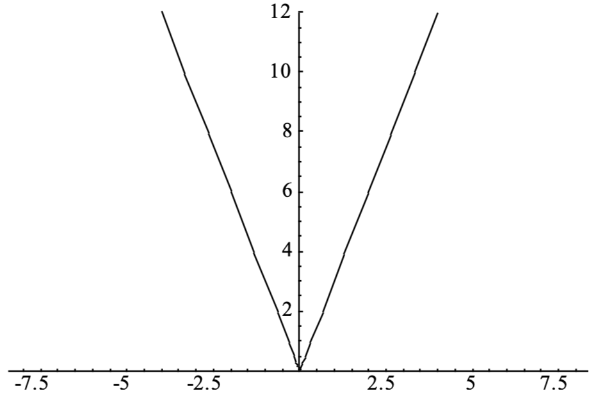
Quadratic

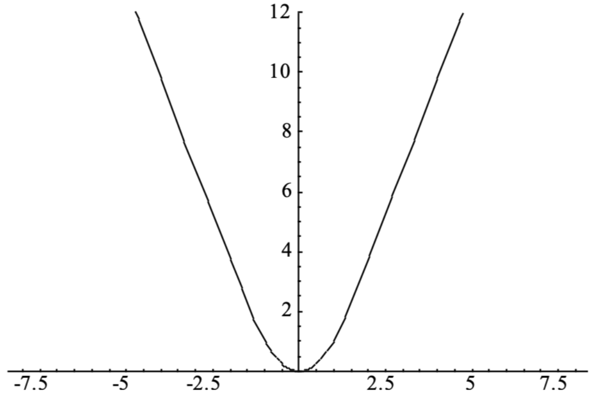




Truncated quadratic

L_1



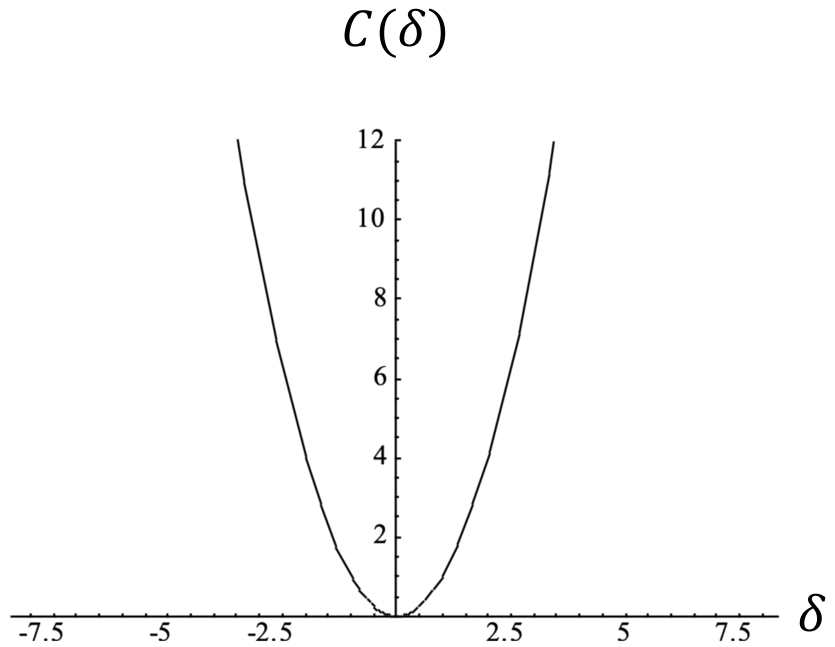


Huber

Quadratic Cost Function

Squared error – the usual default cost function

Convex.



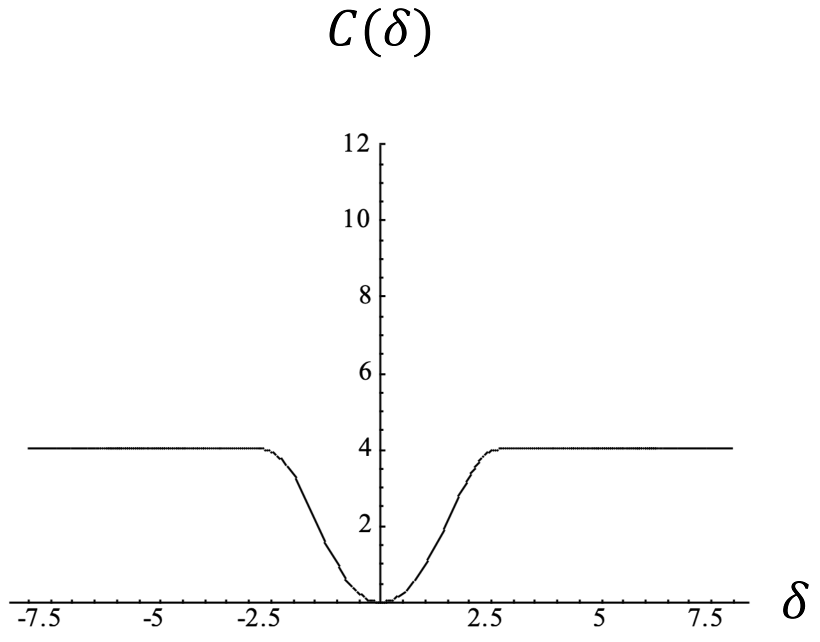
$C(\delta) = \delta^2$

Truncated Quadratic Cost Function

For inliers behaves as a quadratic.

Truncated so that outliers only incur a fixed cost.

Non-convex.



$$C(\delta) = \min(\delta^2, a) = \begin{cases} \delta^2 \text{ if } |\delta| < \sqrt{\alpha} \\ \alpha \end{cases}$$

L_1 Cost Function

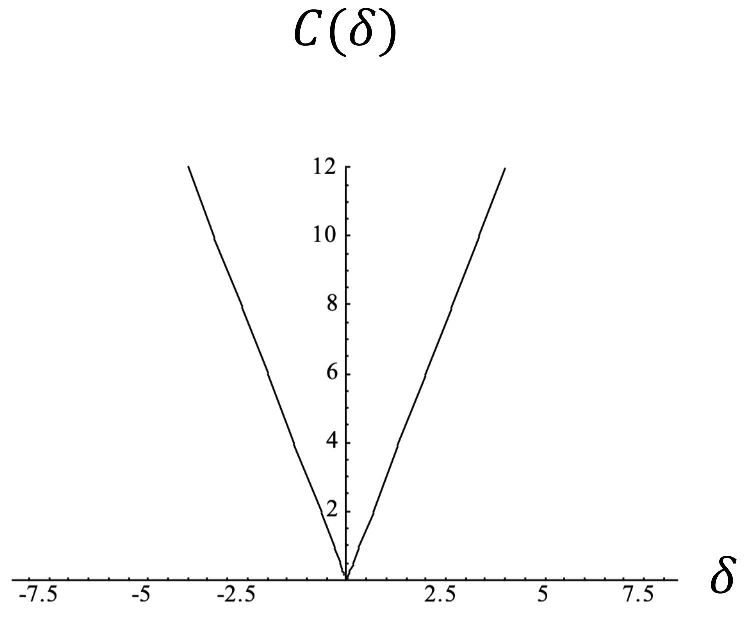
Absolute error.

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.

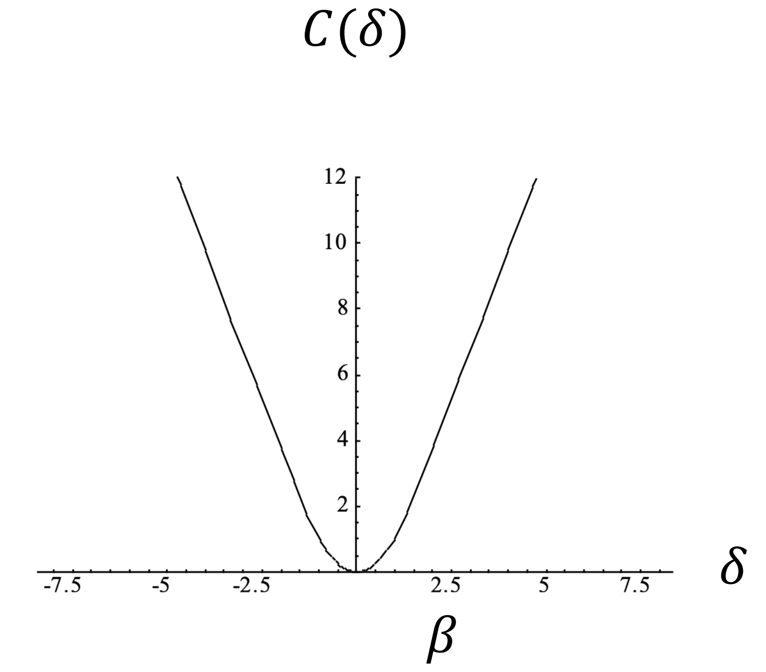
Continuous first derivative.

For small values is quadratic.

For larger values becomes linear.

Thus has the outlier stability of L_1.

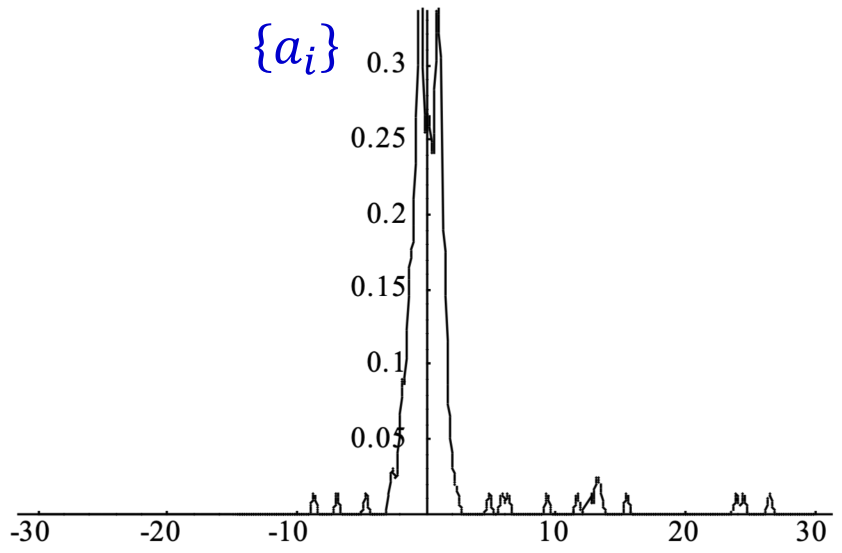
Convex.



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

Example 1: Measurement with Outliers

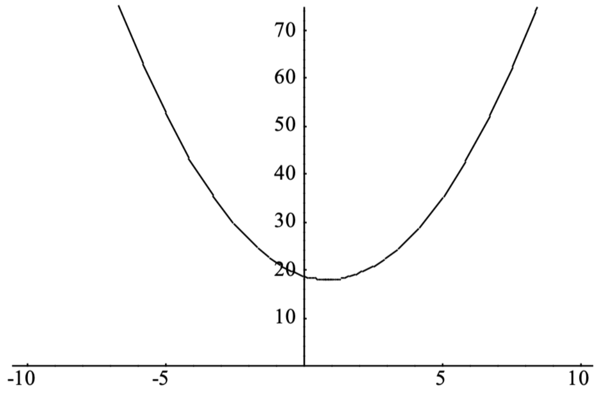
Consider minimizing the data $\{a_i\}$

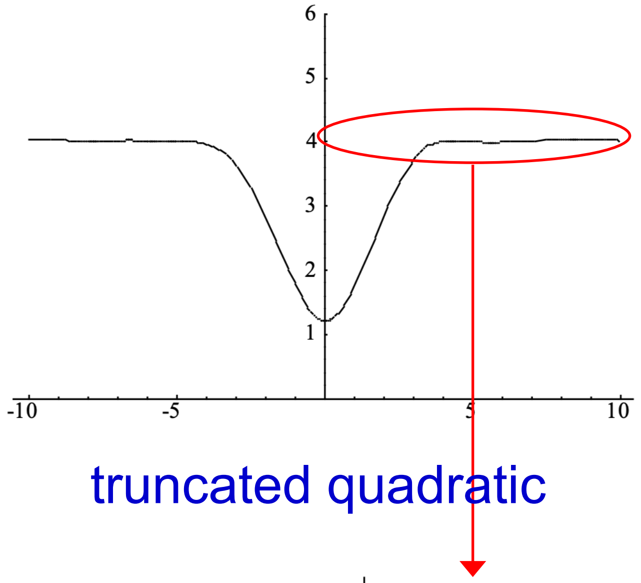


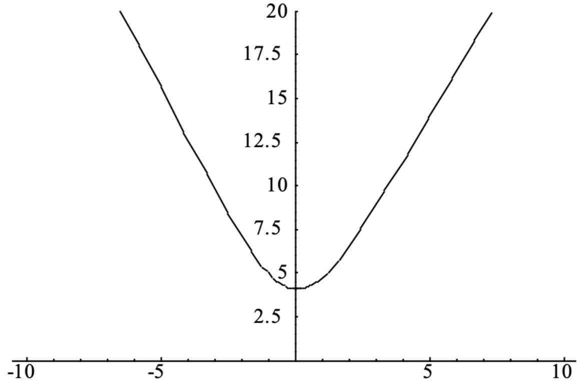
- The data $\{a_i\}$ may be thought of as repeated measurements of a fixed value (at 0), subject to Gaussian noise and some outliers.
- It has 10% of outliers biased towards the right of the true value.
- The minimum of $f(x)$ does not correspond to the true value.

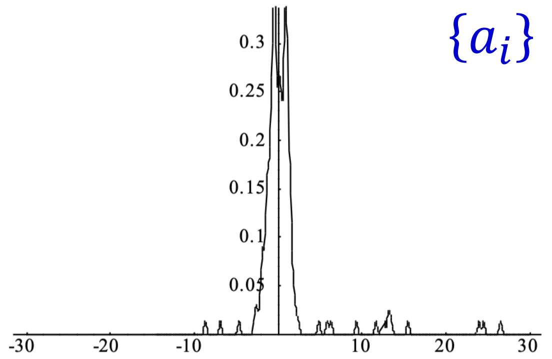
Example 1: Measurement with Outliers

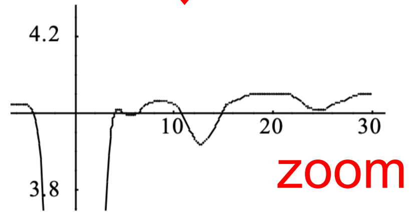
$$f(x) = \sum_i C(|x - a_i|)$$





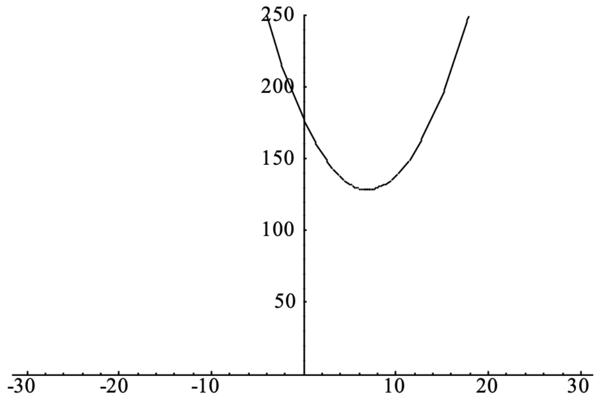




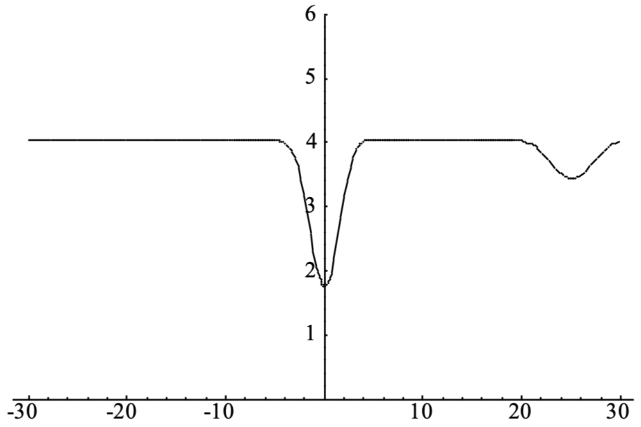


Example 2: Measurement with Outliers

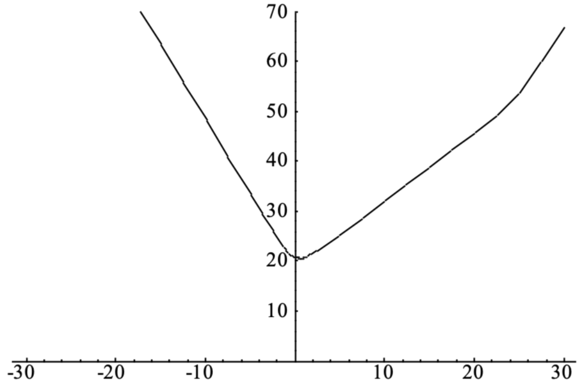
$$f(x) = \sum_i C(|x - a_i|)$$



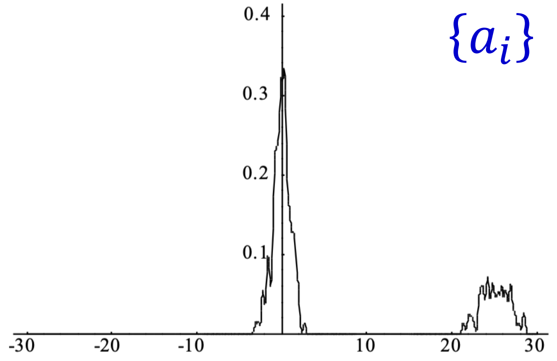
quadratic



truncated quadratic



Huber



$\{a_i\}$

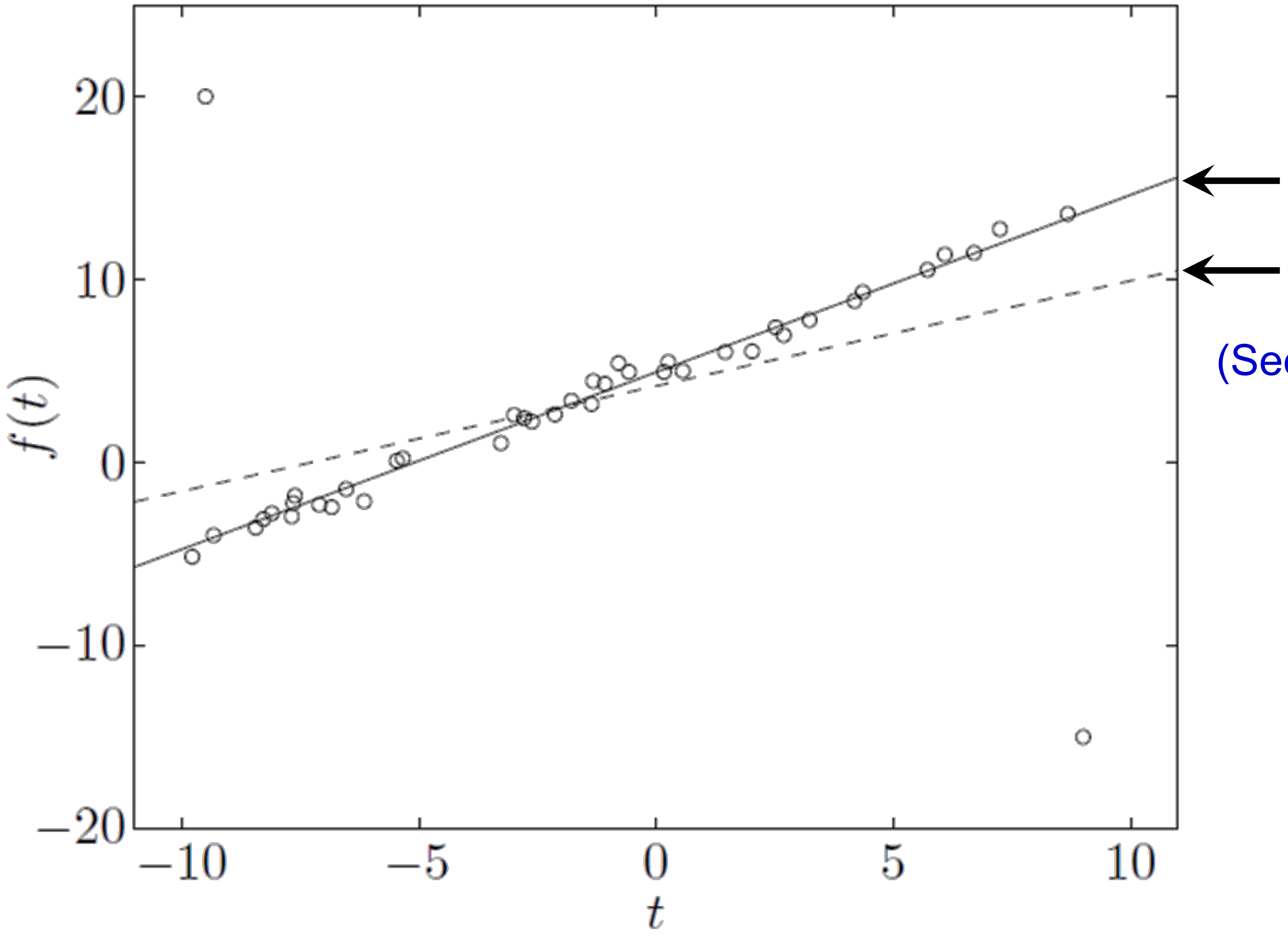
bimodal measurements

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

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)

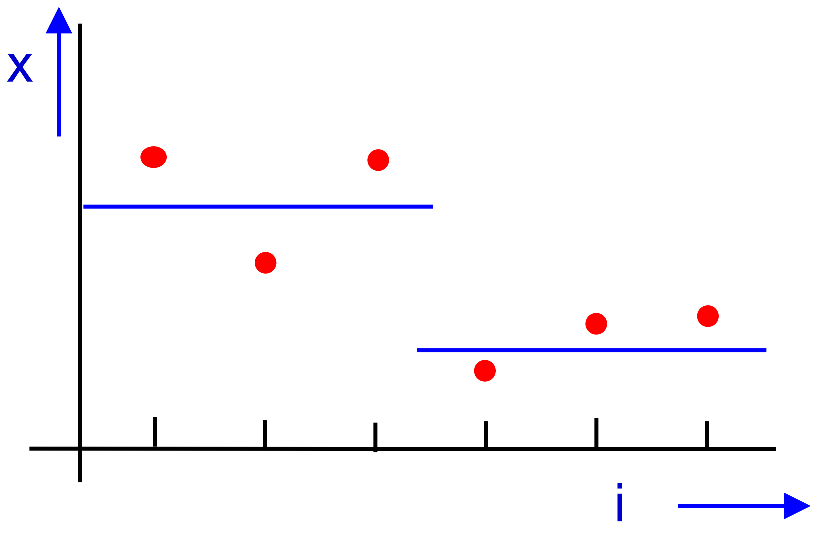
Boyd & Vandenberghe

Application 2: Signal Restauration

Measurements z_i are original signal x_i corrupted with additive noise

$$z_i = x_i + w_i$$

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



Quadratic Smoothing

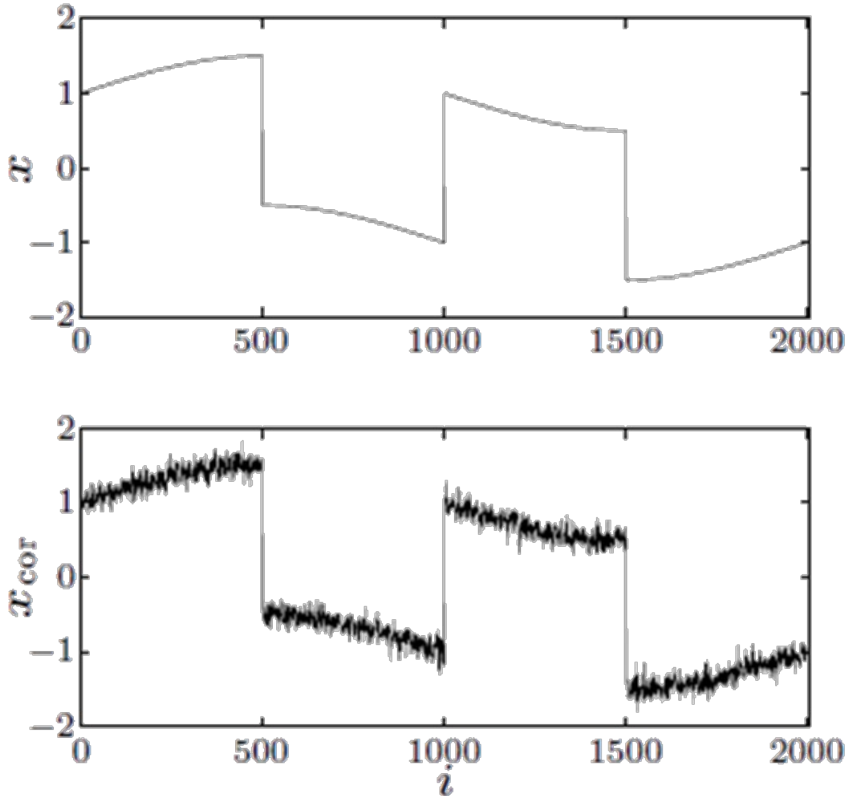
$$f(\mathbf{x}) = \sum_i (z_i - x_i)^2 + \lambda (x_i - x_{i-1})^2$$

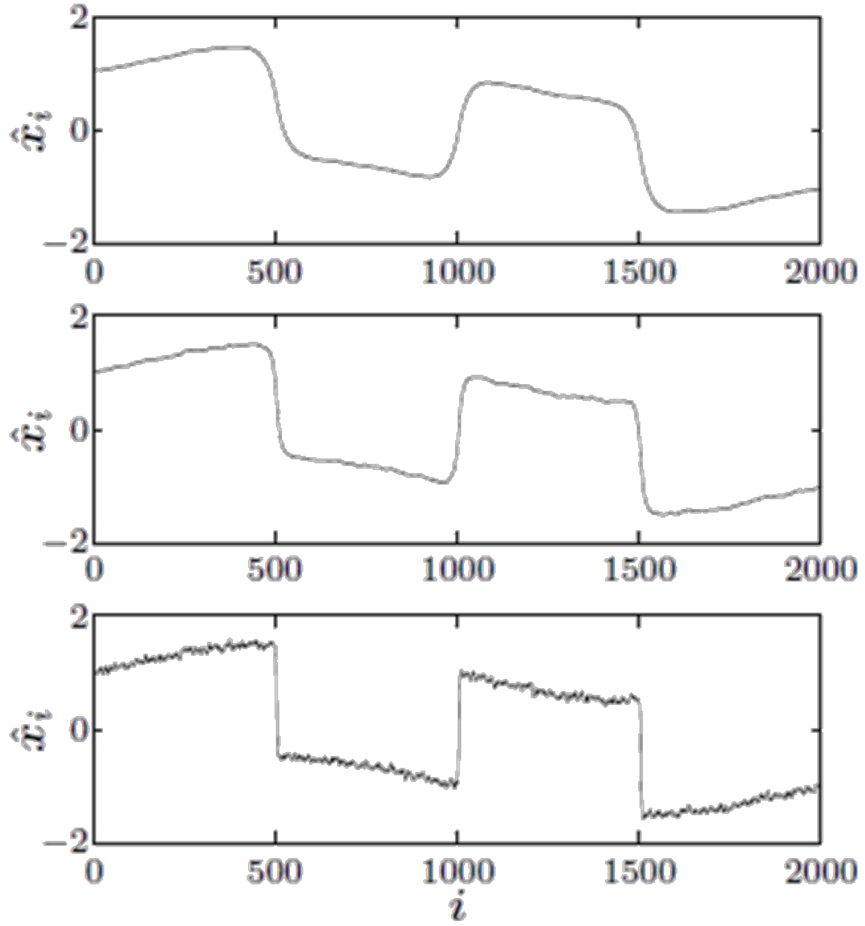
Total Variation

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

Application 2: Signal Restauration: Quadratic Smoothing

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



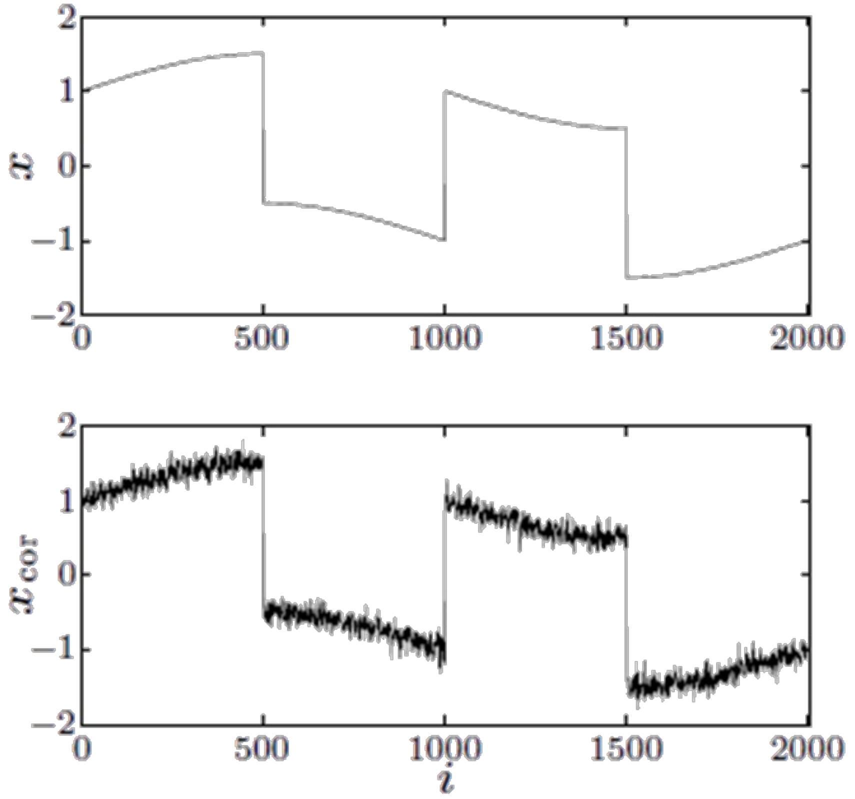


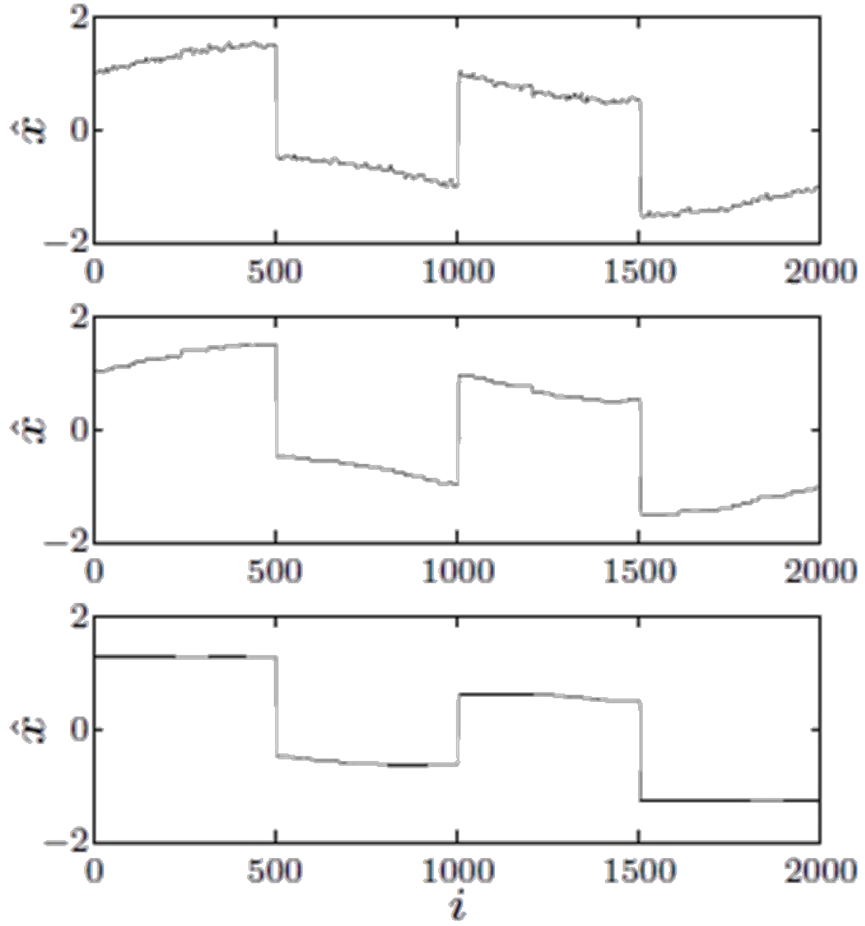
Quadratic smoothing smooths out noise **and** steps in the signal

Boyd & Vandenberghe

Application 2: Signal Restauration: Total Variation

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



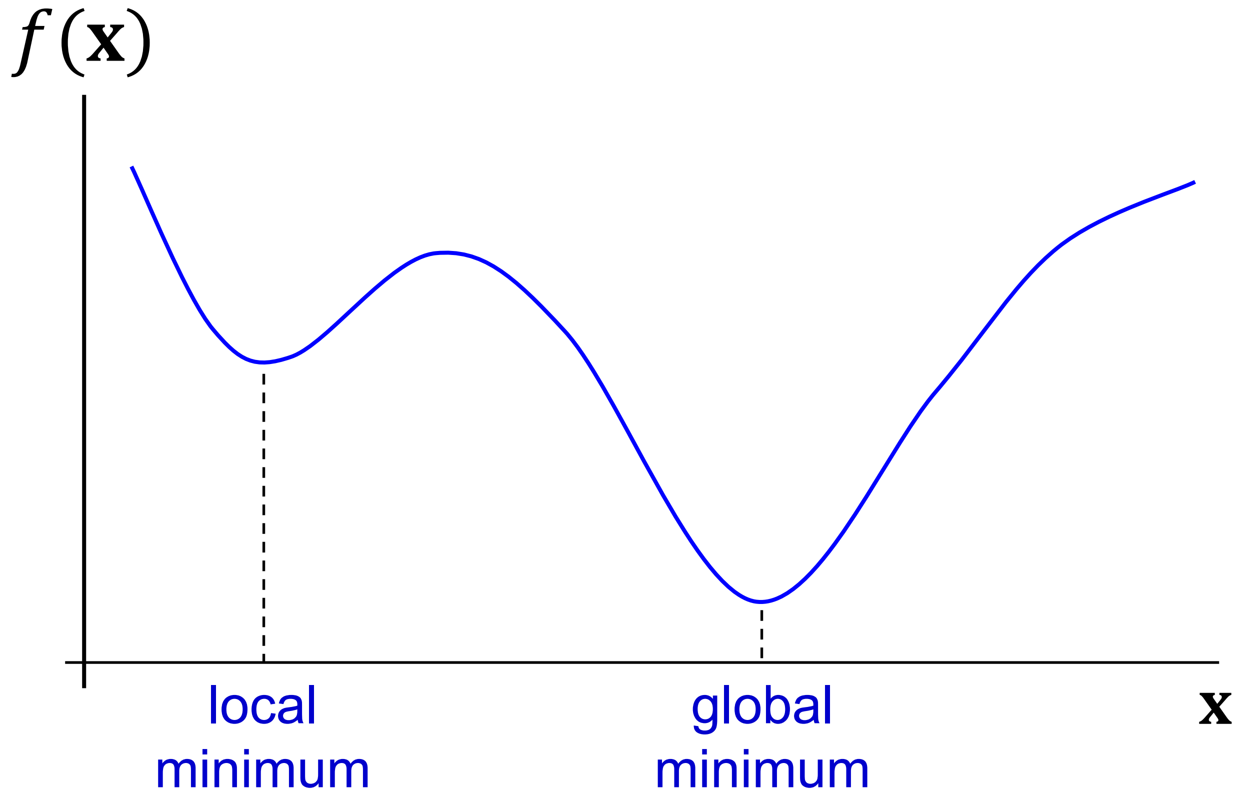


Total variation smoothing preserves steps in the signal

Boyd & Vandenberghe

Optimizing Non-Convex Functions

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

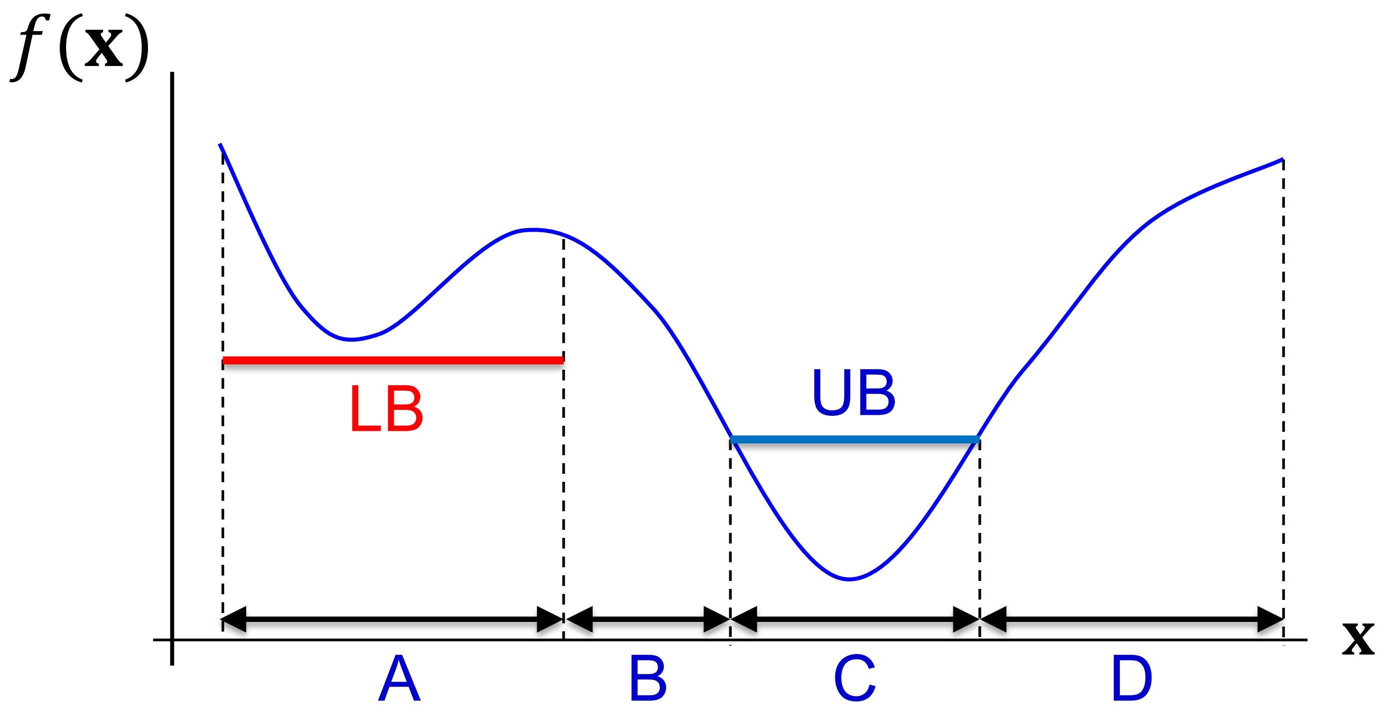


Sketch three methods:

1. branch and bound
2. simulated annealing
3. evolutionary optimization

Branch and Bound

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

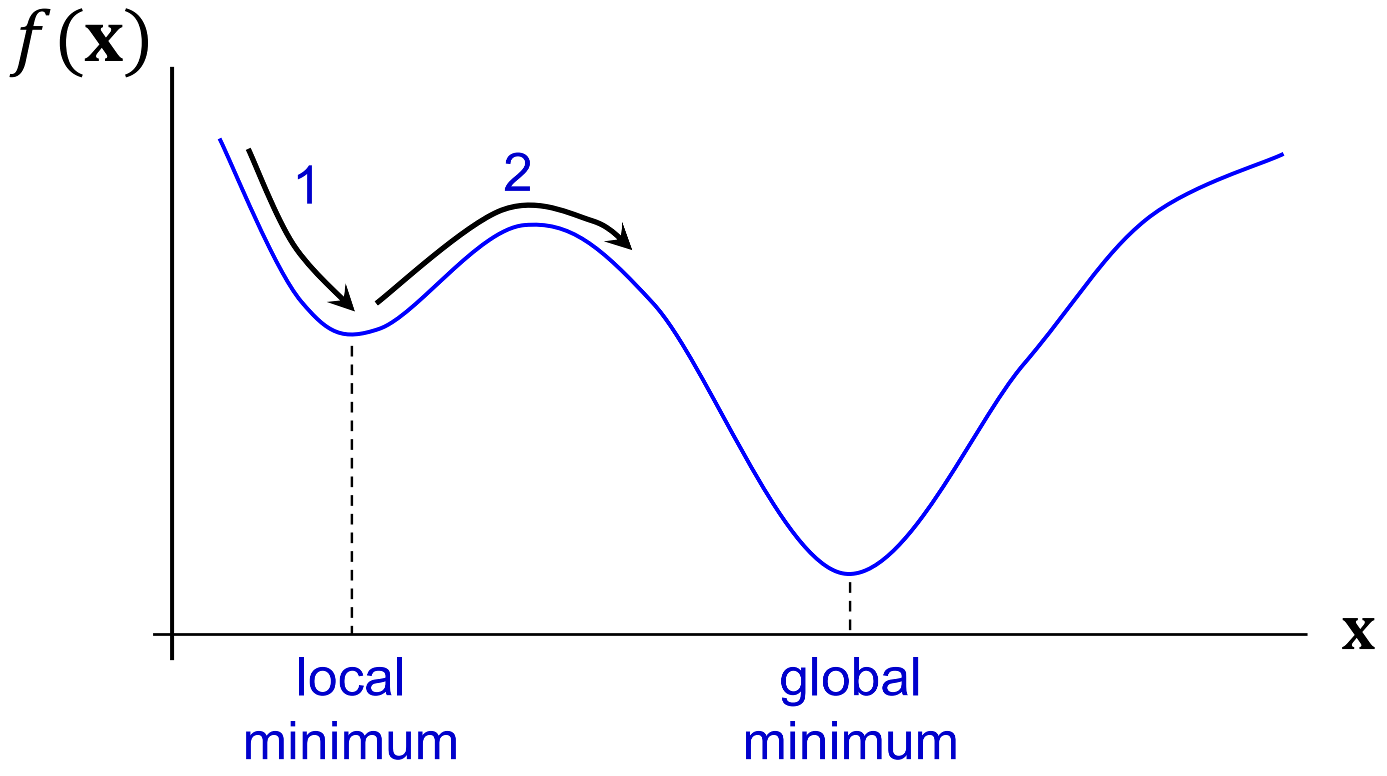


Key idea:

- Split region into sub-regions and compute bounds.
- Bound: Consider two regions A and C: if lower bound of A is greater than upper bound of C then A can be discarded.
- Branch: Divide regions and repeat.

Simulated Annealing

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



- 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 (i.e. improves on the minimum), then accept it.
  - If the move increases the cost by ΔE:
    - Accept it with a probability $\sim \exp \frac{-\Delta E}{T}$.
    - 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 $\exp\frac{-\Delta E}{T}$ is approx. 1, and all moves are accepted.

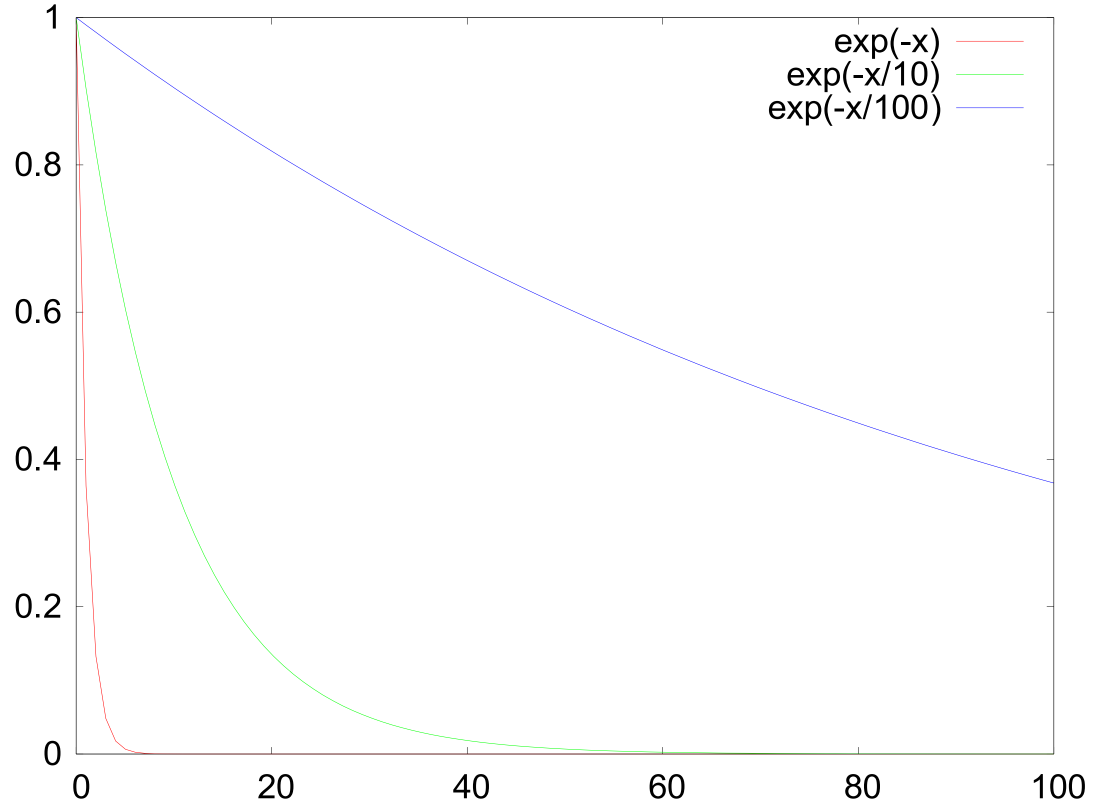
Many cooling schedules are possible, but the simplest is:

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

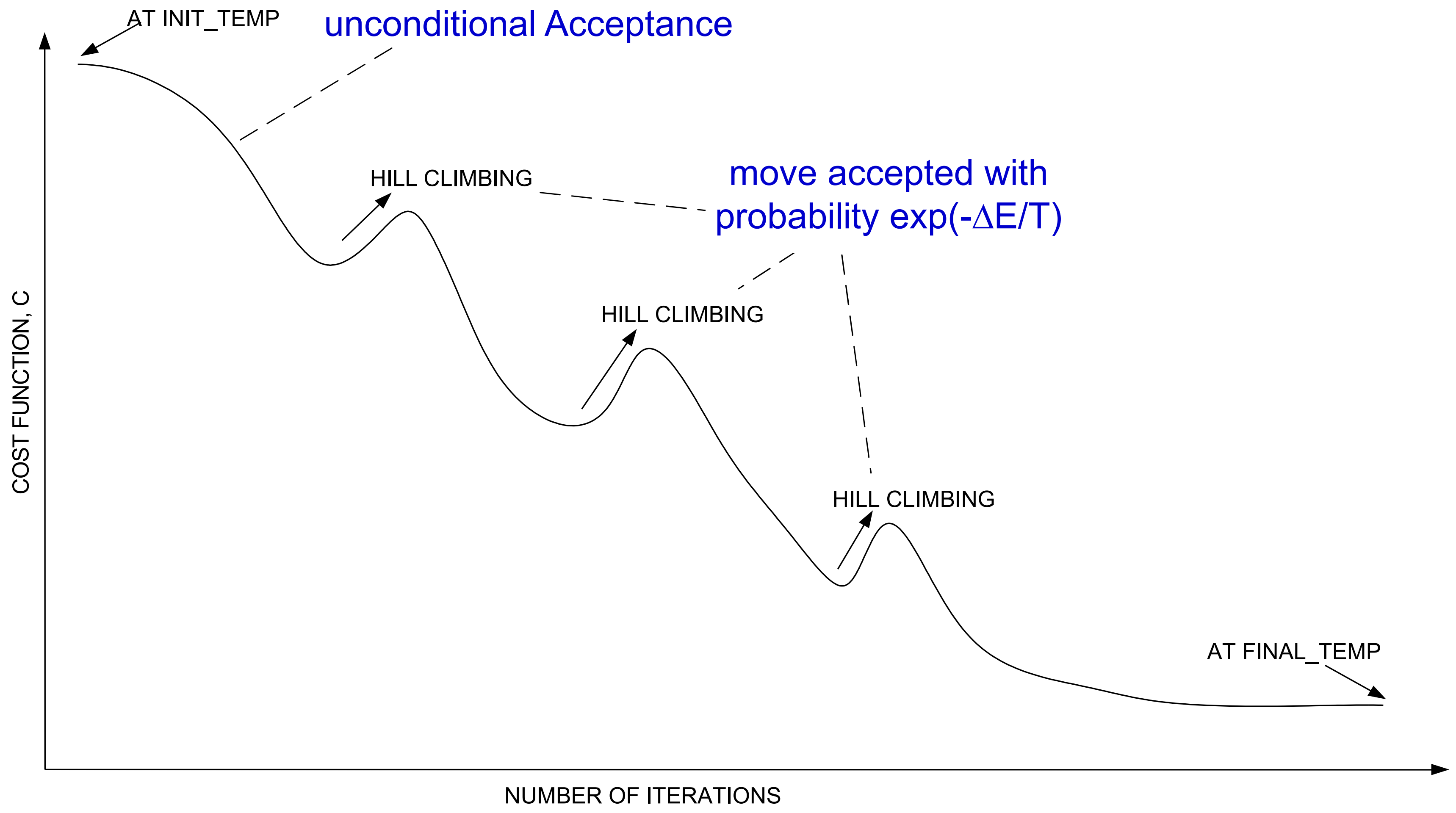
where k is the iteration number.

The algorithm can be very slow to converge …

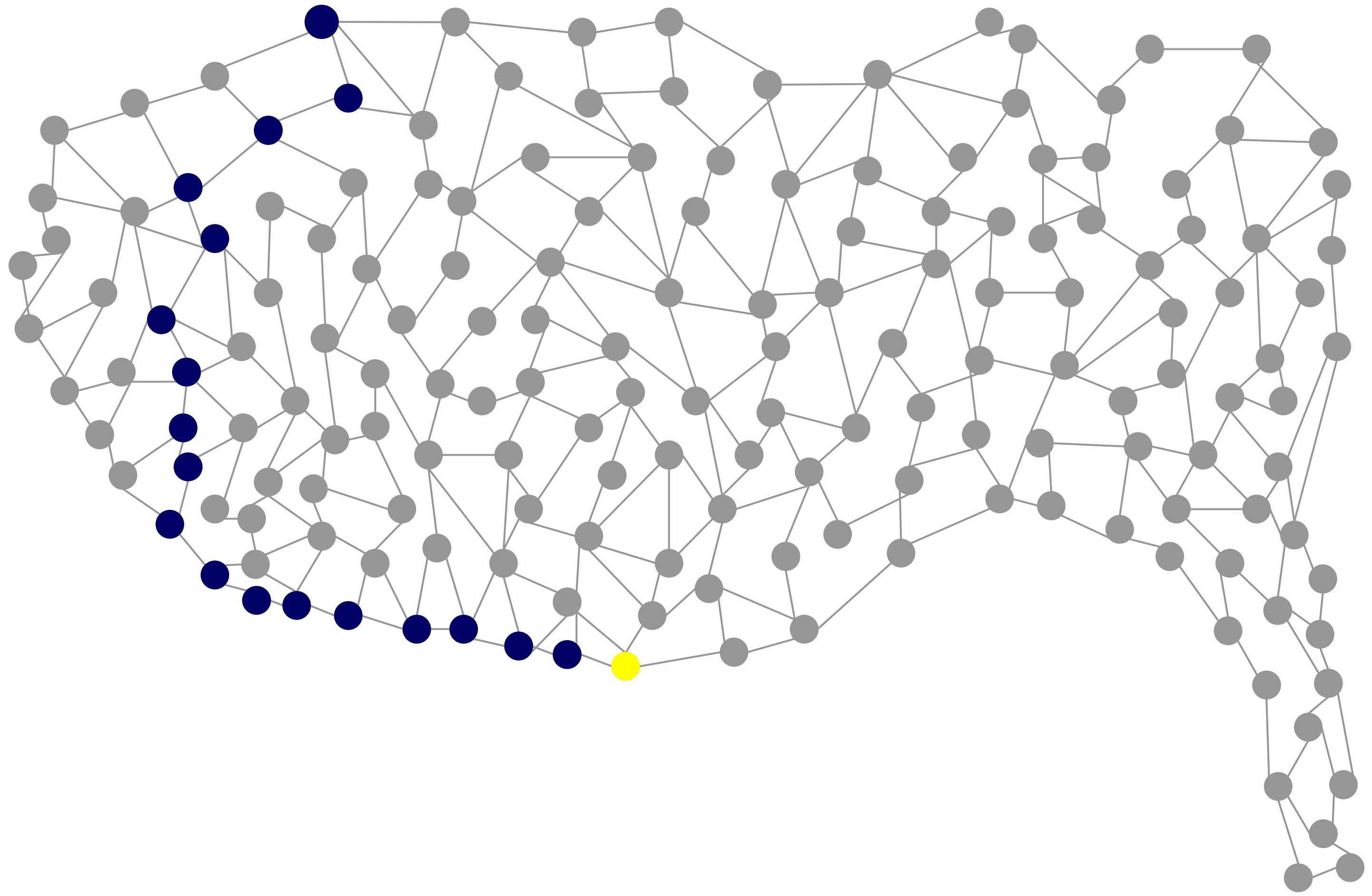
Boltzmann distribution $\exp\frac{-\Delta E}{T}$



Example: Convergence of Simulated Annealing

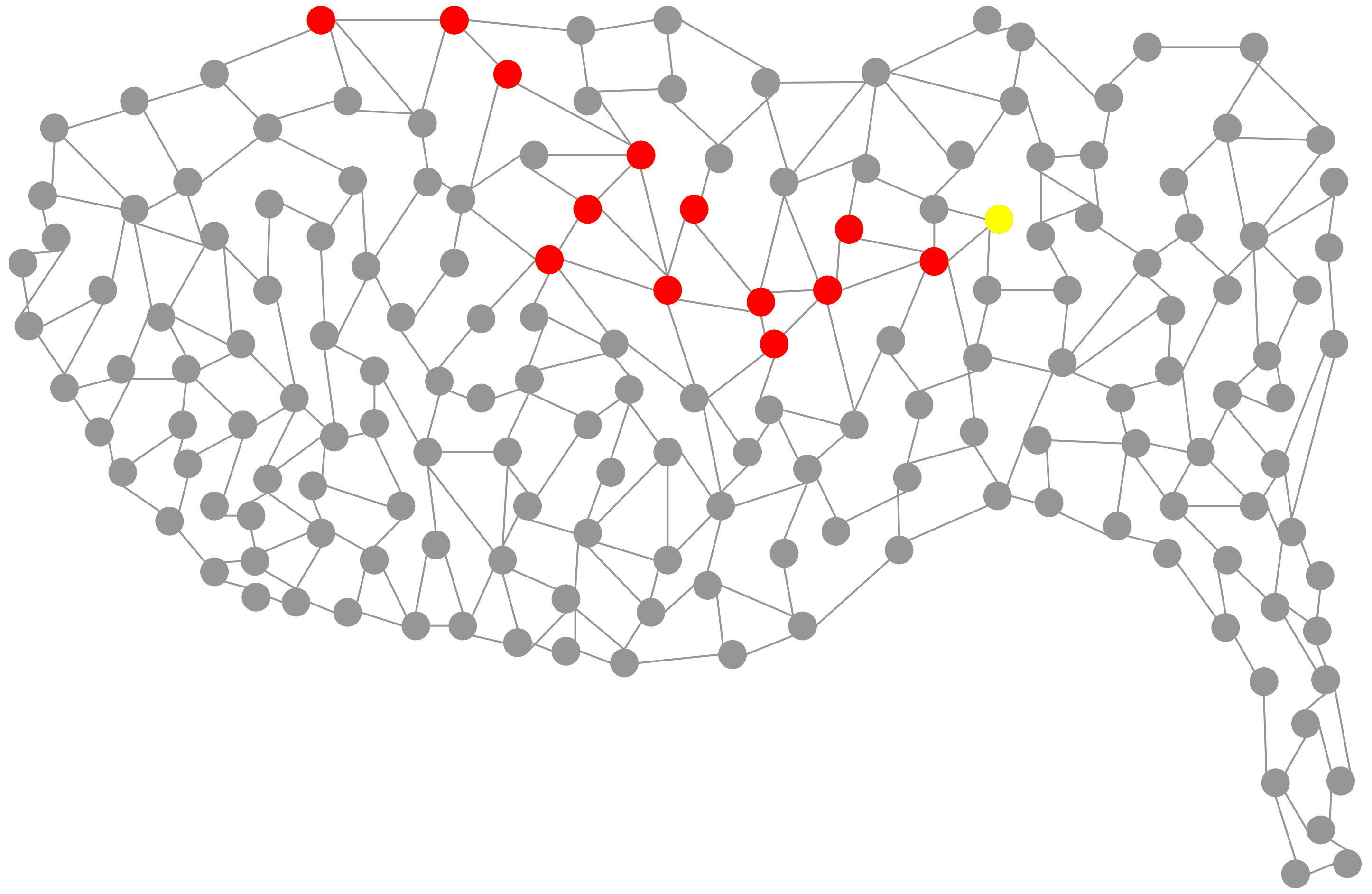


Steepest Descent on a Graph



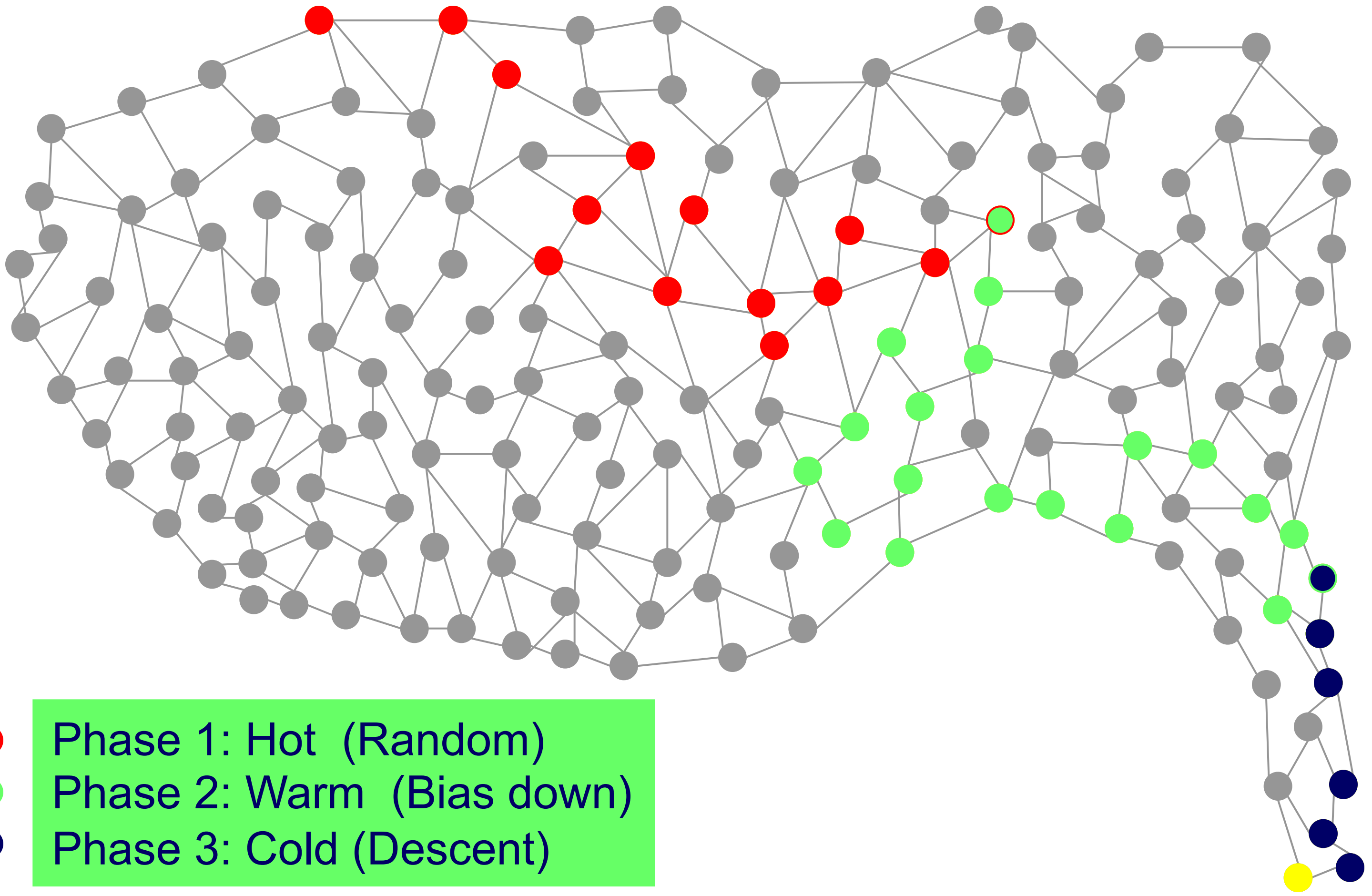
Slide from Adam Kalai and Santosh Vempala

Random Search on a Graph



Slide from Adam Kalai and Santosh Vempala

Simulated Annealing on a Graph



● Phase 1: Hot (Random)
● Phase 2: Warm (Bias down)
● Phase 3: Cold (Descent)

Slide from Adam Kalai and Santosh Vempala

Simulated Annealing for the Rosenbrock function

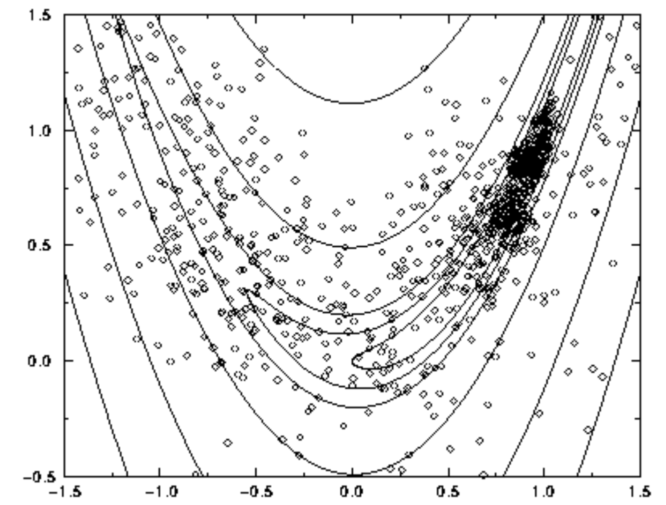


Figure 18: Minimization of the two-dimensional Rosenbrock function by simulated annealing--search pattern.

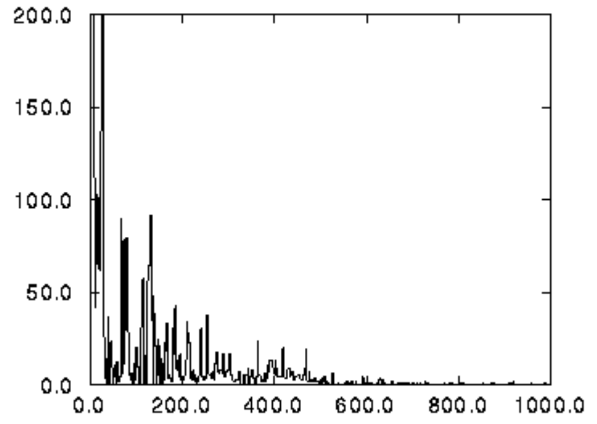


Figure 19: Minimization of the two-dimensional Rosenbrock function by simulated annealing objective reduction.

- Figure shows accepted solutions over a 1000 trial search.
- The search is wide-ranging but ultimately concentrates in the neighborhood of the optimum.
  - Initially, when the annealing temperature is high, some large increases in **f** are accepted and some areas far from the optimum are explored.
  - As execution continues and **T** falls, fewer uphill excursions are tolerated (and those that are tolerated are of smaller magnitude).
  - The last 40% of the run is spent searching around the optimum.
- This performance is typical of the SA algorithm.

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

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.

Evolutionary Optimization

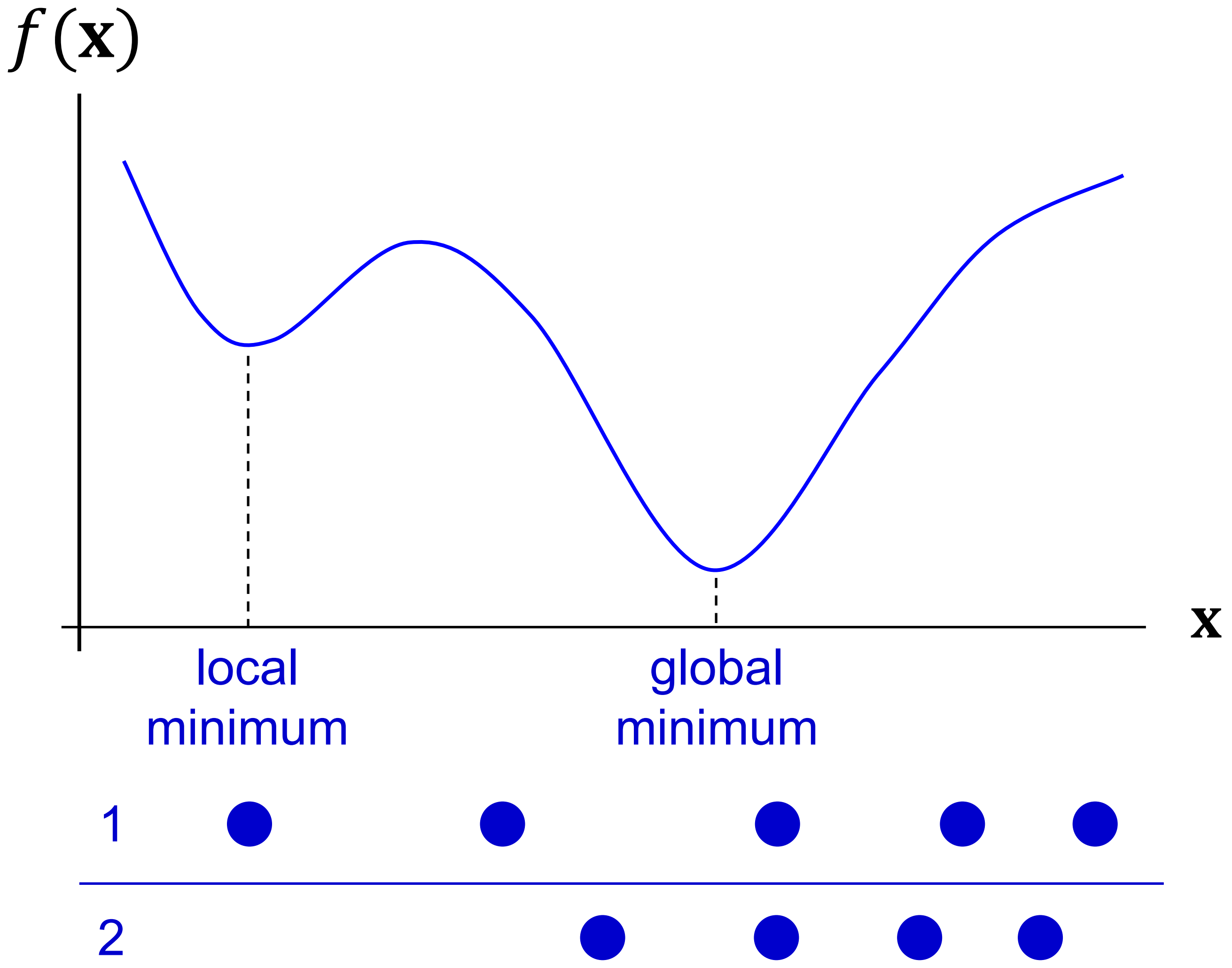
Uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the cost function determines the quality of the solutions. Evolution of the population then takes place after repeated application of the above operators.

Algorithm:

1. Generate the initial population of individuals randomly.
2. Evaluate the fitness of each individual in that population (cost function)
3. Repeat the following regenerational steps until termination:
    - Select the best-fit individuals for reproduction. (Parents)
    - Breed new individuals through crossover and mutation operations to give birth to offspring.
    - Evaluate the individual fitness of new individuals.
    - Replace least-fit population with new individuals.

(from Wikipedia)

Example



Genetic Algorithm for the Rosenbrock function

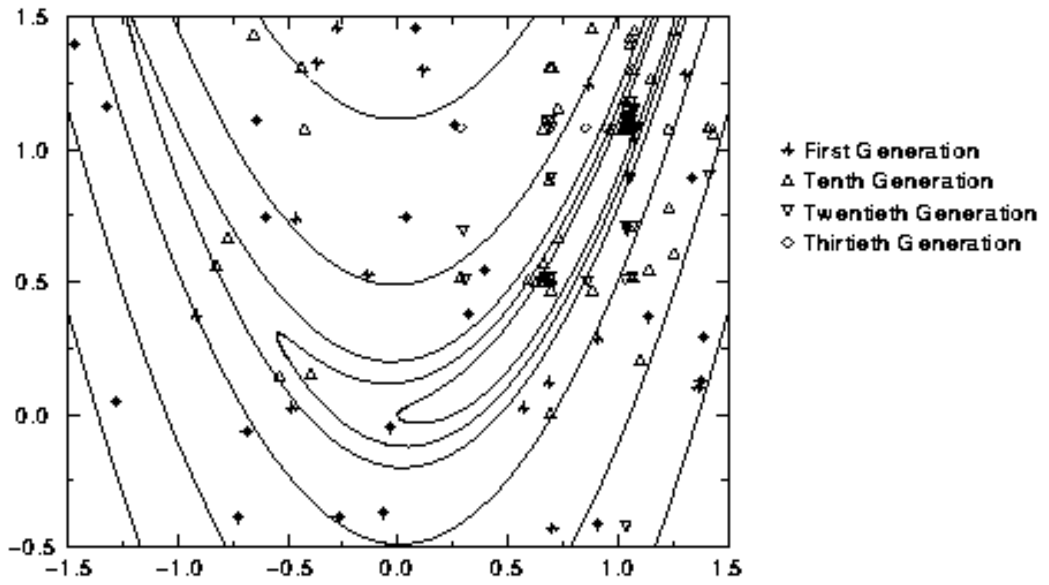


Figure 21: Minimization of two-dimensional Rosenbrock function by a genetic algorithm---population distributions of the first, tenth, twentieth, and thirtieth generations.

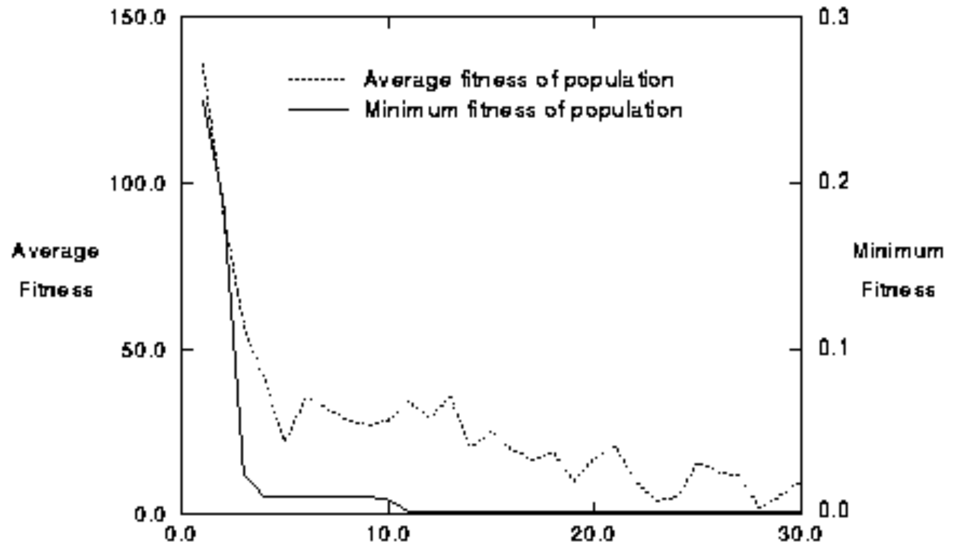


Figure 22: Minimization of the two-dimensional Rosenbrock function by a genetic algorithm---population distribution of the first, tenth, twentieth, and thirtieth generations.

Figures from https://www.phy.ornl.gov/csep/mo/node44.html

Example: Evolving Virtual Creatures

Evolved Virtual Creatures

Karl Sims

Karl Sims, 1994

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 books

Go to the C Optimization lectures next year (AZ)