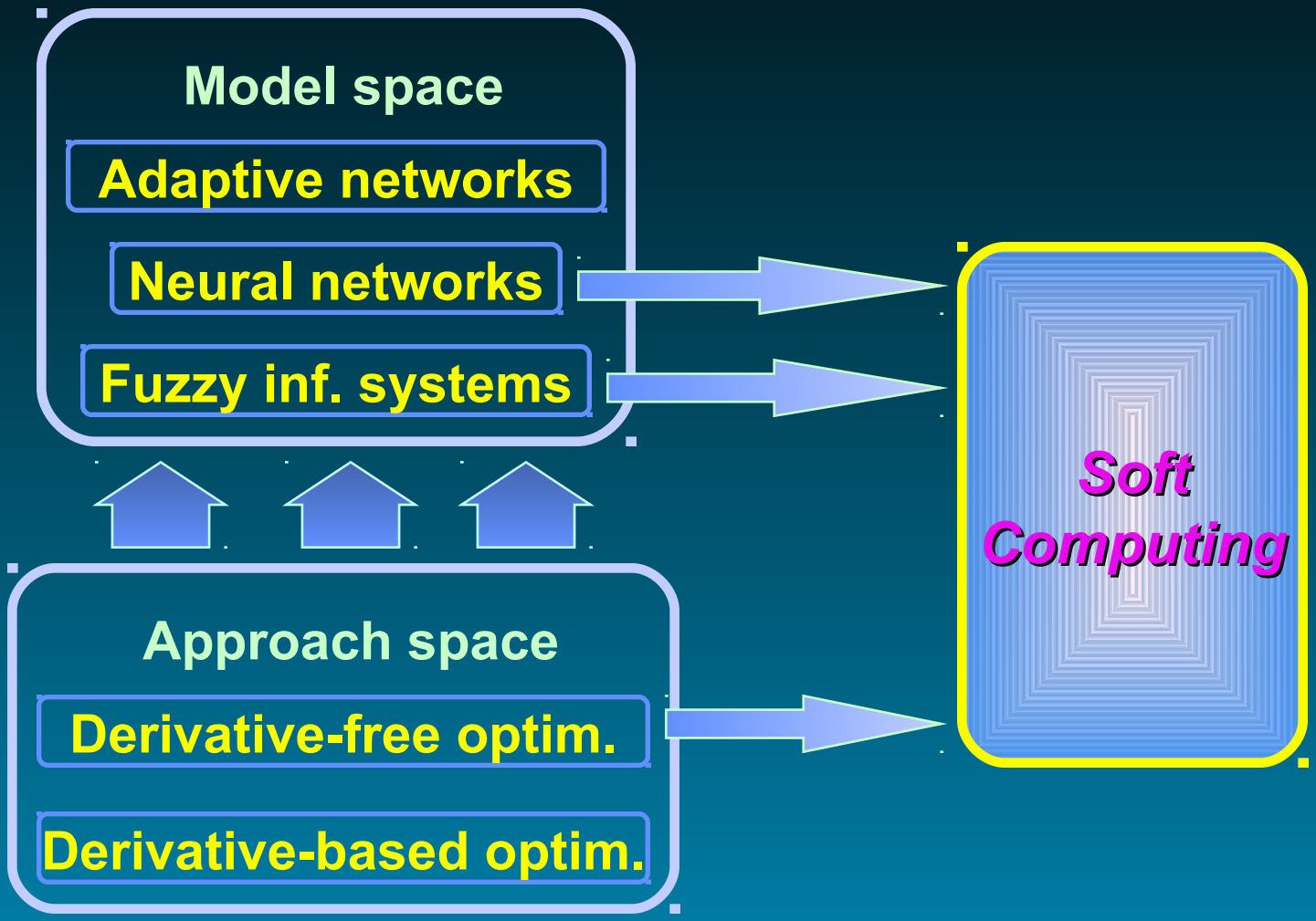
Derivative-Free Optimization

J.-S. Roger Jang (張智星)
CS Dept., Tsing Hua Univ., Taiwan
http://www.cs.nthu.edu.tw/~jang
jang@cs.nthu.edu.tw

Outline

- Genetic algorithms (GA)
- Simulated Annealing (SA)
- Downhill simplex search
- Random search

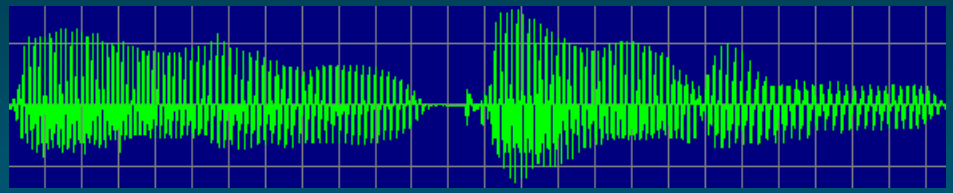
The Big Picture



Genetic Algorithms

Motivation

- **Look at what evolution brings us?**
  - Vision
  - Hearing
  - Smelling
  - Taste
  - Touch
  - Learning and reasoning



- **Can we emulate the evolutionary process with today's fast computers?**

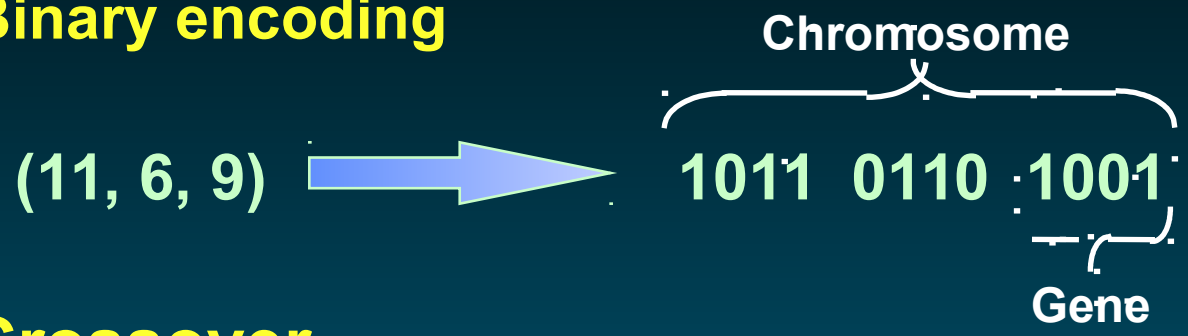
Genetic Algorithms

Terminology:

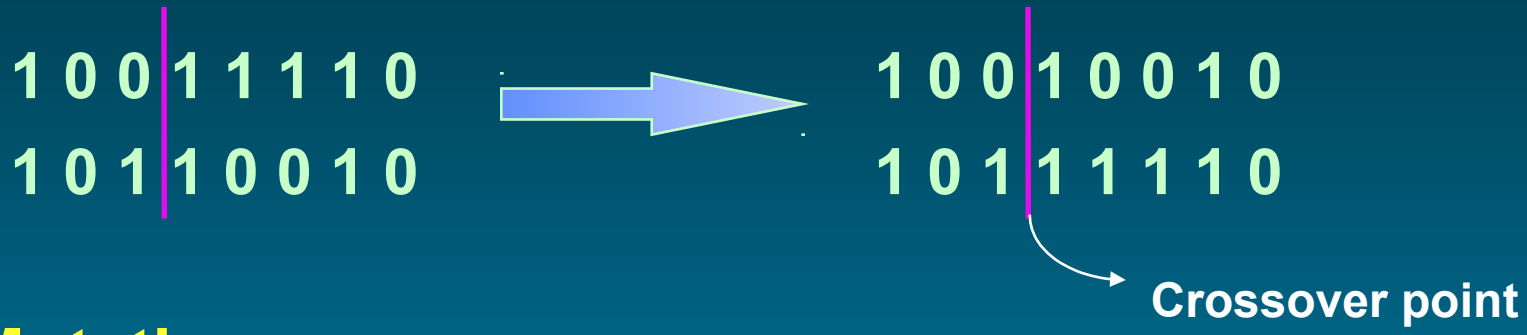
- Fitness function
- Population
- Encoding schemes
- Selection
- Crossover
- Mutation
- Elitism

Genetic Algorithms

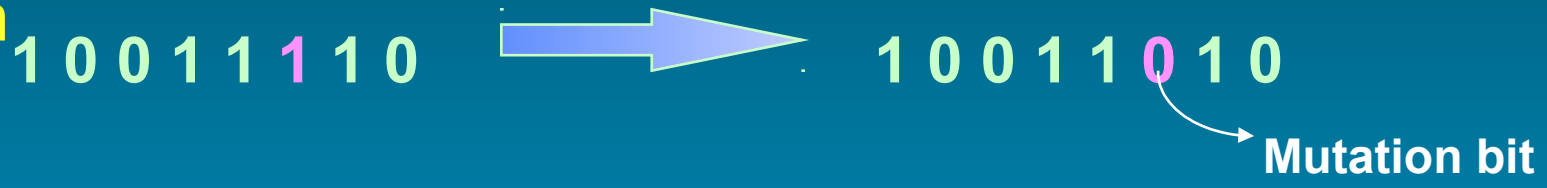
Binary encoding



Crossover

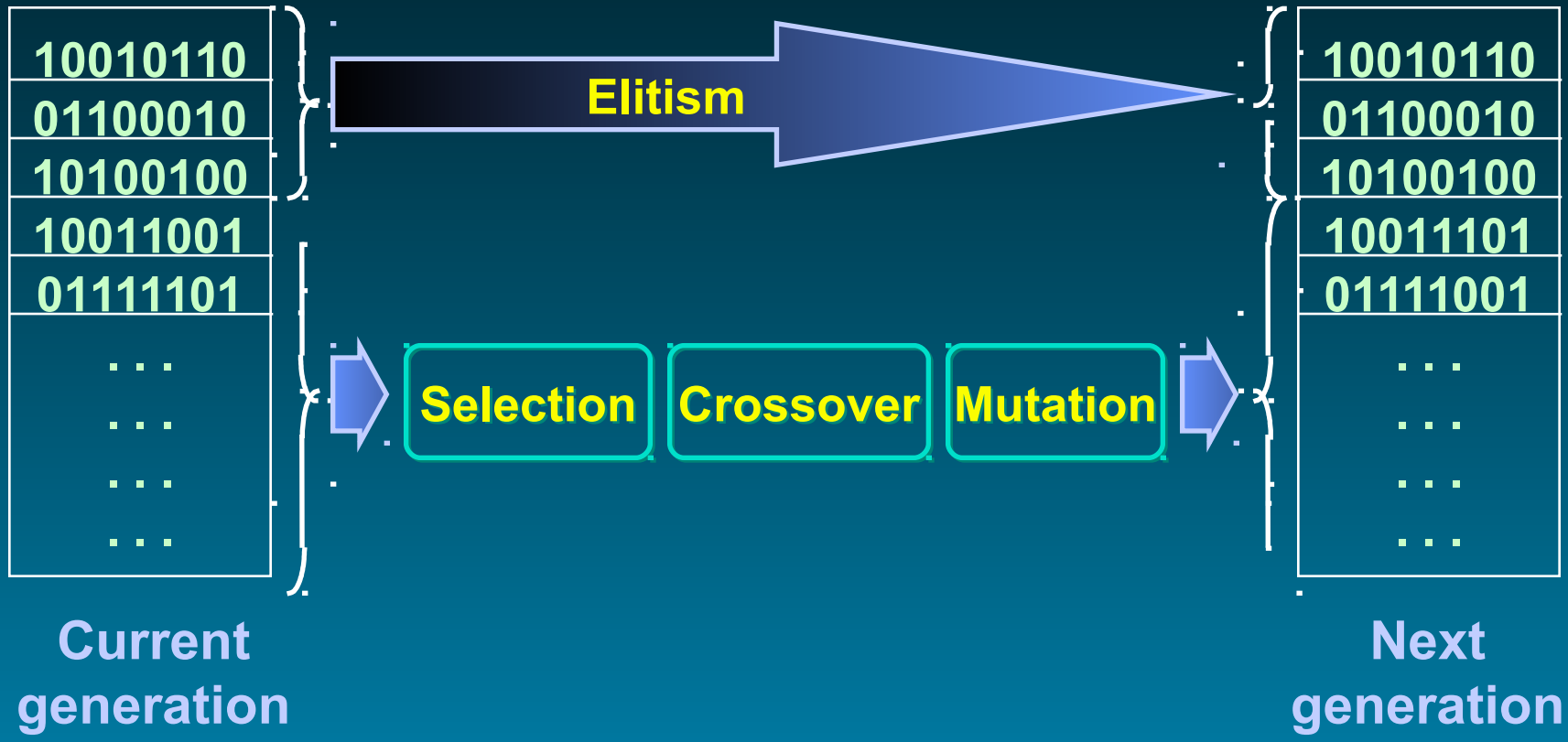


Mutation



Genetic Algorithms

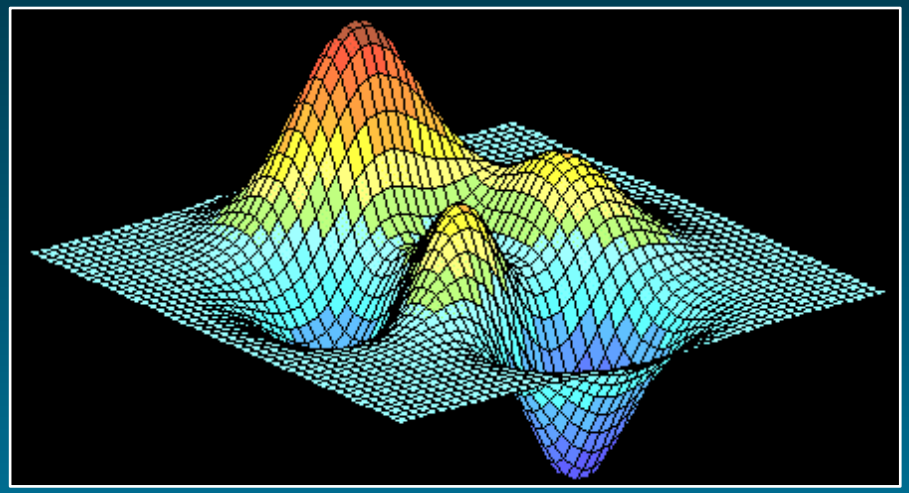
Flowchart

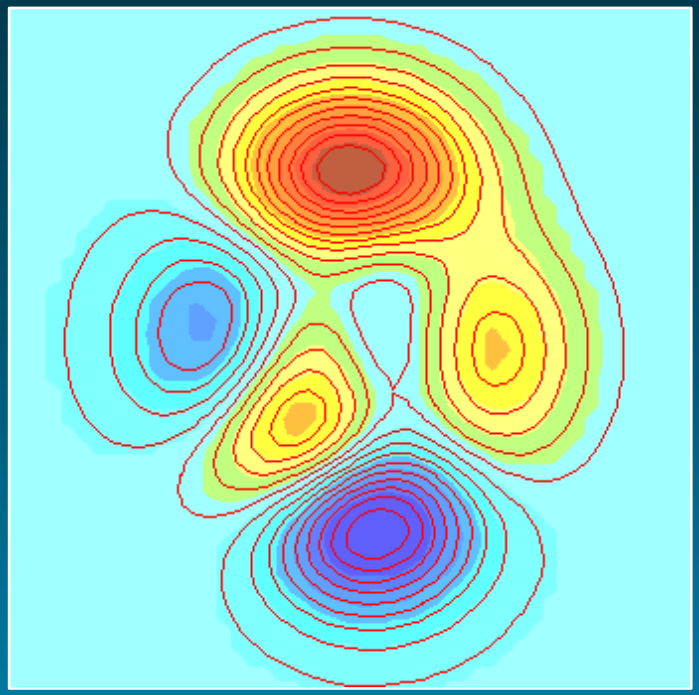


Genetic Algorithms

Example: Find the max. of the "peaks" function

z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).





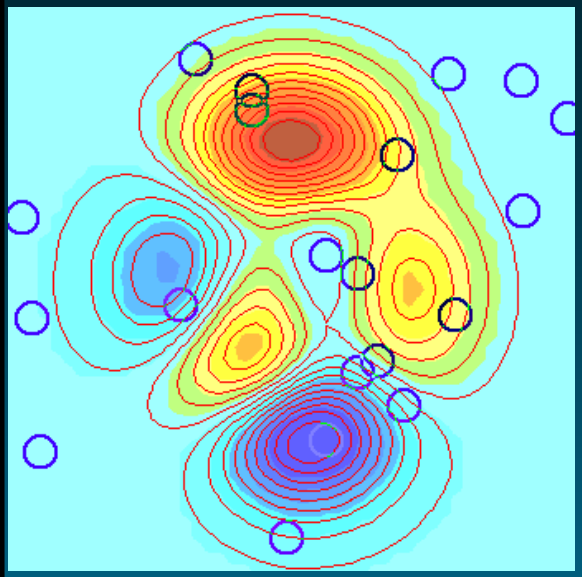
Genetic Algorithms

Derivatives of the "peaks" function

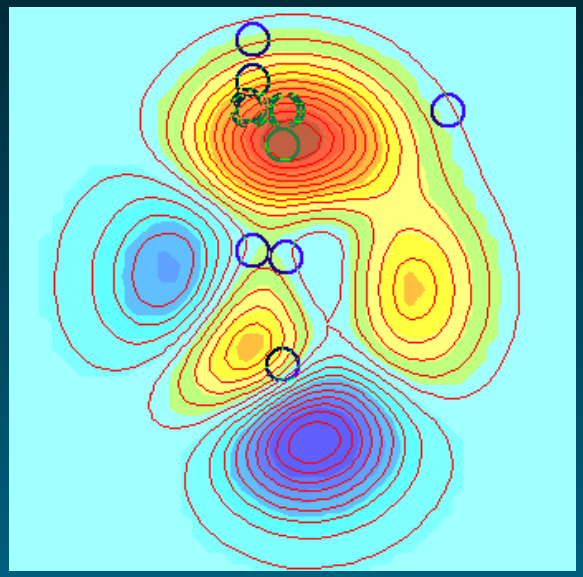
- dz/dx = -6*(1-x)*exp(-x^2-(y+1)^2) - 6*(1-x)^2*x*exp(-x^2-(y+1)^2) - 10*(1/5-3*x^2)*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*x*exp(-x^2-y^2) - 1/3*(-2*x-2)*exp(-(x+1)^2-y^2)
- dz/dy = 3*(1-x)^2*(-2*y-2)*exp(-x^2-(y+1)^2) + 50*y^4*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*y*exp(-x^2-y^2) + 2/3*y*exp(-(x+1)^2-y^2)
- d(dz/dx)/dx = 36*x*exp(-x^2-(y+1)^2) - 18*x^2*exp(-x^2-(y+1)^2) - 24*x^3*exp(-x^2-(y+1)^2) + 12*x^4*exp(-x^2-(y+1)^2) + 72*x*exp(-x^2-y^2) - 148*x^3*exp(-x^2-y^2) - 20*y^5*exp(-x^2-y^2) + 40*x^5*exp(-x^2-y^2) + 40*x^2*exp(-x^2-y^2)*y^5 -2/3*exp(-(x+1)^2-y^2) - 4/3*exp(-(x+1)^2-y^2)*x^2 -8/3*exp(-(x+1)^2-y^2)*x
- d(dz/dy)/dy = -6*(1-x)^2*exp(-x^2-(y+1)^2) + 3*(1-x)^2*(-2*y-2)^2*exp(-x^2-(y+1)^2) + 200*y^3*exp(-x^2-y^2)-200*y^5*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*exp(-x^2-y^2) - 40*(1/5*x-x^3-y^5)*y^2*exp(-x^2-y^2) + 2/3*exp(-(x+1)^2-y^2)-4/3*y^2*exp(-(x+1)^2-y^2)

Genetic Algorithms

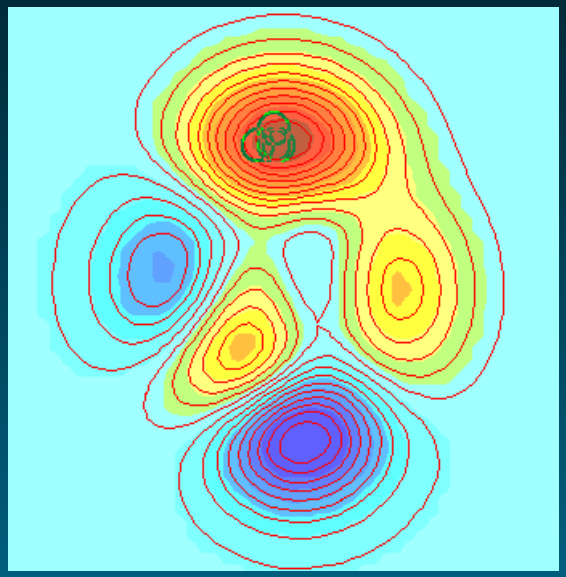
GA process:



Initial population



5th generation

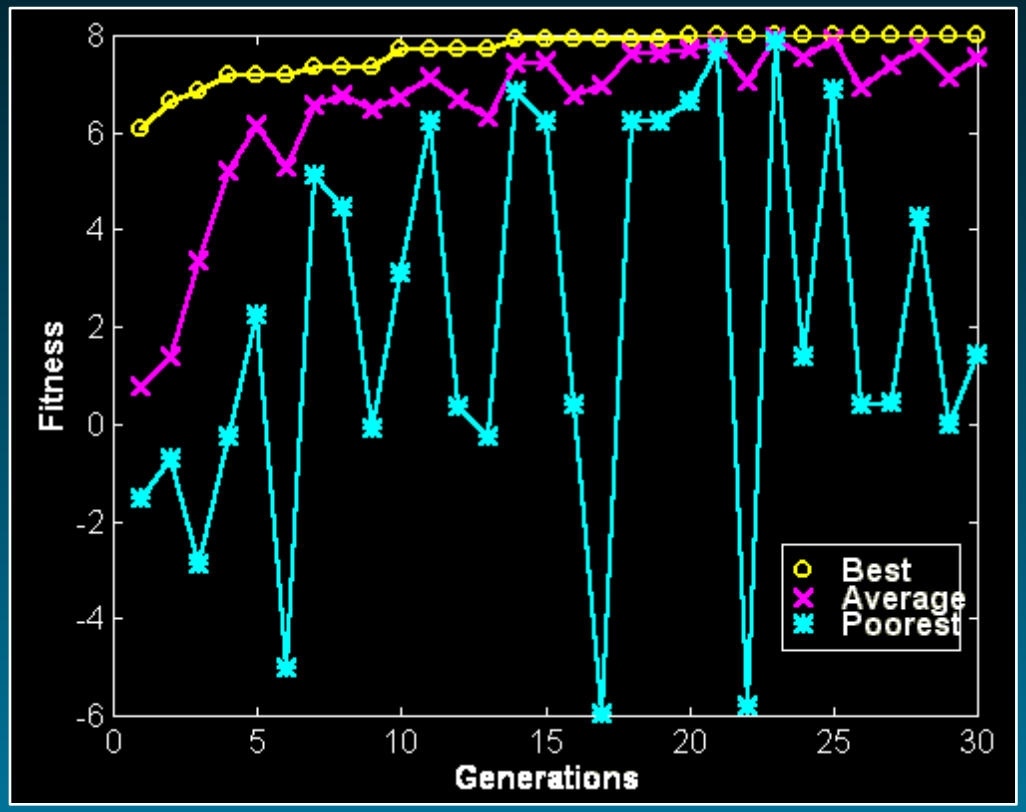


10th generation

MATLAB file: go_ga.m

Genetic Algorithms

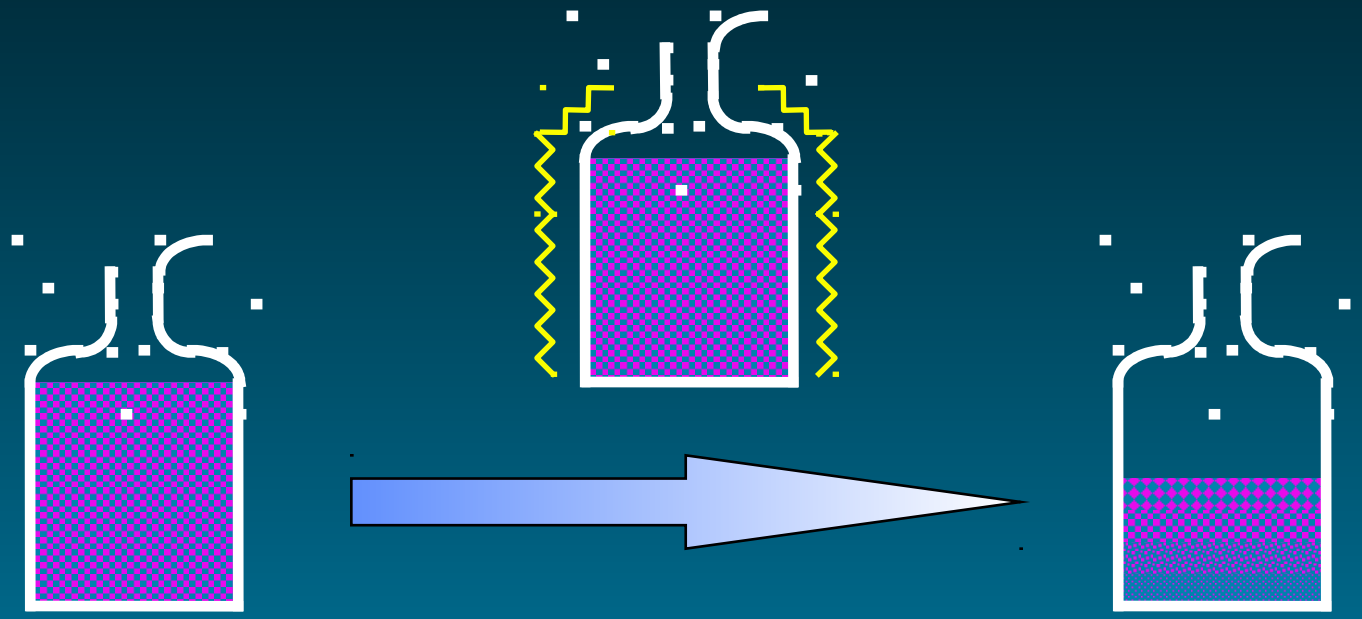
Performance profile



MATLAB file: go_ga.m

Simulated Annealing

Analogy



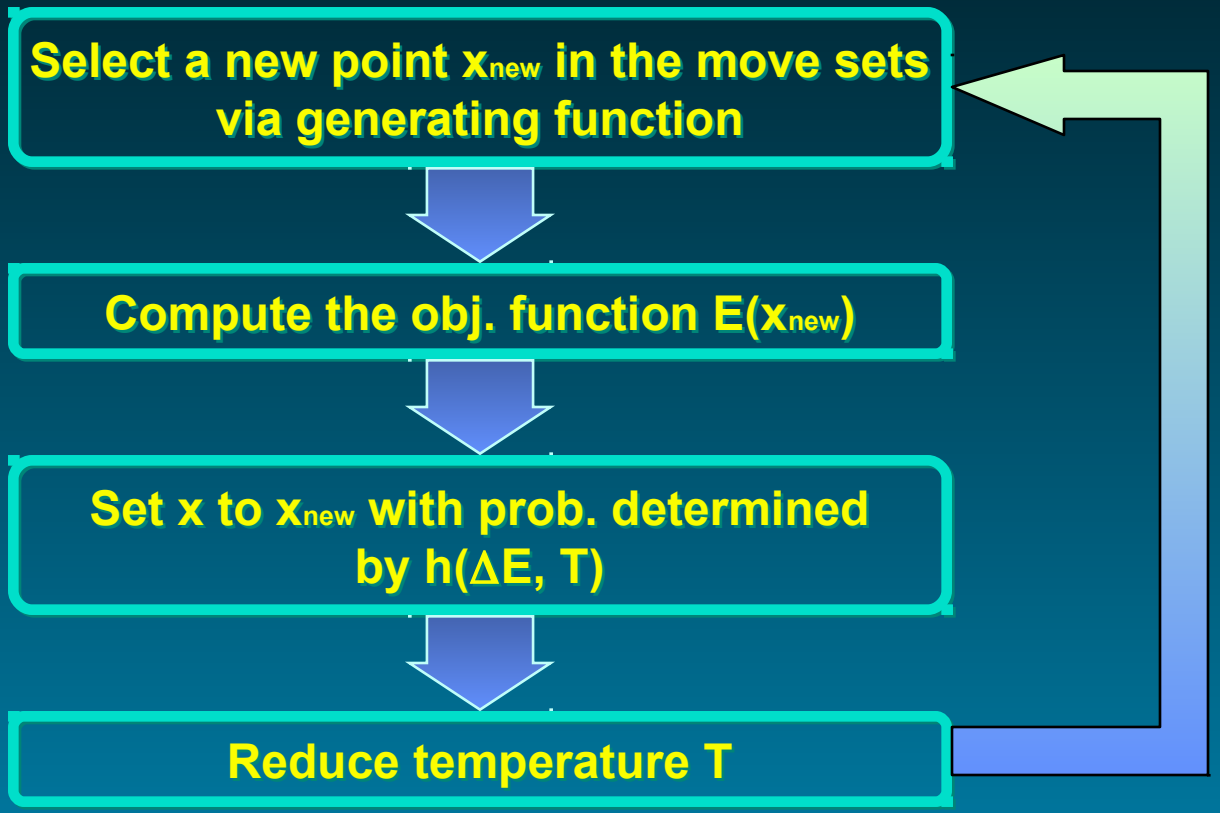
Simulated Annealing

Terminology:

- Objective function E(x): function to be optiimized
- Move set: set of next points to explore
- Generating function: to select next point
- Acceptance function h(ΔE, T): to determine if the selected point should be accept or not. Usually h(ΔE, T) = 1/(1+exp(ΔE/(cT)).
- Annealing (cooling) schedule: schedule for reducing the temperature T

Simulated Annealing

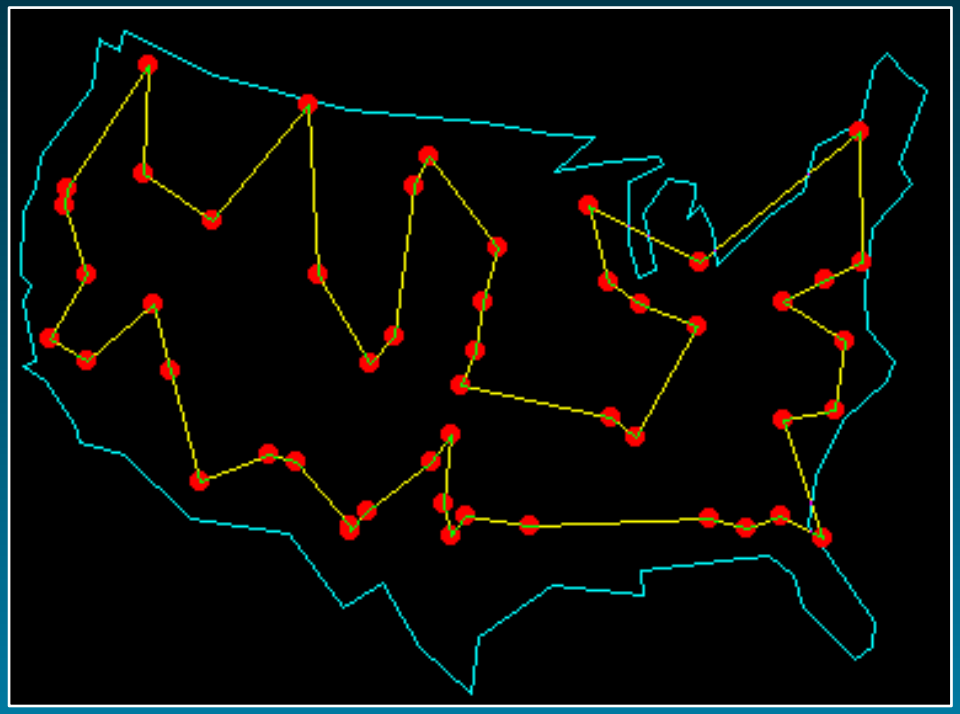
Flowchart:



Simulated Annealing

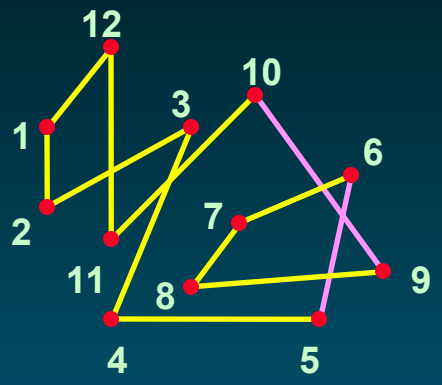
Example: Travel Salesperson Problem (TSP)

How to transverse n cities once and only once with a minimal total distance?



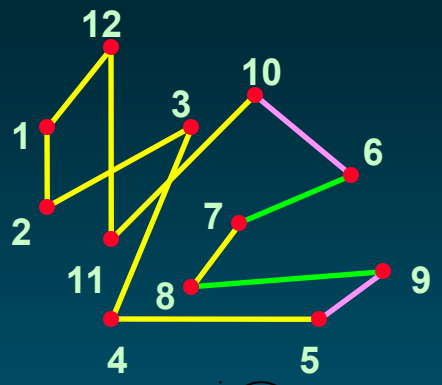
Simulated Annealing

Conjunto de movimentos para o Mascate



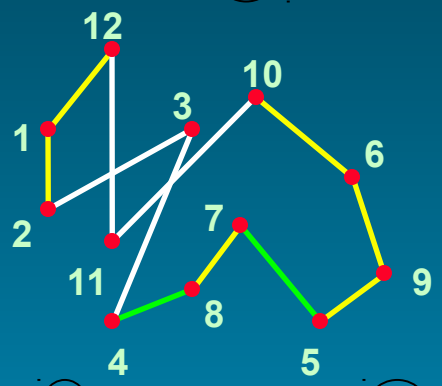
1-2-3-4-5-6-7-8-9-10-11-12

Inversão



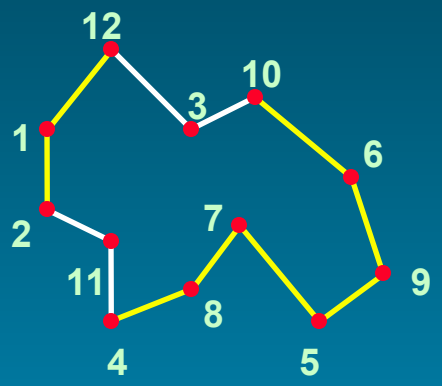
1-2-3-4-5-9-8-7-6-10-11-12

Translação



1-2-3-4-8-7-5-9-6-10-11-12

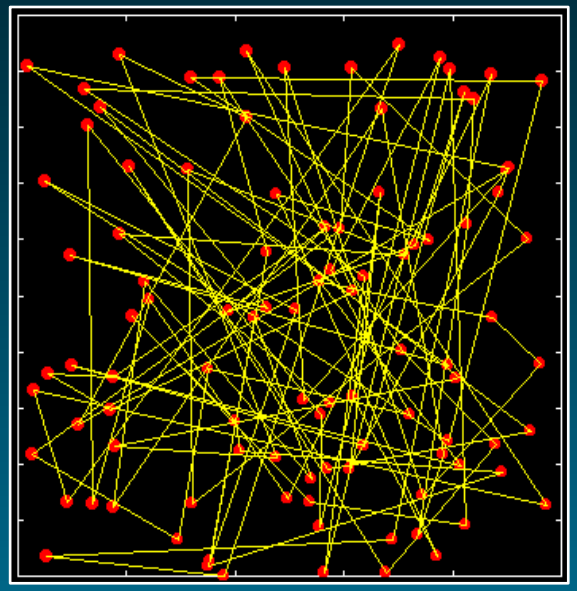
Troca



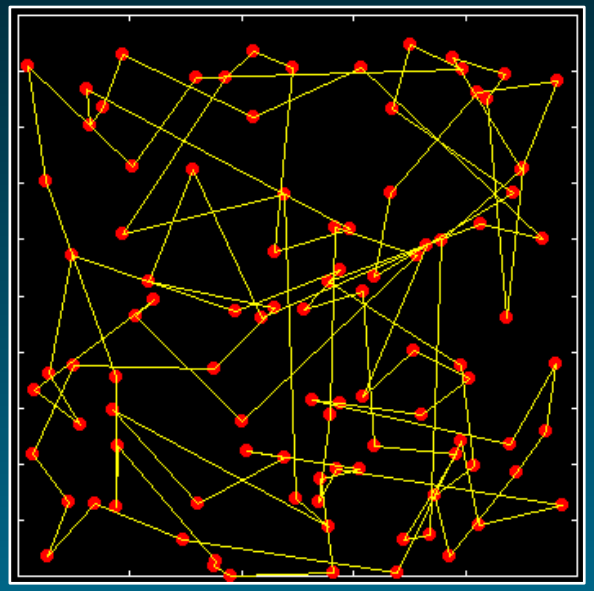
1-2-11-4-8-7-5-9-6-10-3-12

Simulated Annealing

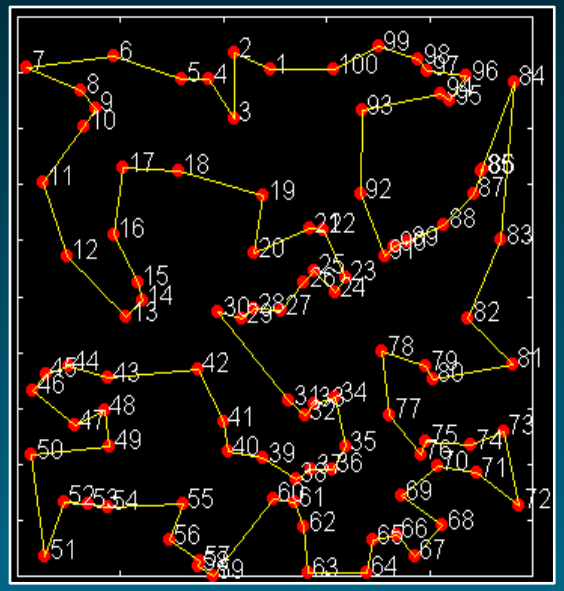
A 100-city TSP using SA



Initial random path



During SA process

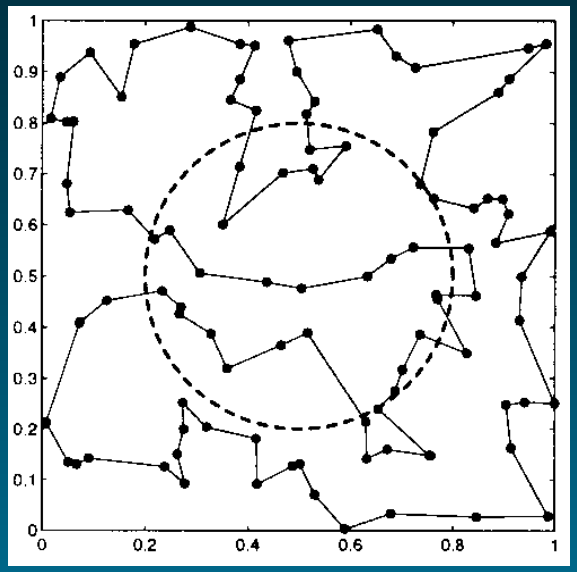


Final path

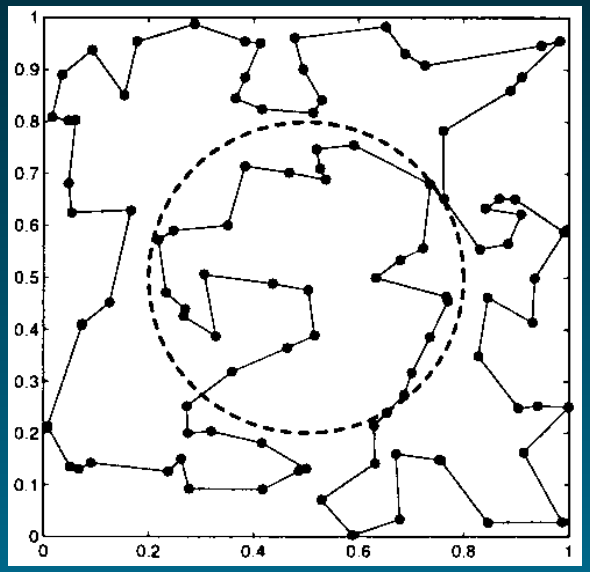
MATLAB file: tsp.m

Simulated Annealing

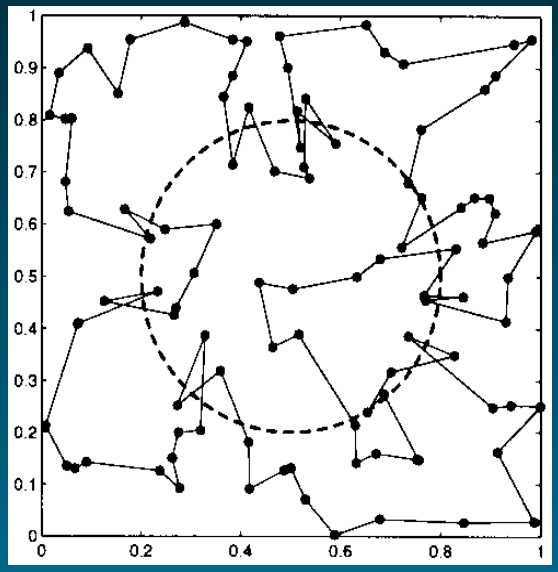
100-city TSP with penalities when crossing the circle



Penalty = 0



Penalty = 0.5



Penalty = -0.3

Random Search

Properties:

- Intuitive
- Simple

Analogy:

- Get down to a valley blindfolded

Two heuristics:

- Reverse step
- Bias direction

Downhill Simplex Search

Simplex: a set of n+1 points in n-dim. space

- A triangle in a 2D space
- A tetrahedron in a 3D space

Concept of downhill simplex search

- Repeatedly replaces the highest points with a lower one
- Consecutive successful replacements lead to the enlargement of the simplex
- Consecutive unsucessful replacements lead to the shrinkage of the simplex

Downhill Simplex Search

Flowchart

Figure 7.9 in page 188

Behavior

The simplex can adapt itself to the objective function landscape (just like an amoeba), and eventually converges to a nearby local minimum.

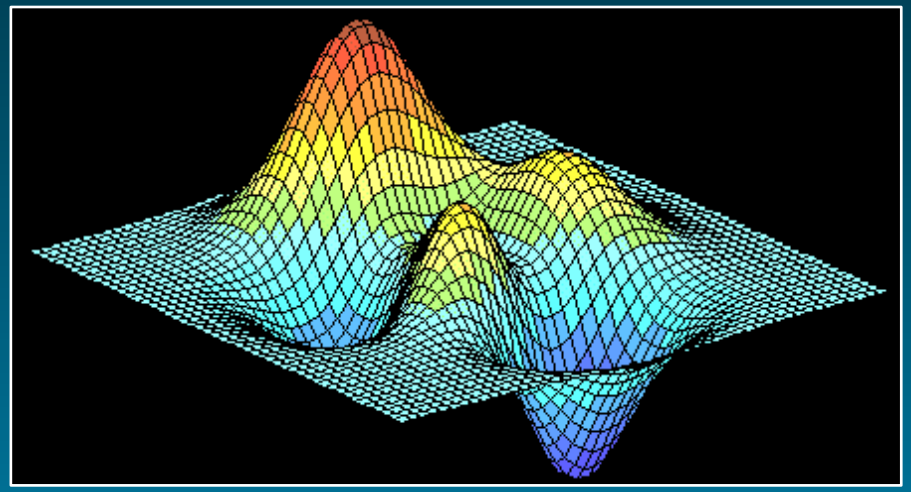
Program

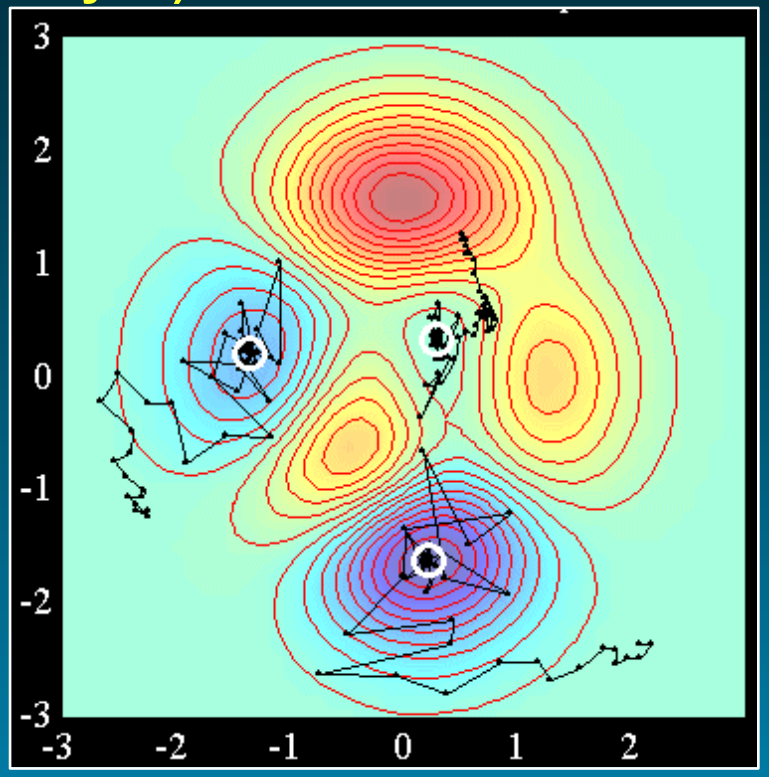
The search procedure is implemented as a function fmins.m that comes with MATLAB.

Downhill Simplex Search

Example: Find the min. of the "peaks" function

z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).

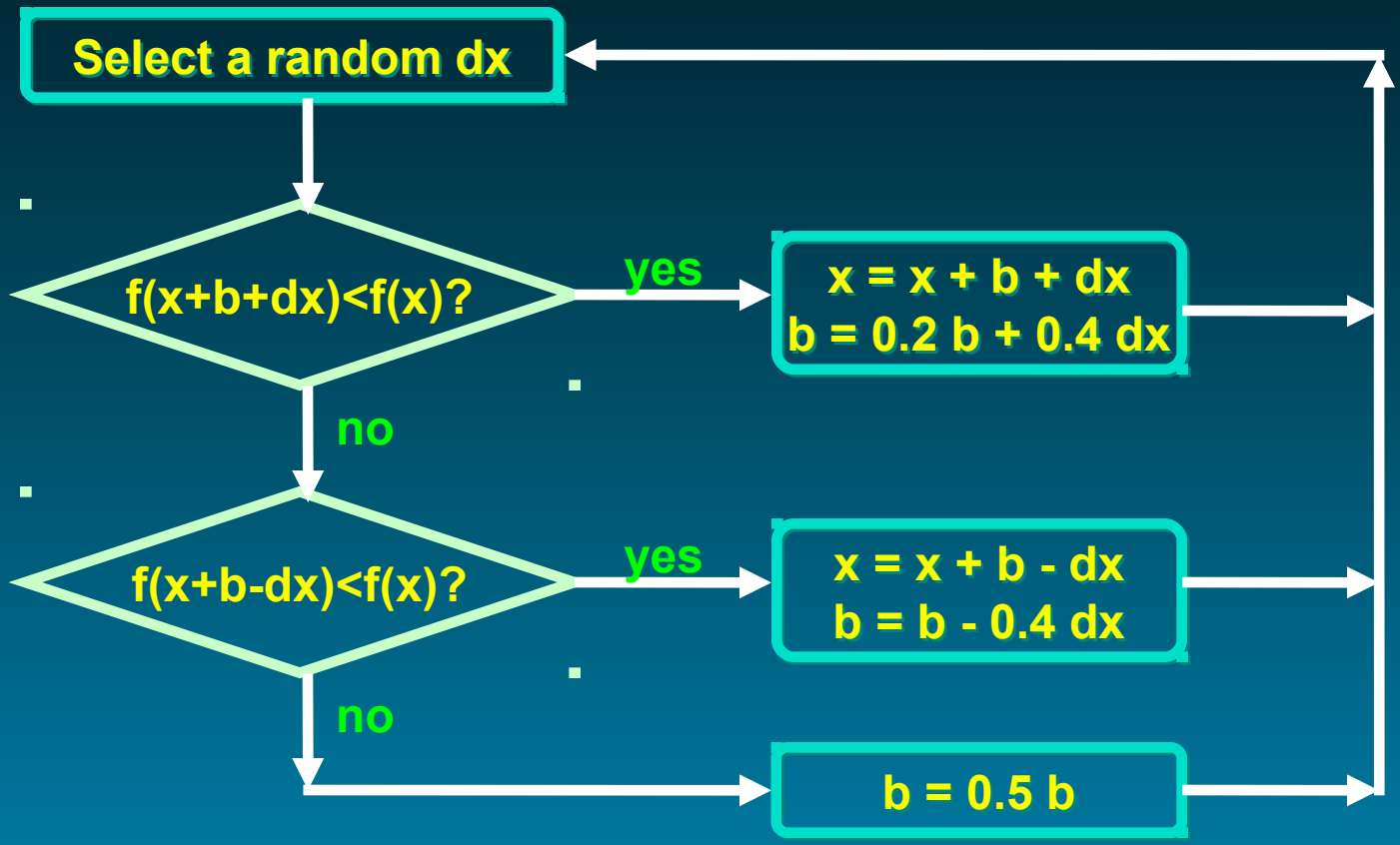




MATLAB file: go_simp.m

Random Search

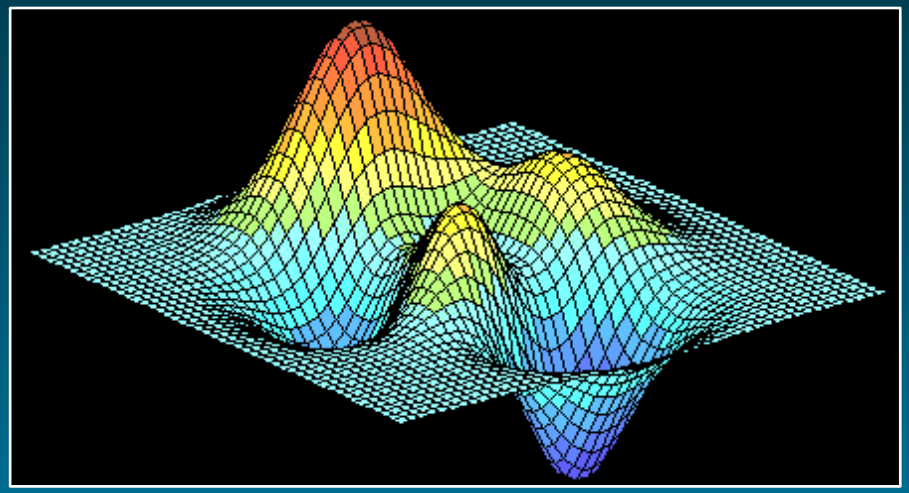
Flowchart:

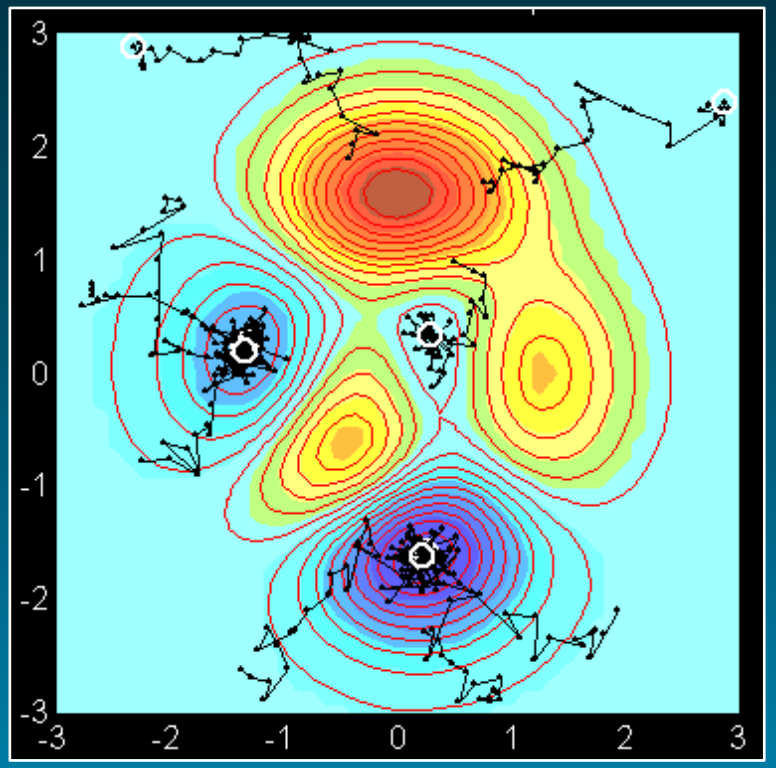


Random Search

Example: Find the min. of the "peaks" function

z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).





MATLAB file: go_rand.m