

Searching for a (good) solution



Main classes of problems

- ◆ optimization
 - ◆ knapsac
 - ◆ travel planning
- ◆ clusterization
 - ◆ web search engines (clustering & ordering)
- ◆ classification
 - ◆ voice/face recognition
 - ◆ word best match: soundness, ortographic correction

All is fitting (function)

It's possible to modelize this kind of problems as a minimization/maximisation problem.

- ◆ Given a set of possible solutions (search space) find the best solution that maximize/minimize a "energy"/"cost" function.
 - ◆ given a set of bags, find the best arrangement in order to minimize the occupied space.
 - ◆ given a set of apples of various size and a price table that define the price respect of the minimal size, find the best size that maximize the earning.

Optimal Solution vs Approximation

For almost all "interesting" problems, it is "hard" to find the best solution.



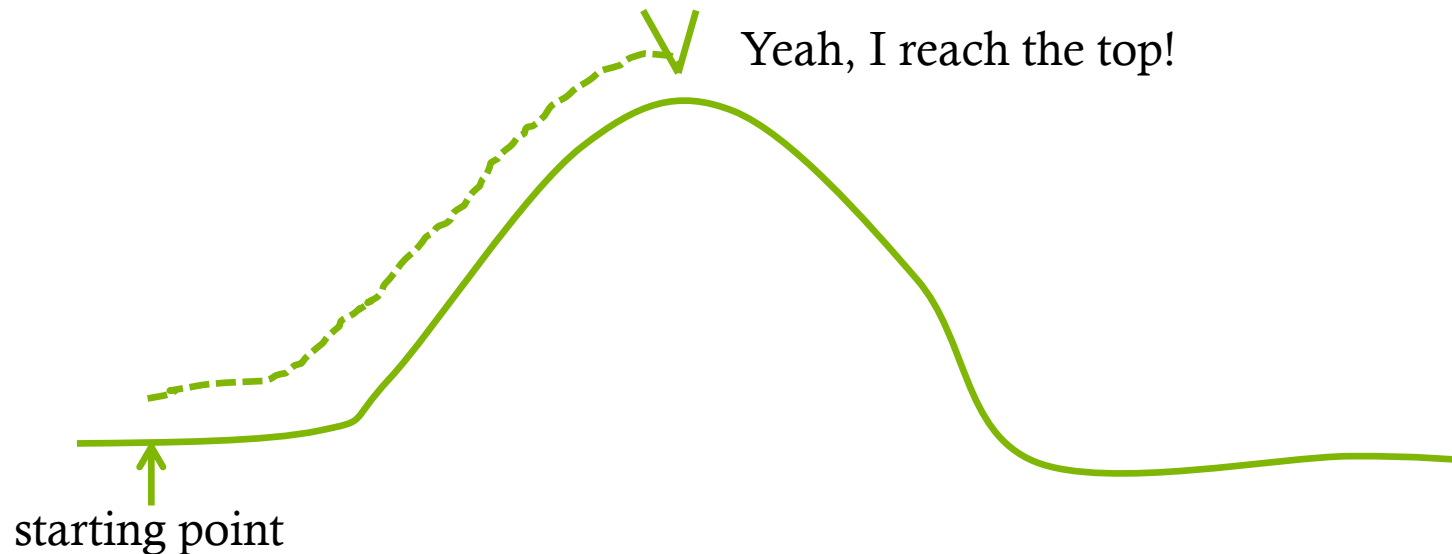
We have to find something similar (near) to the best.



We need a search path for a good approximation

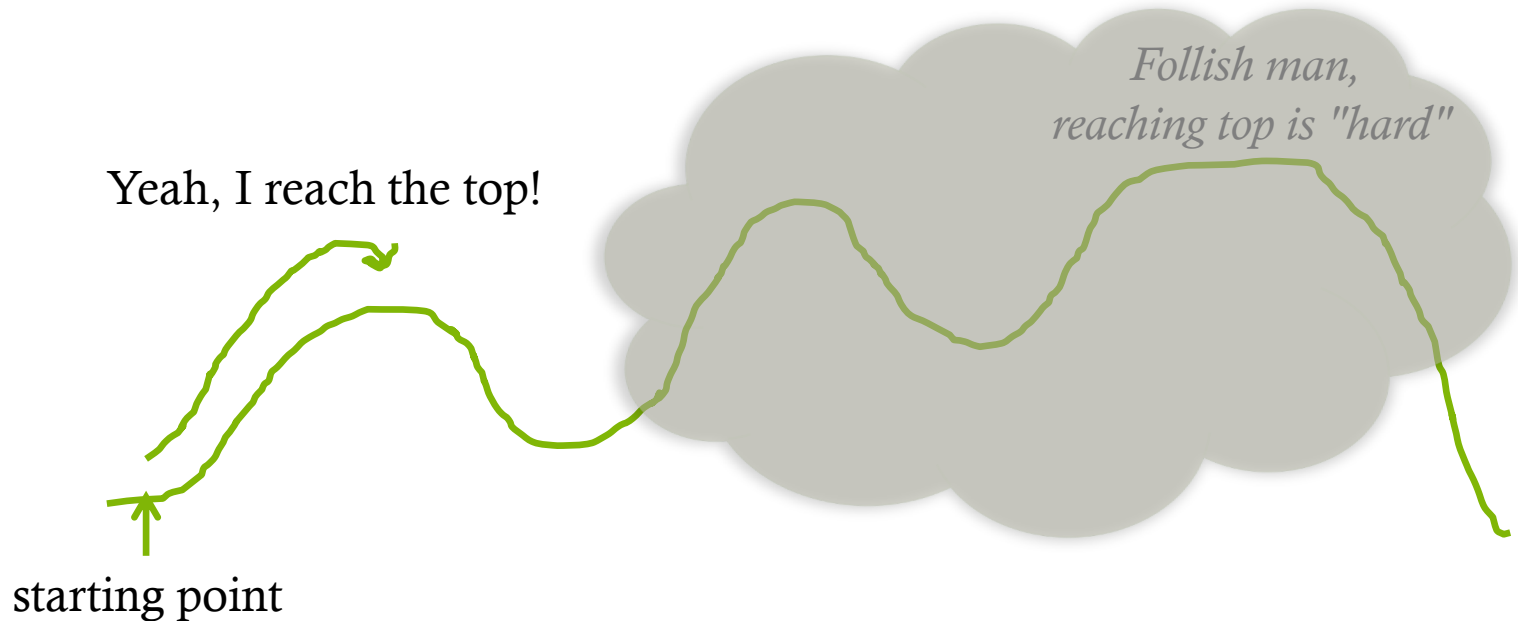
Greedy Methods

- ◆ Move myself toward the best I can see (or I can guess)



The local-max problem

💧 ...may be I cant see the best...

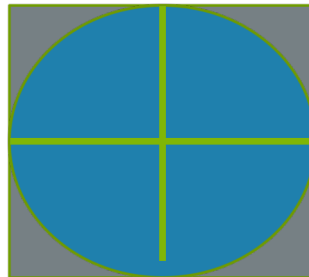


Finding (for accident) the best

Sometime randomness can help to find the solution

- ◆ A (not so) related example:
 - ◆ The Montecarlo method for calculates π : I uniform randomly take a set of point inside a square containing a circle, if it is inside (easy to calculates) I count it on the area of the circle and the square, otherwise I count it only for the square.

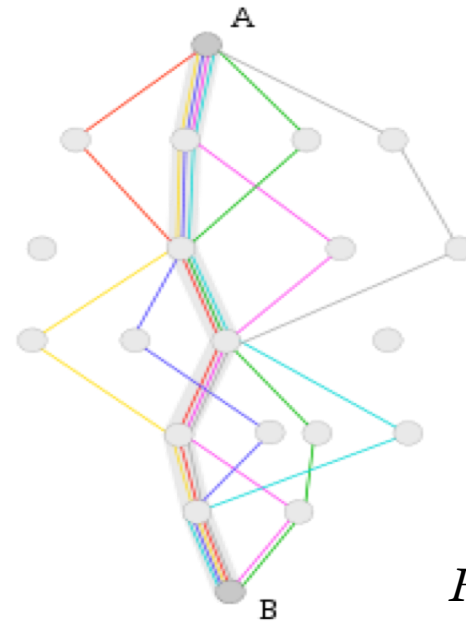
$$A_{\text{circle}} = \pi r^2$$



$$A_{\text{square}} = (2r)^2 = 4r^2$$

Ant Approach

- Move myself randomly around, sign the local best paths and (likely) proceeds the reserches on it.
- The idea is that the best path will be visited more often...
- ..but the random exploration of the world is the engine!



From wikipedia

Genetic algorithms

Probably the best (random) algorithm inspired by nature.

- ◆ Take a set of candidates, the parent generation
- ◆ randomly combine or change its in order to find other candidate
- ◆ select a new set of candidates from it, the child generation.

Genetic: How it Works!

Given a generation G_n , the next G_{n+1} is obtained by union of:

- ◆ THE WINNERS: some actual "best fitting" individual (not all, not the bestest)
 - ◆ THE SONS: some new individual obtained by join pieces of other individuals (crossover)
 - ◆ THE MUTANTS: some actual individual in which we have made some radomn changes (mutations)
- ◆ ...and the first generation G_0 ? randomly chosen? crafted selections? Does not matter!

Genetic: But It Works?!?

- ◆ The WINNERS selection act as a greedy algorithm provide an improvement toward th local maxima
- ◆ The SONS and the MUTANTS permits to occasionally jump to other best maxima.

The typical behaviour is slow or nothing improvements interlaced with some very fast climb toware best solution.